

Python - Booleanos

Booleanos Python (bool)

Em Python, **bool** é um subtipo de **int type** . Um objeto bool possui dois valores possíveis e é inicializado com palavras-chave Python, **True** e **False** .

Exemplo

```
>>> a=True
>>> b=False
>>> type(a), type(b)
(<class 'bool'>, <class 'bool'>)
```

Um objeto bool é aceito como argumento para **digitar** funções de conversão. Com True como argumento, a função int() retorna 1, float() retorna 1,0; enquanto para False, eles retornam 0 e 0,0 respectivamente. Temos uma versão de um argumento da função **complex()** .

Se o argumento for um objeto complexo, ele é tomado como parte real, fixando o coeficiente imaginário em 0.

Exemplo

```
a=int(True)
print ("bool to int:", a)
a=float(False)
print ("bool to float:", a)
a=complex(True)
print ("bool to complex:", a)
```

Ao executar este código, você obterá a seguinte **saída** -

```
bool to int: 1
bool to float: 0.0
bool to complex: (1+0j)
```

Expressão Booleana Python

A expressão booleana do Python é uma expressão avaliada como um valor booleano. Q sempre envolve um **operador de comparação** . No exemplo abaixo veremos como os



operadores de comparação podem nos fornecer os valores booleanos. O método `bool()` é usado para retornar o valor verdade de uma expressão.

Syntax: `bool([x])`

Returns `True` if X evaluates to true else false.

Without parameters it returns false.

Abaixo temos exemplos que usam fluxos de números e valores booleanos como parâmetros para a função `bool`. Os resultados são verdadeiros ou falsos dependendo do parâmetro.

Exemplo

```
# Check true
a = True
print(bool(a))

# Check false
a = False
print(bool(a))

# Check 0
a = 0.0
print(bool(a))

# Check 1
a = 1.0
print(bool(a))

# Check Equality
a = 5
b = 10
print(bool( a==b))

# Check None
a = None
print(bool(a))

# Check an empty sequence
a = ()
print(bool(a))

# Check an empty mapping
a = {}
print(bool(a))

# Check a non empty string
a = 'Tutorialspoint'
print(bool(a))
```