UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75789
ARIEL GUSTAVO RODRIGUES DA S
24PRG307 - 1 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(39)
2. As raizes de $7x^2 + 26x + 11$
3. O volume da esfera de raio 1.23000

Responda neste espaço

1.	2.	3.

307-75789 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75796
CASSIANO MAGNO CHAGAS E SA 24PRG307 - 2 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(37)
2. As raizes de $7x^2 + 15x + 2$
3. O volume da esfera de raio 1.04000

Responda neste espaço

1.	2.	3.

307-75796 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75808
CHRISTOPHER BUCH FILIPAK 24PRG307 - 3
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(35)
2. As raizes de $11x^2 + 30x + 14$
3. O volume da esfera de raio 2.77000

Responda neste espaço

1.	2.	3.

307-75808 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75815
ERAN MARTINEZ RAMOS 24PRG307 - 4
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(31)
2. As raizes de $2x^2 + 17x + 11$
3. O volume da esfera de raio 1.63000

Responda neste espaço

1.	2.	3.

307-75815 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75822
HENRIQUE IVANKIO GUSSE 24PRG307 - 5
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(32)
2. As raizes de $25x^2 + 22x + 3$
3. O volume da esfera de raio 3.36000

Responda neste espaço

1.	2.	3.

307-75822 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75839
IAN DOLABELLA OHATA 24PRG307 - 6
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(39)
2. As raizes de $4x^2 + 23x + 11$
3. O volume da esfera de raio 3.40000

Responda neste espaço

1.	2.	3.

307-75839 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75846
JOAO BONFIN LINO 24PRG307 - 7 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(39)
2. As raizes de $2x^2 + 21x + 5$
3. O volume da esfera de raio 1.92000

Responda neste espaço

1.	2.	3.

307-75846 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75853
JOAO LUCA FERNANDES DE OLIVE
24PRG307 - 8 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(33)
2. As raizes de $4x^2 + 30x + 21$
3. O volume da esfera de raio 3.17000

Responda neste espaço

1.	2.	3.

307-75853 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75860
JOAO PEDRO DE BRITO DUARTE 24PRG307 - 9 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(31)
2. As raizes de $16x^2 + 31x + 14$
3. O volume da esfera de raio 2.20000

Responda neste espaço

1.	2.	3.

307-75860 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75877
JOAO PEDRO MARIANO SUEKE 24PRG307 - 10 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(33)
2. As raizes de $11x^2 + 28x + 6$
3. O volume da esfera de raio 2.18000

Responda neste espaço

1.	2.	3.

307-75877 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75884
JOAO VICTOR GOMES VILELA GER
24PRG307 - 11 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(31)
2. As raizes de $10x^2 + 28x + 8$
3. O volume da esfera de raio 2.68000

Responda neste espaço

1.	2.	3.

307-75884 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75989
LUCAS ANTOSZCZYSZEN 24PRG307 - 12
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(31)
2. As raizes de $7x^2 + 23x + 18$
3. O volume da esfera de raio 3.97000

Responda neste espaço

1.	2.	3.

307-75989 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75891
LUIZ FELIPE GUEDES BUCCHERI 24PRG307 - 13 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(36)
2. As raizes de $2x^2 + 30x + 24$
3. O volume da esfera de raio 3.03000

Responda neste espaço

1.	2.	3.

307-75891 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75903
MANUELLA LEAL DE MEIRELLES F
24PRG307 - 14 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(32)
2. As raizes de $15x^2 + 28x + 7$
3. O volume da esfera de raio 2.88000

Responda neste espaço

1.	2.	3.

307-75903 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75910
MARIA EDUARDA DA SILVA 24PRG307 - 15
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(35)
2. As raizes de $8x^2 + 26x + 17$
3. O volume da esfera de raio 3.90000

Responda neste espaço

1.	2.	3.

307-75910 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75927
MATHEUS AITA FABRICIO DE CAR
24PRG307 - 16 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(39)
2. As raizes de $2x^2 + 15x + 27$
3. O volume da esfera de raio 2.26000

Responda neste espaço

1.	2.	3.

307-75927 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75934
MATHEUS LEANDRO DE BITENCOUR
24PRG307 - 17 entregar ate 8/agosto _____ / / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(32)
2. As raizes de $14x^2 + 25x + 6$
3. O volume da esfera de raio 2.73000

Responda neste espaço

1.	2.	3.

307-75934 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75941
MAYCON TAVARES WOLPE 24PRG307 - 18
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(39)
2. As raizes de $14x^2 + 17x + 3$
3. O volume da esfera de raio 3.85000

Responda neste espaço

1.	2.	3.

307-75941 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75958
PEDRO TOLEDO LEAL 24PRG307 - 19
entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(37)
2. As raizes de $12x^2 + 17x + 4$
3. O volume da esfera de raio 3.32000

Responda neste espaço

1.	2.	3.

307-75958 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75965
PHELIPE GABRIEL LIMA DA SILV 24PRG307 - 20 entregar ate 8/agosto _____ / _____ /

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(33)
2. As raizes de $13x^2 + 15x + 2$
3. O volume da esfera de raio 2.03000

Responda neste espaço

1.	2.	3.

307-75965 - ga/ a

UFPR - UP - UTFPR - PUCPr – 20/07/2024 - 09:40:36.6
Prof Dr P Kantek (pkantek@gmail.com)
Introdução a Python VIVX080a, V: 2.05 75972
REBECA CABRAL DOS SANTOS 24PRG307 - 21 entregar ate 8/agosto _____ / _____ / _____

Introdução a Python

Nascido no finzinho do século XX, e portanto herdeiro de:

- centenas de boas linguagens de programação que vieram antes e que – cada uma – e todas, tinham coisas boas ou ótimas que deviam ser aproveitadas (C, smalltalk, APL, Pascal, C++, Java,...). Aproveitadas é modo educado de falar: as coisas deviam ser copiadas *as is*.
- máquina sobrando: até aqui, faziam-se das tripas coração para economizar uns poucos bytes ou alguns ciclos de CPU. Isto não era mais necessário: havia e há máquina sobrando.

Tenha essas duas coisas em mente ao procurar entender porque o Python é a linguagem mais usada no mundo em 2024 e mais além. Se tiver alguma dúvida consulte `https://www.tiobe.com/tiobe-index/`. Aqui vale a pena lembrar aquela velha propaganda: *Tostines vende mais porque é fresquinho ou é fresquinho porque vende mais ?*

Instalação

Na continuação, você precisa instalar e usar algum Python. As alternativas

1. sob windows: vá em `python.org.br` e siga as instruções para ter o ambiente no seu computador (você precisa ser o administrador).
2. sob android: vá na loja do android e busque `python`. Escolha uma opção (pode ser o QPython 3) e mande ver. Mas, tenha em mente que a ergonomia aqui não é o ponto forte (basicamente pela dificuldade de usar o teclado).
3. em um pendrive: Se você tiver que usar computadores públicos e/ou emprestados, uma alternativa é ir a algum repositório público (por exemplo `sourceforge.net`) e lá baixar o `winpython` possivelmente em um pendrive: você terá um ambiente Python completo podendo ser usado em qualquer computador. Só precisa ter alguma paciência que o dito cujo é grande.
4. Jupyter: um ambiente sui-generis, mistura de compilador com editor de textos (este usando o padrão Markdown). Funciona em uma interface HTML (usada na web) e permite misturar textos e códigos. Aceita igualmente JUlia e há outros ambientes na fila (C++?).

Freeware - Não é desimportante que o Python entregue tudo o que ele entrega a custo zero. Seus concorrentes em boa medida cobram licenças de centenas ou milhares de dólares. Sem contar que a comunidade Python é a maior do mundo.

Pacotes - milhares deles. Eis alguns que podem interessar:

NUMPY, SYMPY e MATPLOTLIB	serão objeto de aulas mais à frente
ASTROPY	Astronomia, mas com profundo impacto no tratamento de imagens
TENSOR FLOW	A rede neural do Google. A base para aprendizado profundo
PYGAME	A base para gerar jogos e animações de maneira leve
PANDAS	ferramentas para Data analisys
OPENCV	Bibliotecas para computer vision
TURTLE	A célebre Tartaruga de Seymour Papert no ambiente Logo
NLTK	Natural Language Toolkit
PILLOW	Ferramentas para tratar imagens digitais

Algumas pérolas

Digite o programa abaixo. Ele calcula e retorna o fatorial de um número inteiro e positivo (a indentação precisa ser religiosamente observada)

```
def fat(x):
   if x>1:
      return x*fat(x-1)
   else:
      return 1
```

Tendo digitado e salvo (escolha um nome qualquer) o programa acima, é hora de executá-lo: Chame `fat(5)` e a resposta deverá ser, como sabemos, `120`. Agora chame `fat(100)` e você deverá ter um número enorme. Veja que nenhuma modificação foi necessária no código.

Baskara: A seguir, digite o programa abaixo, que como se pode ver, acha as 2 raízes de uma equação de segundo grau: $ax^2 + bx + c = 0$.

```
def baskara(a,b,c):
   return [(-b+(b**2-4*a*c)**0.5)/2*a, \
   (-b-(b**2-4*a*c)**0.5)/2*a]
```

A seguir, de volta ao `prompt` do Python digite `baskara(1,4,-10)` e veja o resultado. Aparecem x_1 e x_2, nenhuma novidade. Mas, agora execute `baskara(1,4,10)`. Perceba que agora as raizes são complexas, mas, de novo, nenhuma novidade, exceto que talvez a unidade imaginária que geralmente é i agora virou j pelo risco de confusão com alguma coisa.

Volume de uma esfera: A questão agora é calcular o volume de uma esfera de raio dado. Como se sabe da geometria $V = 4/3\pi \times r^3$. Então, o programa Python que calcula este volume é

```
def volesf(r):
   pi=3.14159265
   return pi*r**3
```

Agora chamando `volesf(1)` teremos a resposta que é igual a 3.14159265

Uma linguagem fácil de aprender

A seguir, alguns conceitos introdutórios que nos ajudarão a começar a navegação no oceano chamado Python. Lembrando que estamos falando de máquinas de Von Neumann e seus 5 comandos: (E/S, movimentação, aritmética e lógica, condicional e desvio).

atribuição a criação de uma variável, ou a alteração de seu valor

Pseudo-cód	C++	Python
A ← valor	int a=valor;	a=valor

O comando de atribuição pode ser dividido em 3 possibilidades:
var = constante, como por exemplo: `a=0`
var = var2, como por exemplo: `a=b`
var = expressão, como por exemplo: `a=a+1`

Aritmética Tudo o que sabemos da matemática vale aqui, sem exceção.

Pseudo-cód	C++	Python
$a = a + 1$	a++;	a=a+1
$a = a^2$	pow(a,2);	a=a**2
$a = a - 3$	a=a-3;	a=a-3
$a = a * x$	a=a*x;	a=a*3
$a = a/x$	a=a/x;	a=a//x (int) a=a/x (float)
$\sqrt{x}$	sqrt(x);	a**0.5
$a + bi$	?	a+bj
$a * (b + c)$	a*(b+c);	a*(b+c)
$b = sen(a)$	b=sin(a);	import numpy as np b=np.sin(a)

entrada de dados - Como os dados são obtidos

Pseudo-cód	C++	Python
leia a	int a; cin>>a;	a=int(input('msg'))

Uma maneira alternativa de fazer a "entrada de dados" meio fake é definir uma função recebendo parâmetros e imediatamente a seguir chamar essa mesma função com valores já digitados no mesmo script. Por exemplo

```
def dobro(a):
   return a*2
dobro(5)
```

Ajuda durante os testes não ter que redigitar as vezes grandes volumes de dados a cada teste. Quando tudo funcionar adequadamente, a função estará apta a funcionar com os dados da hora.

Tipos de dados - Embora o Python assuma esta responsabilidade (ao contrário de C++, por exemplo), ainda assim às vezes será necessário especificar algum tipo particular. Lembrando que os tipos básicos são `int`, `float`, `str` e `logic` (`True` e `False`). Existem funções Python que fazem algumas conversões:

int(a)	converte o string a em inteiro (se possível)
float(a)	converte o string a em flutuante (idem)
str(b)	converte o numérico b em string

Saída de dados - Como os resultados são comunicados

Pseudo-cód	C++	Python
escreva a	cout<< a;	print(a)
		print('msg')
		print('m',a)

Uma grande vantagem do print() é que não é necessário compatibilizar os formatos numérico e caracter (como por exemplo em `print('valor = ',val)`. A função print() se encarrega disso.

Python é interpretado

Se isso acarreta um desempenho menor (e quem se importa ?) em compensação ele têm diversas vantagens

- uso como calculadora
- aceita scripts e não apenas programas
- permite execução passo a passo, interativa, inclusive mudando o valor de variáveis

Conceito de bloco

Para atender à programação estruturada, qualquer linguagem de programação precisa implementar o conceito de bloco: um conjunto que qualquer quantidade de comandos e/ou blocos que são tratados como se fossem um único comando.

Em C++, tal conceito é implementado pelo uso de chaves. Tudo o que estiver entre { e } é tratado como um bloco. Em Python, e com o objetivo de não poluir o código, usa-se a indentação para estabelecer os blocos. Portanto, o que lá no C++ era apenas uma recomendação para ajudar o leitor (humano) a entender o código, em Python é mandatório: Um código bagunçado no Python não levará a lugar algum, certamente dando erros sintáticos ou semânticos.

Ainda falta muita coisa

Não é muita, mas é bem importante. Ainda estase limitado aqui no Python por não dispor do comando condicional (`if condição:`) nem dos comandos de repetição (`while condição:` ou `for...:`), mas eles virão nas próximas aulas.

☞ Para você fazer

Você deve arrumar um acesso a algum Python e digitar os 3 scripts acima e descobrir:

1. Os 3 primeiros dígitos de fat(32)
2. As raizes de $17x^2 + 29x + 5$
3. O volume da esfera de raio 3.60000

Responda neste espaço

1.	2.	3.

307-75972 - ga/ a