

Python - Atualizar tuplas

Em Python, tupla é um tipo de dados imutável. Um objeto imutável não pode ser modificado depois de criado na memória.

Exemplo 1

Se tentarmos atribuir um novo valor a um item da tupla com o operador slice, o Python gerará `TypeError`. Veja o exemplo a seguir -

```
tup1 = ("a", "b", "c", "d")
tup1[2] = 'Z'
print ("tup1: ", tup1)
```

Ele produzirá a seguinte **saída** -

```
Traceback (most recent call last):
  File "C:\Users\mlath\examples\main.py", line 2, in <module>
    tup1[2] = 'Z'
    ~~~~^^^
TypeError: 'tuple' object does not support item assignment
```

Portanto, não é possível atualizar uma tupla. Portanto, a classe tupla não fornece métodos para adicionar, inserir, excluir e classificar itens de um objeto tupla, como a classe lista.

Como atualizar uma tupla Python?

Você pode usar uma solução alternativa para atualizar uma tupla. Usando a função `list()`, converta a tupla em uma lista, execute as operações desejadas de anexar/inserir/remover e, em seguida, analise a lista de volta ao objeto tupla.

Exemplo 2

Aqui, convertemos a tupla em uma lista, atualizamos um item existente, acrescentamos um novo item e classificamos a lista. A lista é convertida novamente em tupla.

```
tup1 = ("a", "b", "c", "d")
print ("Tuple before update", tup1, "id(): ", id(tup1))

list1 = list(tup1)
list1[2]='F'
```



```
list1.append('Z')
list1.sort()
print ("updated list", list1)

tup1 = tuple(list1)
print ("Tuple after update", tup1, "id(): ", id(tup1))
```

Ele produzirá a seguinte **saída** -

```
Tuple before update ('a', 'b', 'c', 'd') id(): 2295023084192
updated list ['F', 'Z', 'a', 'b', 'd']
Tuple after update ('F', 'Z', 'a', 'b', 'd') id(): 2295021518128
```

No entanto, observe que o `id()` de `tup1` antes e depois da atualização são diferentes. Isso significa que um novo objeto tupla é criado e o objeto tupla original não é modificado no local.