

Python para C++

Um pequeno auxílio para quem conhece C++ e pretende migrar para Python 3.

função	em C++	Em Python
Criação da linguagem	C em 1972 (Dennis Ritchie) e C++ em 1983 por Bjarne Stroustrup	Python em 1991 por Guido van Rossum.
Tipagem	forte e estática <code>(int x=10; float y=1.1;)</code>	forte e dinâmica <code>(x=10; y=1.1)</code>
Leitura de dados	<code>#include<iostream></code> <code>cout<<"Informe x";</code> <code>int x;</code> <code>cin>>x;</code>	<code>x=int(input("informe x"))</code>
Impressão de dados	<code>cout<<"valor = " << x;</code>	<code>print("valor = ",x)</code>
aspas	" para strings e ' para 1 caracter	" ou ', tanto faz, desde que pareadas. <code>x="don't do this"</code>
tipo booleano	Não existe, logo 0 é falso e qualquer coisa diferente de zero é verdadeiro	<code>True</code> e <code>False</code>
igualdade	<code>==</code>	<code>==</code>
atribuição	<code>=</code>	<code>=</code>
origem da contagem	zero	zero
divisão	depende dos envolvidos	<code>/</code> real e <code>//</code> inteira
potência	<code>#include<cmath></code> <code>x=pow(y,3);</code>	<code>x=y**3</code>
comando condicional	<code>if (condição) {</code> <code>...</code> <code>}</code>	<code>if condição:</code> <code>....</code>
repetição com while	<code>while (condição) {</code> <code>...</code> <code>}</code>	<code>while condição:</code> <code>....</code>
repetição com for	<code>for(i=0;i<n;i++) {</code> <code>...</code> <code>}</code>	<code>for i in range(n):</code> <code>....</code>
saída incondicional	<code>break</code>	<code>break</code>
saída com re-teste	<code>continue</code>	<code>continue</code>
for como interador	não existe	<code>for i in [1,2,5]:</code> <code>for s in 'Curitiba':</code>
vetor	<code>int x[10]={0};</code>	<code>x=[0]*10</code>
vetor heterogêneo	não existe	<code>x=[1,2,'oi',3.1]</code>
fatiamento	não existe	<code>[início:até:incremento]</code>
matriz zerada	<code>int x[10][20]={0};</code>	<code>import numpy as np</code> <code>x=np.zeros((10,20),int)</code>
matriz valorada	<code>float x[2][3]={10,20,30},</code> <code>{40,50,60};</code>	<code>import numpy as np</code> <code>x=np.array([[10,20,30],</code> <code>....[40,50,60]],float)</code>
tabela hash	não existe	<code>x = {chave:valor, ...}</code>
conjunto	não existe	<code>x = {valor, valor, ...}</code> ou <code>x=set([valor, valor, ...])</code>
pacotes	restritos	170.000 pacotes, o mundo é o limite
objetos	maravilhoso, uso opcional	maravilhoso, uso opcional
acentuação	depende de ajustes locais	Unicode e UTF
leitura arquivo	<code>#include<fstream></code> <code>ifstream f;</code> <code>f.open("c:/p/file.xxx");</code> <code>for(...) {</code> <code>f>>variável;</code> <code>}</code>	<code>f=open("c:/p/file.xxx","r")</code> <code>x=f.readline()</code> <code>pa=x.split()</code> <code>p=[]</code> <code>for j in pa:</code> <code>....p.append(int(j))</code> <code>....</code>
definição de função (não se executa nada depois de return)	<code>int função(int a, float b){</code> <code>...</code> <code>return x;</code> <code>}</code>	<code>def função(a,b):</code> <code>....</code> <code>....return x</code>

E, lembre-se, em qualquer caso: *Don't panic!*

Alguns exemplos

Em C++

```
#include<iostream>
#include<cmath>
using namespace std;
float pita(float a, float b){
    return sqrt(pow(a,2)+pow(b,2));
}
int main(){
    float ca,cb,hp;
    cout<<"Informe o cateto A: "<< endl;
    cin>>ca;
    cout<<"Informe o cateto B: "<< endl;
    cin>>cb;
    hp = pita(ca,cb);
    cout<<"A hipotenusa eh: "<<hp;
}
```

Em Python

```
def pita(a,b):
    return (a**2+b**2)**0.5
ca=float(input("Informe o cateto A: "))
cb=float(input("Informe o cateto B: "))
hp=pita(ca,cb)
print("A hipotenusa é: ",hp)
```

Em C++

```
#include<iostream>
using namespace std;
main() {
    int a[2][3]={1,2,3,4,5,6};
    int i,j,soma;
    soma=0;
    for (i=0;i<2;i++){
        for(j=0;j<3;j++){
            soma=soma+a[i][j];
        }
    }
    cout<<soma;
}
```

Em Python

```
import numpy as np
a=np.array([[1,2,3],[4,5,6]])
print(a.sum()) #--ou--
print(sum(a.ravel()))
----- ou -----
a=[[1,2,3],[4,5,6]]
i=0
s=0
while i<2:
    j=0
    while j<3:
        s=s+a[i][j]
        j=j+1
    i=i+1
print(s)
```

Em C++

```
#include<iostream>
using namespace std;
long long modulo = 1e9+7;
long long dp[10101];
long long alfa;
int i;
int main(){
    int n;
    cin>>n;
    dp[0]=1;
    dp[1]=1;
    dp[2]=5;
    for (i=3;i<=n;i++){
        alfa=dp[i-1]+4*dp[i-2]+2*dp[i-3];
        dp[i]=alfa%modulo;
    }
    cout<<dp[n]<<endl;
}
```

Em Python

```
d={0:1, 1:1, 2:5}
def muro(n):
    aa=d.get(n)
    if aa is None:
        d[n]=(muro(n-1)+4*muro(n-2)+2*muro(n-3))%(10**9+7)
        return d[n]
    else:
        return aa
print(muro(...))
```

Em C++

```
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;
float dcart(float x1, float y1, float x2, float y2){
    return sqrt(pow((x1-x2),2)+pow((y1-y2),2));
}
int main() {
    ifstream f;
    f.open("c:/p/n/184/f184001.myd");
    float dab=0,dacb=0,x1,x2,y1,y2;
    for (i=0;i<1000;i++){
        f>>x1;
        f>>y1;
        f>>x2;
        f>>y2;
        dab=dab+dcart(x1,y1,x2,y2);
        dacb=dacb+dcart(x1,y1,0.0,0.0)+dcart(0.0,0.0,x2,y2);
    }
    cout<<dab<<" " <<dacb;
}
```

Em Python

```
import numpy as np
def dcart(a,b):
    return (((a[0]-b[0])**2)+((a[1]-b[1])**2))**0.5
def f184():
    f=open("c:/p/n/184/f184001_exemplo.myd","r")
    i=0
    dab=dacb=0
    while i<1000:
        x=f.readline()
        nums=x.split()
        p1=[int(nums[0]),int(nums[1])]
        p2=[int(nums[2]),int(nums[3])]
        dab=dab+dcart(p1,p2)
        dacb=dacb+dcart(p1,[0,0])+dcart([0,0],p2)
        i=i+1
    print("D1 = ",dab," D2=",dacb)
f184
```