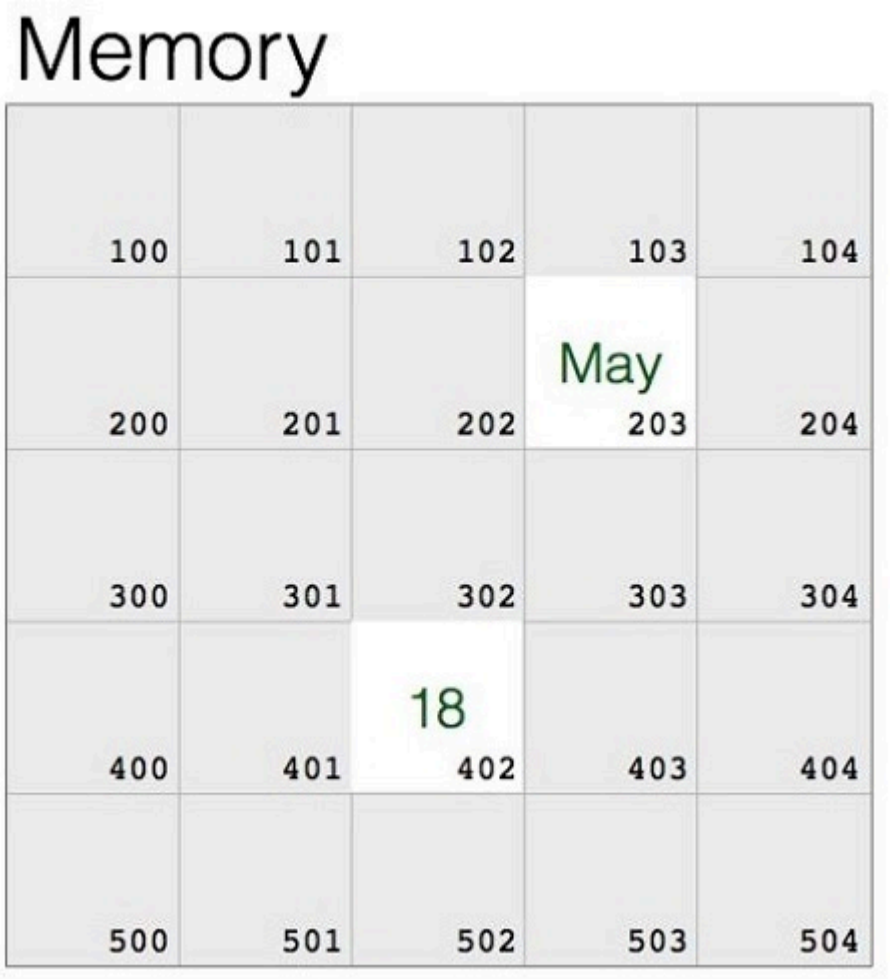
Python - Variáveis

Variáveis Python

Variáveis Python são locais de memória reservados usados para armazenar valores em um programa Python. Isso significa que ao criar uma variável você reserva algum espaço na memória. Com base no tipo de dados de uma variável, o interpretador Python aloca memória e decide o que pode ser armazenado na memória reservada. Portanto, ao atribuir diferentes tipos de dados às variáveis Python, você pode armazenar números inteiros, decimais ou caracteres nessas variáveis.

Endereços de memória

Itens de dados pertencentes a diferentes tipos de dados são armazenados na memória do computador. Os locais de memória do computador possuem um número ou endereço, representado internamente em forma binária. Os dados também são armazenados em formato binário, pois o computador funciona segundo o princípio da representação binária. No diagrama a seguir, uma string **May** e um número **18** são mostrados armazenados em locais de memória.



Se você conhece a linguagem assembly, você converterá esses itens de dados e o endereço de memória e fornecerá instruções em linguagem de máquina. No entanto, não é fácil para todos. O tradutor de linguagem, como o intérprete Python, realiza esse tipo de conversão. Ele armazena o objeto em um local de memória escolhido aleatoriamente. **A função id()** integrada do Python retorna o endereço onde o objeto está armazenado.

```
>>> "May"
May
>>> id("May")
2167264641264

>>> 18
18
>>> id(18)
140714055169352
```

Uma vez armazenados os dados na memória, eles devem ser acessados repetidamente para a realização de determinado processo. Obviamente, buscar os dados do seu ID é complicado. Linguagens de alto nível como Python tornam

possível fornecer um alias ou rótulo adequado para se referir à localização da memória.

No exemplo acima, vamos rotular o local de maio como mês e o local em que 18 está armazenado como idade. Python usa o operador de atribuição (=) para vincular um objeto ao rótulo.

```
>>> month="May"
>>> age=18
```

O objeto de dados (maio) e seu nome (mês) possuem o mesmo id(). O id() de 18 e idade também são iguais.

```
>>> id(month)
2167264641264
>>> id(age)
140714055169352
```

O rótulo é um identificador. Geralmente é chamado de variável. Uma variável Python é um nome simbólico que é uma referência ou ponteiro para um objeto.

Criando Variáveis Python

Variáveis Python não precisam de declaração explícita para reservar espaço de memória ou você pode dizer para criar uma variável. Uma variável Python é criada automaticamente quando você atribui um valor a ela. O sinal de igual (=) é usado para atribuir valores às variáveis.

O operando à esquerda do operador = é o nome da variável e o operando à direita do operador = é o valor armazenado na variável. Por exemplo -

Exemplo para criar variáveis Python

Este exemplo cria diferentes tipos (um número inteiro, um ponto flutuante e uma string) de variáveis.

```
counter = 100          # Creates an integer variable
miles   = 1000.0       # Creates a floating point variable
name    = "Zara Ali"   # Creates a string variable
```

Imprimindo variáveis Python

Depois de criarmos uma variável Python e atribuirmos um valor a ela, podemos imprimi-la usando a função **print()** . A seguir está a extensão do exemplo anterior e mostra como imprimir diferentes variáveis em Python:

Exemplo para imprimir variáveis Python

Este exemplo imprime variáveis.

```
counter = 100          # Creates an integer variable
miles   = 1000.0       # Creates a floating point variable
name    = "Zara Ali"   # Creates a string variable

print (counter)
print (miles)
print (name)
```

Aqui, 100, 1000,0 e "Zara Ali" são os valores atribuídos às variáveis counter , miles e name , respectivamente. Ao executar o programa Python acima, produz o seguinte resultado -

```
100
1000.0
Zara Ali
```


Excluindo Variáveis Python

Você pode excluir a referência a um objeto numérico usando a instrução del. A sintaxe da instrução del é -

```
del var1[,var2[,var3[....,varN]]]
```

Você pode excluir um único objeto ou vários objetos usando a instrução del. Por exemplo -

```
del var
del var_a, var_b
```

Exemplo

Os exemplos a seguir mostram como podemos excluir uma variável e se tentarmos usar uma variável excluída, o interpretador Python gerará um erro:

```
counter = 100
print (counter)

del counter
print (counter)
```

Isso produzirá o seguinte resultado:

```
100
Traceback (most recent call last):
  File "main.py", line 7, in <module>
    print (counter)
NameError: name 'counter' is not defined
```


Obtendo o tipo de uma variável

Você pode obter o tipo de dados de uma variável Python usando a função interna type() do python como segue.

Exemplo: Tipo de variáveis de impressão

```
x = "Zara"
y =  10
z =  10.10

print(type(x))
print(type(y))
print(type(z))
```

Isso produzirá o seguinte resultado:

```
<class 'str'>
<class 'int'>
<class 'float'>
```


Castando variáveis Python

Você pode especificar o tipo de dados de uma variável com a ajuda da conversão da seguinte maneira:

Exemplo

Este exemplo demonstra a distinção entre maiúsculas e minúsculas das variáveis.

```
x = str(10)    # x will be '10'
y = int(10)    # y will be 10
z = float(10)  # z will be 10.0

print( "x =", x )
print( "y =", y )
print( "z =", z )
```

Isso produzirá o seguinte resultado:

```
x = 10
y = 10
z = 10.0
```

Sensibilidade a maiúsculas e minúsculas de variáveis Python

Variáveis Python diferenciam maiúsculas de minúsculas, o que significa que **Idade** e **idade** são duas variáveis diferentes:

```
age = 20
Age = 30
```

```
print( "age =", age )
print( "Age =", Age )
```

Isso produzirá o seguinte resultado:

```
age = 20
Age = 30
```

Variáveis Python - Atribuição Múltipla

Python permite inicializar mais de uma variável em uma única instrução. No caso a seguir, três variáveis têm o mesmo valor.

```
>>> a=10
>>> b=10
>>> c=10
```

Em vez de atribuições separadas, você pode fazer isso em uma única instrução de atribuição da seguinte forma -

```
>>> a=b=c=10
>>> print (a,b,c)
10 10 10
```

No caso a seguir, temos três variáveis com valores diferentes.

```
>>> a=10
>>> b=20
>>> c=30
```

Essas instruções de atribuição separadas podem ser combinadas em uma. Você precisa fornecer nomes de variáveis separados por vírgula à esquerda e valores separados por vírgula à direita do operador =.

```
>>> a,b,c = 10,20,30
>>> print (a,b,c)
10 20 30
```

Vamos tentar alguns exemplos no modo script: -

```
a = b = c = 100

print (a)
print (b)
print (c)
```

Isso produz o seguinte resultado:

```
100
100
100
```

Aqui, um objeto inteiro é criado com o valor 1 e todas as três variáveis são atribuídas ao mesmo local de memória. Você também pode atribuir vários objetos a diversas variáveis. Por exemplo -

```
a,b,c = 1,2,"Zara Ali"

print (a)
print (b)
print (c)
```

Isso produz o seguinte resultado:

```
1
2
Zara Ali
```

Aqui, dois objetos inteiros com valores 1 e 2 são atribuídos às variáveis aeb respectivamente, e um objeto string com o valor "Zara Ali" é atribuído à variável c.

Variáveis Python - Convenção de Nomenclatura

Cada variável Python deve ter um nome exclusivo como a, b, c. O nome de uma variável pode ser significativo, como cor, idade, nome, etc. Existem certas regras que devem ser observadas ao nomear uma variável Python:

- O nome de uma variável deve começar com uma letra ou sublinhado
- O nome de uma variável não pode começar com um número ou qualquer caractere especial como $, (, *% etc.

- Um nome de variável só pode conter caracteres alfanuméricos e sublinhados (Az, 0-9 e _)
- Os nomes das variáveis Python diferenciam maiúsculas de minúsculas, o que significa que Nome e NOME são duas variáveis diferentes em Python.
- Palavras-chave reservadas do Python não podem ser usadas para nomear a variável.

Se o nome da variável contiver várias palavras, devemos usar estes padrões de nomenclatura -

- **Camel case** - A primeira letra é minúscula, mas a primeira letra de cada palavra subsequente está em maiúscula. Por exemplo: kmPorHora, preçoPorLitro
- **Pascal case** - A primeira letra de cada palavra está em maiúscula. Por exemplo: KmPorHora, PreçoPorLitro
- **Snake case** - Use um único caractere de sublinhado (_) para separar palavras. Por exemplo: km_por_hora, preço_por_litro

Exemplo

A seguir estão os nomes de variáveis Python válidos:

```
counter = 100
_count  = 100
name1 = "Zara"
name2 = "Nuha"
Age  = 20
zara_salary = 100000

print (counter)
print (_count)
print (name1)
print (name2)
print (Age)
print (zara_salary)
```

Isso produzirá o seguinte resultado:

```
100
100
```

```
Zara
Nuha
20
100000
```

Exemplo

A seguir estão nomes de variáveis Python inválidos:

```
1counter = 100
$_count  = 100
zara-salary = 100000

print (1counter)
print ($count)
print (zara-salary)
```

Isso produzirá o seguinte resultado:

```
File "main.py", line 3
   1counter = 100
       ^
SyntaxError: invalid syntax
```

Exemplo

Depois de usar uma variável para identificar um objeto de dados, ela poderá ser usada repetidamente sem seu valor id(). Aqui, temos variáveis de altura e largura de um retângulo. Podemos calcular a área e o perímetro com essas variáveis.

```
>>> width=10
>>> height=20
>>> area=width*height
>>> area
200
>>> perimeter=2*(width+height)
>>> perimeter
60
```

O uso de variáveis é especialmente vantajoso ao escrever scripts ou programas. O script a seguir também usa as variáveis acima.

```
#! /usr/bin/python3

width = 10
height = 20
area = width*height
perimeter = 2*(width+height)
print ("Area = ", area)
print ("Perimeter = ", perimeter)
```

Salve o script acima com extensão .py e execute na linha de comando. O resultado seria -

```
Area = 200
Perimeter = 60
```

Constantes em Python

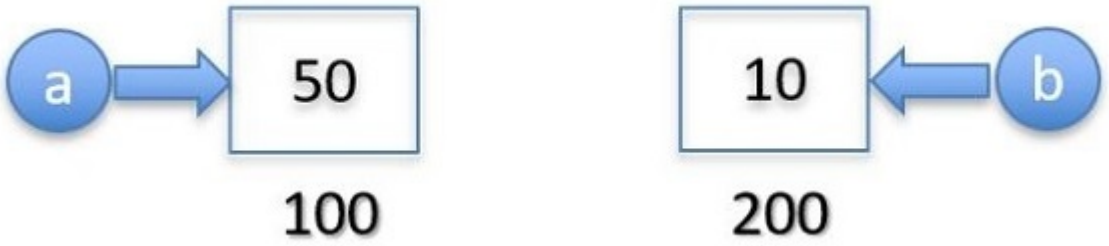
Python não possui nenhuma constante definida formalmente. No entanto, você pode indicar uma variável a ser tratada como uma constante usando nomes em letras maiúsculas com sublinhados. Por exemplo, o nome PI_VALUE indica que você não deseja que a variável seja redefinida ou alterada de forma alguma.

> *A convenção de nomenclatura usando letras maiúsculas é às vezes chamada de caixa de cobra gritante - onde estão todas em letras maiúsculas (gritando) e os sublinhados (cobras).*

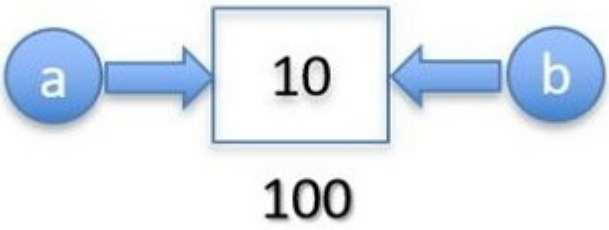
Variáveis Python versus C/C++

O conceito de variável funciona de maneira diferente em Python e em C/C++. Em C/C++, uma variável é um local de memória nomeado. Se a=10 e também b=10, ambos são dois locais de memória diferentes. Suponhamos que o endereço de memória seja 100 e 200, respectivamente.

Se um valor diferente for atribuído a "a" - digamos 50, 10 no endereço 100 será substituído.

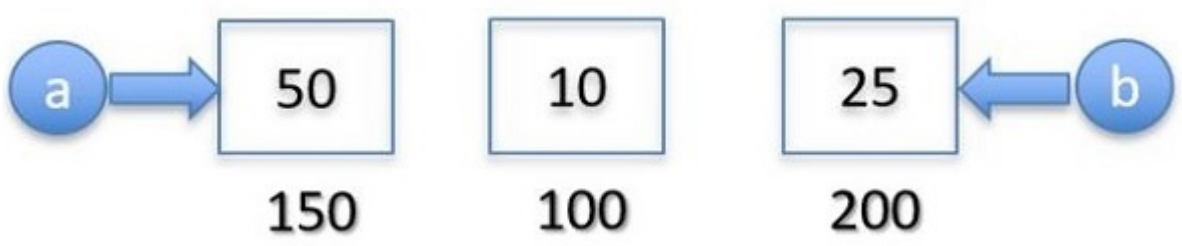


Uma variável Python refere-se ao objeto e não ao local da memória. Um objeto é armazenado na memória apenas uma vez. Múltiplas variáveis são, na verdade, vários rótulos para o mesmo objeto.



A instrução a=50 cria um novo objeto **int** 50 na memória em algum outro local, deixando o objeto 10 referido por "b".

Além disso, se você atribuir algum outro valor a b, o objeto 10 permanecerá sem referência.



O mecanismo coletor de lixo do Python libera a memória ocupada por qualquer objeto não referenciado.

O operador de identidade do Python **retorna** True se ambos os operandos tiverem o mesmo valor id().

```
>>> a=b=10
>>> a is b
True
>>> id(a), id(b)
(140731955278920, 140731955278920)
```