Nesta aula...

Conteúdo

1 Tabulação

Tabulação

Queremos tabular a função logaritmo:

x	$\log(x)$
1	$\log(1)$
2	$\log(2)$
$\vdots$	$\vdots$
10	$\log(10)$

- inicialmente: $x = 1$
- ciclo `while` $x \leq 10$
- em cada iteração: $x \leftarrow x + 1$

Tabular o logaritmo

```
import math   # para usar log(x)

# imprime o cabeçalho
print 'x', '\t', 'log(x)'
# valor inicial de x
x = 1
# ciclo
while x<=10:
   # imprime uma linha da tabela
   print x, '\t', math.log(x)
   # próximo valor de x
   x = x+1
```

Tabular o logaritmo

- `math.log(x)` dá o logaritmo de x na base e
- Logaritmo noutra base b:

$$\log_b(x) = \frac{\log(x)}{\log(b)}$$

- Em Python: `math.log(x,b)`
- Variantes:
  - incremento de x diferente
  - tabelar em ordem decrescente
  - em progressão geométrica

Tabulação em duas dimensões

Imprimir a tabuada da multiplicação:

$\times$	1	2	3	...	9
1	1	2	3	...	
2	2	4	6	...	
3	3	6	9	...	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
9					

- linhas e colunas: valores de 1 a 9
- intersecções: valor de $linha \times coluna$

Decomposição do problema

imprimeTabuada() função para imprimir a tabuada

imprimeLinha() função para imprimir uma linha

Imprimir a tabuada

```
def imprimeTabuada():
    "Imprime a tabuada da multiplicação."
    linha = 1
    while linha<=9:
      imprimeLinha(linha)
      linha = linha+1
```

Imprimir uma linha da tabela

```
def imprimeLinha(linha):
    "Imprime os produtos de uma linha."
    coluna = 1
    while coluna<=9:
      print linha*coluna, # vírgula no final
      coluna = coluna+1
    print                 # muda de linha
```

Execução

```
>>> imprimeTabuada()
1 2 3 4 5 6 7 8 9
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
```

Formatação

```
>>> '%d' % 52         # inteiro
'52'

>>> '%3d' % 52        # inteiro com 3 caracteres
' 52'

>>> '%f' % 6.1        # vírgula flutuante
'6.100000'

# v.f. com 10 caracteres e 2 casas decimais
>>> '%10.2f' % 6.1
'      6.10'
```

Alinhar as colunas

```
def imprimeLinha(linha):
    "Imprime os produtos de uma linha."
    coluna = 1
    while coluna<=9:
      # formata com 2 caracteres
      print '%2d' % (linha*coluna),
      coluna = coluna+1
    print
```

Execução

```
>>> imprimeTabuada()
 1  2  3  4  5  6  7  8  9
 2  4  6  8 10 12 14 16 18
 3  6  9 12 15 18 21 24 27
 4  8 12 16 20 24 28 32 36
```

```
 5 10 15 20 25 30 35 40 45
 6 12 18 24 30 36 42 48 54
 7 14 21 28 35 42 49 56 63
 8 16 24 32 40 48 56 64 72
 9 18 27 36 45 54 63 72 81
```

Tabulação triangular

- metade da tabela é redundante porque $linha \times coluna = coluna \times linha$
- introduzir uma restrição adicional: $coluna \leq linha$
- resultado: uma tabela triangular

Tabulação triangular

```
def imprimeLinha(linha):
    "Imprime os produtos de uma linha."
    coluna = 1
    while coluna<=linha:
      # cada entrada com 2 caracteres
      print '%2d' % (linha*coluna),
      coluna = coluna+1
    print
```

Execução

```
>>> imprimeTabuada()
 1
 2  4
 3  6  9
 4  8 12 16
 5 10 15 20 25
 6 12 18 24 30 36
 7 14 21 28 35 42 49
 8 16 24 32 40 48 56 64
 9 18 27 36 45 54 63 72 81
```

Códigos de caracteres

- cada caracter ASCII/ISO10646/Unicode tem associado um código numérico
- funções `chr` e `ord` convertem entre caracteres e códigos

```
# caracter com código 33?
>>> chr(33)
'!'
```

```
# caracter com código 65?
>>> chr(65)
'A'

# qual o código de 'B'?
>>> ord('B')
66
```

Tabelar códigos dos caracteres

```
def tabela():
   col = 0          # número da coluna
   code = 32        # desde código 32
   while code<=126: # até código 126
      print code,chr(code),'\t',
      code = code+1
      col = col+1
      if col>=6:    # mudar de linha?
         col = 0
         print
   print
```

Tabelar códigos dos caracteres

```
32      33 !    34 "    35 #    36 $    37 %
38 &    39 '    40 (    41 )    42 *    43 +
44 ,    45 -    46 .    47 /    48 0    49 1
50 2    51 3    52 4    53 5    54 6    55 7
56 8    57 9    58 :    59 ;    60 <    61 =
62 >    63 ?    64 @    65 A    66 B    67 C
68 D    69 E    70 F    71 G    72 H    73 I
74 J    75 K    76 L    77 M    78 N    79 O
80 P    81 Q    82 R    83 S    84 T    85 U
86 V    87 W    88 X    89 Y    90 Z    91 [
92 \    93 ]    94 ^    95 _    96 `    97 a
98 b    99 c    100 d   101 e   102 f   103 g
104 h   105 i   106 j   107 k   108 l   109 m
110 n   111 o   112 p   113 q   114 r   115 s
116 t   117 u   118 v   119 w   120 x   121 y
122 z   123 {   124 |   125 }   126 ~
```