



Python Básico

Variáveis

Pode dar um nome qualquer a uma variável desde que obedeça às seguintes regras:

1. Apenas uma palavra,
2. Apenas pode usar letras, números e o carácter sublinhado (underscore),
3. Não pode começar com um número,
4. Variáveis que começam com sublinhado () são consideradas "desnecessárias".

Exemplo:

```
spam = 'Hello'
_spam = 'Hello'
```

Operadores matemáticos

Operadores	Operação	Exemplo
**	Exponente	2 ** 3 = 8
%	Módulo/Resto	22 % 8 = 6
//	Divisão inteira	22 // 8 = 2
/	Divisão	22 / 8 = 2.75
*	Multiplificação	3 * 3 = 9
-	Subtração	5 - 2 = 3
+	Adição	2 + 2 = 4

Comentários

Comentário em linha: `# This is a comment`

Comentário multilinha: `# This is a
multiline comment`

Código com comentário: `a = 1 # initialization`
Tenha em atenção os dois espaços em frente ao comentário.

Função docstring: `def foo():
 """ This is a function docstring
 You can also use:
 ''' Function Docstring '''
 """`

Tipos de dados

Tipos de dados	Exemplos
Inteiros	-2, -1, 0, 1, 2, 3, 4, 5
Números de virgula fluante	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', 'W0rld'

Co-funded by the
Erasmus+ Programme
of the European Union



Python Básico

Variáveis

Pode dar um nome qualquer a uma variável desde que obedeça às seguintes regras:

1. Apenas uma palavra,
2. Apenas pode usar letras, números e o carácter sublinhado (underscore),
3. Não pode começar com um número,
4. Variáveis que começam com sublinhado () são consideradas "desnecessárias".

Exemplo:

```
spam = 'Hello'
_spam = 'Hello'
```

Operadores matemáticos

Operadores	Operação	Exemplo
**	Exponente	2 ** 3 = 8
%	Módulo/Resto	22 % 8 = 6
//	Divisão inteira	22 // 8 = 2
/	Divisão	22 / 8 = 2.75
*	Multiplificação	3 * 3 = 9
-	Subtração	5 - 2 = 3
+	Adição	2 + 2 = 4

Comentários

Comentário em linha: `# This is a comment`

Comentário multilinha: `# This is a
multiline comment`

Código com comentário: `a = 1 # initialization`
Tenha em atenção os dois espaços em frente ao comentário.

Função docstring: `def foo():
 """ This is a function docstring
 You can also use:
 ''' Function Docstring '''
 """`

Tipos de dados

Tipos de dados	Exemplos
Inteiros	-2, -1, 0, 1, 2, 3, 4, 5
Números de virgula fluante	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', 'W0rld'

Co-funded by the
Erasmus+ Programme
of the European Union





Python Básico

Condições

Um bloco:

```
x = 3
if x == 3:
    print('x is 3')
```

Dois blocos:

```
mark = 80
if mark >= 50:
    print('pass')
else:
    print('fail')
```

Múltiplos blocos:

```
mark = 80
if mark >= 65:
    print('credit')
elseif mark >= 50:
    print('pass')
else:
    print('fail')
```

Range

Conta de 0 a 9:

```
range(10)
```

Começa em 0 e vai até, mas não inclui 10

Conta de 1 a 10:

```
range(1, 11)
```

Conta de 10 a 1:

```
range(10, 0, -1)
```

Conta em passos de 2:

```
range(0, 11, 2)
```

Contagem decrescente em passos de 2:

```
range(10, 0, -2)
```

Ciclos

While:

```
x = 0
while x < 4:
    print(x)
    x = x + 1
```

Sair de um ciclo utilizando o break:

```
x = 0
while x < 4:
    print(x)
    if x == 2:
        break
    x = x + 1
```

Reiniciar o ciclo utilizando o continue:

```
x = 0
while x <= 10:
    x = x + 1
    if x % 2 == 0:
        continue
    print('%s is odd' % x)
```

For:

```
for i in range(10):
    print(i)
```

Itera sobre uma string:

```
for c in 'Hello':
    print(c)
```

Entrada e saída

Imprime mensagem:

```
print('Hello world!')
```

Imprime múltiplos valores:

```
ndays = 365
print('There are', ndays, 'in a year')
```

Pede ao utilizador uma string:

```
name= input('What is your name? ')
```

Pede ao utilizador um número inteiro:

```
num = int(input('Enter a number: '))
```

Comparação

Igual a:

```
x == 3:
```

Diferente de:

```
x != 3:
```

Menor que:

```
x < 3:
```

Maior que:

```
x > 3:
```

Menor que ou igual a:

```
x <= 3:
```

Maior que ou igual a:

```
x >= 3:
```

O resultado de uma comparação é um booleano:

```
True
```

ou

```
False
```



Python Básico

Condições

Um bloco:

```
x = 3
if x == 3:
    print('x is 3')
```

Dois blocos:

```
mark = 80
if mark >= 50:
    print('pass')
else:
    print('fail')
```

Múltiplos blocos:

```
mark = 80
if mark >= 65:
    print('credit')
elseif mark >= 50:
    print('pass')
else:
    print('fail')
```

Range

Conta de 0 a 9:

```
range(10)
```

Começa em 0 e vai até, mas não inclui 10

Conta de 1 a 10:

```
range(1, 11)
```

Conta de 10 a 1:

```
range(10, 0, -1)
```

Conta em passos de 2:

```
range(0, 11, 2)
```

Contagem decrescente em passos de 2:

```
range(10, 0, -2)
```

Ciclos

While:

```
x = 0
while x < 4:
    print(x)
    x = x + 1
```

Sair de um ciclo utilizando o break:

```
x = 0
while x < 4:
    print(x)
    if x == 2:
        break
    x = x + 1
```

Reiniciar o ciclo utilizando o continue:

```
x = 0
while x <= 10:
    x = x + 1
    if x % 2 == 0:
        continue
    print('%s is odd' % x)
```

For:

```
for i in range(10):
    print(i)
```

Itera sobre uma string:

```
for c in 'Hello':
    print(c)
```

Entrada e saída

Imprime mensagem:

```
print('Hello world!')
```

Imprime múltiplos valores:

```
ndays = 365
print('There are', ndays, 'in a year')
```

Pede ao utilizador uma string:

```
name= input('What is your name? ')
```

Pede ao utilizador um número inteiro:

```
num = int(input('Enter a number: '))
```

Comparação

Igual a:

```
x == 3:
```

Diferente de:

```
x != 3:
```

Menor que:

```
x < 3:
```

Maior que:

```
x > 3:
```

Menor que ou igual a:

```
x <= 3:
```

Maior que ou igual a:

```
x >= 3:
```

O resultado de uma comparação é um booleano:

```
True
```

ou

```
False
```