

Cartela de APLX

Instalação

Pode ser buscada em

www.dyalog.com/aplx.htm. Este produto era comprado (eu paguei 100 libras por ele), mas a empresa Microapl resolveu descontinuar seu desenvolvimento e desse ponto em diante ele foi integralmente liberado as i.s. Pode ser instalado nas versões 32 e 64 e pode ser liberado nas versões windows com o número 70702153 e serial W7CBS4CX.

Arrays

APL é orientada a arrays, que são arranjos de até 63 dimensões.

Tipo	Forma	Rank
escalar	0	0
vetor	elementos	1
matriz	lins e cols	2
sólido	planos, lins e cols	3
...	... linhas e colunas	n

arrays podem ser numéricos ou de caracteres.

Funções escalares

Funções escalares operam elemento a elemento, gerando a mesma forma no resultado. Se um dos operandos for escalar e o outro não, o escalar será repassado a todos os elementos do outro operando. Esta propriedade é conhecida como extensão de escalar. As funções escalares são pervasivas, isto é aplicam-se em todos os níveis de aninhamento. Por exemplo

(1 2 3)(2 2p4)+1
2 3 4 5 5
5 5

Note que pode usar o operador dieresis (¨) para aplicar uma função não escalar a cada elemento de um array.

Funç	Nome	o que faz	Exemplo
a+b	adição	soma de a com b	2+3 5
-a	oposto	calcula 0-a	-3 2 1 -3 -2 -1
a-b	subtração	a-b	2 3 8-3 -1 0 5
xa	sinal	se é negativo é -1, se é zero é 0 e se é positivo é 1	x0 5 -2 0 1 -1
a×b	multipli- cação	a vezes b	1 2 4x3 3 6 12
+a	inverso	1+a	+1 2 0.3 1 0.5 3.33333
a÷b	divisão	divide a por b	3 4 5÷2 1.5 2 2.5
f a	teto	o próximo inteiro maior ou igual a a	f2.1 3 3.1 3 3 4
a f b	máximo	o maior entre a e b	3 7 9 f5 5 7 9
l a	chão	o próximo inteiro menor ou igual a a	l5 2.1 -8.9 5 2 -9
a l b	mínimo	o menor entre a e b	1 2 3 l3 2 1 1 2 1
l a	absoluto	a sem sinal	l0 5 -2 0 5 2
a l b	resto de	o resto de b÷a (note a inversão)	8 3 7 l3 2 9 3 2 2
*a	exponen- cial	e elevado a a	*1 2.718281828
a*b	potência	a elevado a b	2 3*3 0.5 8 1.73205
o a	pi vezes	a vezes 3.1415...	o2 6.283185307
a o b	trigono- métrica	a=0→(1-b*2)*0.5 a=1→seno(b) a=2→cosseno(b) a=3→tan(b) a=4→ ((b*2)+1)*0.5 a=5→senh(b) a=6→cosh(b) a=7→tanh(b) a=-1→arcsen(b) a=-2→arccos(b) a=-3→arctan(b) a=-4→ ((b*2)-1)*0.5 a=-5→arcsenh(b) a=-6→arccosh(b) a=-7→arctanh(b)	o0 1 0.2 0 0.9797 1o3 0.1411200081 2o3 -0.98999249 3o3 -0.14254654 4o3 3.16227766 5o1 1.175201194 6o1 1.543080635 7o2.1 0.9704519366 -100.3 0.304692654 -200.3 1.266103673 -301.5 0.9827937232 -401.5 1.118033989 5o1 0.881373587 -601 -700.5 0.5493061443

Funç	Nome	o que faz	Exemplo
⌈a	log natural	log base e de a	⌈1 2 0 0.69314
a⌈b	logaritmo	log b na base a	5⌈3 0.6826061945
!a	fatorial	fat(a)	!5 120
a!b	binomial	combinações de b objetos tomados a a a	2!7 21
a>b	a maior que b	responde 1 se verdadeiro e 0 se falso	3>1 2 3 1 1 0
a<b	a menor que b	responde 1 se verdadeiro e 0 se falso	3<8 2 3 1 0 0
a≥b	a maior ou igual a b	responde 1 se verdadeiro e 0 se falso	3≥8 2 3 0 1 1
a≤b	a menor ou igual a b	responde 1 se verdadeiro e 0 se falso	3≤8 2 3 1 0 1
a=b	igual	1=sim, 0=não	'ABC'='AAA' 1 0 0
a≠b	diferente	1=sim, 0=não	'ABC'≠'AAA' 0 1 1
a~b	a E b	1=sim, 0=não	1 0 1~0 1 1 0 0 1
a∨b	a OU b	1=sim, 0=não	1 0 0∨0 1 0 1 1 0
a∨b	a não-E b	1=sim, 0=não	1 0 1~0 1 1 1 1 0
a∨b	a não-OU b	1=sim, 0=não	1 0 0∨0 1 0 0 0 1
~a	não a	inverte a	~0 1 0 0 1 0 1 1
?a	randômico	gera um aleatório entre 0 e o limite fornecido	?5 6 7 2 1 4

Funções mistas

As funções mistas podem alterar a forma do resultado.

Funç	Nome	o que faz	Exemplo
⌈a	matriz inversa	matriz inversa	⌈2 2p14 -2 1 1.5 -0.5
a⌈b	div matr	a é o vetor independente e b é a matriz de coeficientes	b÷3 5 7 a÷3 3p2 1 3 4 5 2 1 2 4 a⌈b 0.397 0.506 0.566
a?b	randômico	a inteiros entre 0 e b sem repetição	?3 6 1 5 6
a↗b	encode	converte números	2 2 2 ↗5 1 0 1
a↘b	decode	converte números	2 2 2↘1 0 1 5
!a	iôta	gera a sequência 1..a (ou 0..a-1)	!5 1 2 3 4 5
a!b	índice de	a primeira ocorrência de b em a	2 5 8 14!8 3
a e b	pertence	1 se a pertence a b e 0 senão	1 2 3 e4 5 1 1 1 0 0
a e b	find	em b onde a ocorre	'OI' e'GOIABA' 0 1 0 0 0 0
va	único	remove duplicatas de um vetor	v 1 1 2 2 4 5 1 2 4 5
a∨b	união	elementos em a ou em b	1 2 3 4∨2 4 6 8 1 2 3 4 6 8
a∩b	inter	elementos em a ou em b	1 2 3 ∩ 2 4 6 2
a~b	sem	remove b de a	'ABCDEF'~'AEI' BCDF
pa	forma	vetor forma de a	p1 2 4 5 6 5
apb	reforma- tação	os elementos b com a forma a	2 2p3 4 5 3 4 5 3
,a	ravel	vetor com elementos de a	,2 2p 1 2 3 2 1 2 3 2
a,b	concatena	a e b tomados juntos	1 2 3,6 7 1 2 3 6 7
φa	reverso	a em ordem reversa	φ5 3 1 2 2 1 3 5
aφb	rotação	roda b conforme a	3φ'CURITIBA' ITIBACUR
φa	transposta	a transposta de a	φ2 2p1 5 6 8 1 6 5 8
f a	primeiro	devolve o primeiro elemento de a	f3 4p15 1
a f b	take	toma a elementos de b	4 f'CURITIBA' CURI
a f b	drop	retira a elementos de b	4 f'CURITIBA' TIBA
ca	enclose	produz um escalar com a	c3 3p15 (vazio)
a c b	partição	divide b em array de vetores conforme a	1 1 2 2 c14 1 2 3 4
ca	disclose	desfaz a enclose	c3 3p15 3 3
a b b	pick	seleciona b de acordo com a	2 3~oi vuje la' j
a f b	indexador	usa a como índice em b	2010 20 30 20
Da	ind cresc	devolve os índices que põem a em ordem	D50 60 10 20 30 3 4 5 1 2
a f b	ind cresc	igual acima mas usa a a como régua (só caracteres)	veja manual

Mais

agora completar

Funç	Nome	o que faz	Exemplo
⌈a	ind decresc	devolve os índices que põem a em ordem	D50 60 10 20 30 2 1 5 4 3
a f b	ind decresc	igual acima mas usa a como régua (só caracteres)	veja manual
a o b	formata	formata b de acordo com a	'ZZZ9.9'∘i01 12 101.0 12.0
∘a	formata	formata a da melhor maneira	∘3.14 -22 1000 3.14 -22 1000
a∘b	format	formata b de acordo com a	5 1∘3.14 10 3.1 10.0
a∘b	format	formata b de acordo com exemplo a	'55.55'∘3.1 10 3.1 10
⌈a	execute	trata a literal a como comando APL	⌈'3+2' 5
-a	stop	retorna vazio mas executa a	-⌈mount 'c:' (vazio)
a- b	left	retorna a	1-2 1
↗a	pass	passa a sem alteração	↗a-14 1 2 3 4
a+ b	right	retorna b	1-2 2
a+⌈	entrada numérica	pára e a recebe digitado	a+⌈ ⌈: 1 2 3<cr>
a+⌈	entrada alfa	pára e a recebe o digitado	a+⌈ ABC<cr>
⌈a	saída alfa	mostra conteúdo de a	⌈a-xyz' xyz
a\ b	expansão	comprime b de acordo com a	1 0 1 1 2 1 0 2
a/ b	compressão	comprime b conforme a	1 0 1/1 2 3 1 3
ea	enlist	produz vetor de a (a profundo...)	e2 3p14 1 2 3 4 1 2
≡a	depth	informa a profundidade+1	≡2 3p14 2
a≡b	match	compara tudo inclusive prof.	1≡c1 2 0

Operadores

Modificam o funcionamento de funções.

Funç	Nome	o que faz	Exemplo
f/	redução	aplica a função	+/16 21
f\	scan	aplica a função sucessivamente	+/16 1 3 6 10 15 21
f"	each	aplica a função a cada elemento	(1 2 3),-4 5 6 1 4 2 5 3 6
f[]	eixo	indica a dimensão a considerar	a,[1]b a e b pela 1. dim.
°.f	produto externo	cria tabelas usando a função	(i2)°.×i3 1 2 3 2 4 6
f.g	produto interno	definição	(2 3p16)+.×(3 4p12) 38 44 50 56 83 98 113 128

Operadores definidos

Na definição de um operador, deve-se usar parênteses no seu cabeçalho. Veja

r←(oo1 operador)op1 A operador monádico - 1 arg
r←op1 (oo1 operador) op2 A operador monádico - 2 arg
r←(oo1 operador oo2) op1 A operador diádico - 1 arg
r←op1 (oo1 operador oo2) op2 A operador diádico - 2 arg

Veja um exemplo. Suponha o operador TROCA que inverte os operadores na chamada

vr←a (oo troca) b
[1] r←b oo av
10÷3
3.333333333
3÷troca 10
3.333333333
'viva' ('bela' troca) 'curitiba
curitiba bela viva

Neste último caso, o operador foi usado com variáveis ('bela') ao invés de ser chamado com uma função.

Funções definidas

Podem ter 0, 1 ou 2 argumentos, além de ambivalentes. A função pode ou não retornar resultado explícito. Para editar uma função clique com o mouse sobre seu nome ou faça JEDIT nome ou faça Vnome. Os cabeçalhos da função podem ser

funcao A niladica, sem resultado
r + funcao A niladica, com resultado
r + funcao op1 A monádica, com resultado
r + op1 funcao op2 A diádica, com resultado

Para localizar variáveis dentro da função, o cabeçalho deve conter seus nomes separados por .: Por exemplo, aqui a, b, c e d são variáveis locais

r + op1 funcao op2 ;a;b;c;d

As funções ambivalentes são definidas como diádicas, mas podem ser chamadas com apenas 1 operador

```

Vr←a funcao b
[1] :If 0=ONC 'a' a a não foi fornecido
[2]   a←88
[3]   :Endif

```

Estruturas de controle

Acrescimo ao APL e para poder aposentar o desvio ↵. São elas:

- :End pode substituir :EndIf, :EndFor, :EndWhile e os demais.
- As palavras começam com : e usualmente começam a linha.
- pode-se colocar o bloco inteiro em uma linha usando ⓪.
- os nomes não são case-sensitive, mas o APL reescreve-os.
- :Continue e :Leave se aplicam ao bloco mais interno, mas se os blocos tiverem nome, o nome pode ser citado e daí vale para ele.
- Tem mais formatos (veja o manual pág. 75)

```

:If <condição> ... :Else ... :EndIf
:While <condição> ... :EndWhile
:Repeat <expressão N> ... EndRepeat -- repete N vezes
:For <var> :In <expr vet> ... :EndFor

```

Variáveis e funções do sistema

Tem nome começando com ⓪. Algumas aceitam receber valores, outras são apenas para consulta.

Nome	função
⓪A	Alfabeto com letras maiúsculas
⓪a	Alfabeto com letras minúsculas
⓪AI	Informações de account
⓪BOX	Transforma vetor de/para matriz de caracteres
⓪CHART	Cria gráficos em janelas adicionais (pag. 272)
⓪DISPLAY	Mostra a estrutura de vaiável profunda
⓪EXPORT	Exporta array apl para arquivo formatado
⓪IMPORT	Importa array gravado por ⓪EXPORT
⓪FMT	Formatação de variáveis numéricas
⓪CR	Transforma função em array de caracteres
⓪FX	Fixa variável de caracteres em função
⓪MOUNT	Aloca bibliotecas (diretórios)
⓪NEW	Cria nova instância de classe
⓪LX	Expressão latente (executada automaticamente na carga do ws
⓪NCREATE	Criação de arquivo nativo (externo ao apl)
⓪NERASE	Eliminação de arquivo nativo (externo ao apl)
⓪READ	Leitura de arquivo nativo (pág. 368)
⓪NTIE	Abertura (open) de arquivo nativo
⓪NUNTIE	Fechamento (close) de arquivo nativo
⓪NWRITE	Gravação de arquivo nativo (pág. 368)
⓪PFKEY	Associa coisas a PFs (1-15), shift-PF (16-30) e CTRL-PF (31-45)
⓪PROFILE	Estudo de desempenho(pág. 380)
⓪R	Simula o Enter. p.ex:
⓪SS	Pesquisa e substituição de strings
⓪THIS	Referência ao objeto atual

Tradução de caracteres APL-externos

Quem faz isto é a função ⓪TR. 1 ⓪TR x: traduz de ASCII (inclusive UTF-8) para caracteres APL. ⓪TR. 1 ⓪TR x: traduz de APL para ASCII.

Programação GUI

Orientação a objetos

O conceito básico é de CLASSE. Uma classe contém funções e operadores (conhecidos como métodos) e dados, conhecidos como propriedades. O nome membros é usado para propriedades e métodos de uma classe. Depois de definir uma classe, você precisa criar uma instância dela conhecida como objeto. A classe é uma abstração (um conceito) enquanto o objeto representa uma particularização do conceito. Tipicamente há muitas instâncias de uma classe, cada uma com dados particulares, mas compartilhando métodos.

Quando se define uma classe pode-se dizer que ela é herdeira de outra classe. A nova classe é chamada child enquanto a classe velha é a pai ou classe base. Todos os métodos e propriedades da classe pai estão disponíveis para classe child. E do pai do pai e assim por diante. Métodos e propriedades podem ser redefinidas nas classes child.

Criando uma classe

```

)clear
'nomeclasse' ⓪IC ''
'nomeclasse' ⓪IC 'propriedade'
'nomeclasse' ⓪IC ⓪BOX 'prop1 prop2 ...'

```

No primeiro caso cria-se a classe 'nomeclasse' sem nenhuma propriedade. No segundo, apenas uma propriedade e no terceiro, quantas você quiser. Para ver o resultado, chame ⓪CLASSES.

Instância de uma classe

```

)clear
'formiga' ⓪IC ⓪BOX 'posx posy'
f1 ← ⓪NEW formiga
f1.posx ← 5
f1.posy ← 17

```

f1 é uma variável no ws. Se você pedir seu conteúdo, a resposta será [formiga] indicando que ela é uma instância de uma classe. Agora pode-se criar 5 instâncias da colonia

```

colonia ← ⓪NEW `5pformiga
colonia.posx ← ?5p100
colonia.posy ← 80

```

Note que colonia.posx para todos os efeitos é um vetor apl de 5 ocorrências.

Adicionando um método na classe Adicionando um método chamado andar na classe formiga, chama-se ⓪EDIT formiga.andar (o ↵ não funciona, nem clicar com o cursor) e escreve-se

```

r←andar
r←2p0
r[1]*posx*posx+`5+?10
r[2]*posy*posy+`10+?20

```

E agora, para fazer a colonia se mexer, faça

```

colonia.andar

```

e este método vai movimentar as 5 formigas, mostrando (e guardando dentro delas) qual a nova posição de cada formiga.

Adicionando nova propriedade Vamos incluir uma variável binária indicando se a formiga está ou não cansada

```

'formiga' ⓪IC 'cansada'
1
)classes
formiga
formiga
{formiga}
)vars
colonia
colonia
[formiga] [formiga] [formiga] [formiga]
colonia[1].cansada←1

```

Criando um construtor Pode-se criar um método chamado construtor (cujo nome é o MESMO da classe). Se existente, ele será chamado automaticamente pelo apl quando um novo objeto é criado.

```

)edit formiga.formiga
fsozinha←⓪NEW formiga
fsozinha.posx,fsozinha.posy
64 76

```

O construtor pode receber parâmetros como qualquer função apl. Neste caso pode-se ter ⓪EDIT formiga.formiga

```

formiga elementos
posx←elementos[1]
posy←elementos[2]

```

e daí na chamada deve-se fazer

```

f2←⓪NEW 'formiga' 8 78

```

Métodos do sistema Existem dúzias deles, mas vamos ver 2:

```

fsozinha.⓪CLASSNAME
formiga a retorna o nome da classe (caracteres)
fsozinha.⓪classname
formiga a não é sensível à caixa
fsozinha.DNL 3
andar a retorna os métodos da classe
fsozinha.DNL 2
cansada a retorna as propriedades da classe
posx
posy

```

Herança Vamos criar uma nova classe 'filhote' que vai herdar elementos de 'formiga'

```

'filhote:formiga' ⓪IC ''
)classes
filhote formiga
⓪CLASS filhote
{filhote} {formiga}
)edit filhote.filhote a criando o construtor
a que 'overraida' definições do pai
filh1←⓪NEW filhote a um novo objeto de filhote

```

Acompanhe agora o que dá para fazer

```

⓪CR'formiga'
formiga {
cansada
posx
posy

```

```

Vr←andar
r←2p0
r[1]*posx*posx+`5+?10
r[2]*posy*posy+`10+?20
v

```

```

Vformiga
posx←?100
posy←?100
v
}
⓪cr'filhote'
filhote : formiga {
vfilhote
posx*posx+`2+?4
posy*posy+`3+?6
v
}

```

Finalmente, veja o que acontece quando voce edita a classe ⓪EDIT formiga



Um pouco de formatação

Existem muitas (muitas!!!) maneiras de formatar números em APLX. Vejamos: tem a função α, tem a função ⓪FMT, tem a função ∇ monádica e a mesma diádica com operando esquerdo numérico simples e completo. O que vai se ver aqui é o com operando esquerdo alfanumérico, que é conhecido como ∇ por exemplo. Ele opera sujeito as seguintes regras: 5=formatação simples, com remoção de zeros à direita e supressão de brancos à esquerda e não aceita números negativos. Decoradores = qualquer coisa diferente de números pode aparecer e será reproduzido na saída.

```

' valor = 555,555.5' ∇ 1234.78 0.1
valor = 1,234.8 valor = .1

```

1=formata com negativo se ele for negativo e 2=formatata sempre com um sinal, seja positivo, seja negativo. 3=decorativo flutuante e não aceita negativo. 4=anula os efeitos de 1, 2 ou 3. 0=imprime o zero quando ele houver. 9=troca o zero por branco. 8=proteção de cheque. 6=encerra o campo de formatação. Este sujeito usa o FC para efeito de formatação. Seu valor padrão é ., *0_ _.

Comandos de sistema

Nome	função
⓪CLASSES	Lista as classes definidas pelo usuário (Há ⓪CLASSES)
⓪CLEAR	Apaga todas as coisas e reestabece o ws. Pode trocar seu tamanho
⓪COPY	copia coisas de um ws gravado para o ws atual
⓪DIGITS	Estabelece a quantidade de casas decimais (Há ⓪PP)
⓪DISPLAY	Mostra variável profunda e sua estrutura
⓪DROP	Apaga um ws em disco
⓪EDIT	editor para funções, operadores, variáveis e classes
⓪ERASE	Apaga coisas no ws atual
⓪FNS	List o nome das funções no ws atual
⓪IN	Carrega um arquivo gravado em outro sistema APL com ⓪OUT
⓪LIB	Mostra o nome dos workspaces gravados
⓪LOAD	Carrega um workspace
⓪OPS	Lista o nome dos operadores definidos
⓪OUT	Grava o conteúdo do ws atual em formato para ser lido em outro APL
⓪REPRESENT	Troca o pai de uma classe que você definiu
⓪RESET	Limpa todas as funções interrompidas
⓪SAVE	Salva o ws atual em disco
⓪SI	Mostra as interrupções de funções
⓪SINL	Mostra as interrupções de funções e suas variáveis locais
⓪SIS	Mostra as interrupções e o comando de cada função interrompida
⓪VARS	Lista os nomes das variáveis no ws atual
⓪WSID	Troca o nome do ws atual

Caracteres de caixas

```

11111111111
44444555555
56789012345
-----
⓪ U-|+|+|+
```

Desenhando com Draw

Deve-se criar uma janela com

```

R←'⓪' ⓪NEW 'Window'
R.scale←5 indica medidas em pixel
R.where←(0,50 450 450) tamanho da janela

```

Depois podem ser pedidos os comandos

```

R.Draw 'Pen' estilo,cor,tamanho:
estilo=1..6, cor=0..255 0.255 0.255, tam=num
R.Draw 'Line' (matriz N,2) entre () e N >= a 2

```

Finalmente, depois de tudo desenhado, fazer

```

0 0p⓪WE R

```