


Python Tuple- Learn with Examples

Tuple

<code>()</code>	An empty tuple
<code>(33, 55, 77)</code>	A tuple of numbers
<code>(33, 3.3, (3+3j))</code>	A tuple of mixed numbers
<code>(33, '33', [3, 3])</code>	A tuple of mixed types
<code>(('x','y'), ('X','Y'))</code>	A tuple of tuples

<code>:</code>	Slicing	<code>tuple()</code>	
<code>+</code>	Concatenation	<code>del keyword</code>	
<code>*</code>	Repeation	<code>in keyword</code>	

Tuple

Python

- ***tupla***: sequenza di valori (***oggetto iterabile***)
 - valori possono essere di qualsiasi tipo
(*anche di diverso tipo all'interno della stessa tupla*)
 - valori ***indicizzati*** (l'indice è un valore intero)
- differenza rispetto alle liste:
 - le tuple sono ***immutabili***

- *assegnamento* di una sequenza di valori separati da , (virgola)
 - eventualmente racchiusi fra parentesi tonde

```
linguaggi = 'python', 'java', 'sql'  
linguaggi = ('python', 'java', 'sql')  
print(type(linguaggi), linguaggi)  
#output <class 'tuple'> ('python', 'java', 'sql')
```

- il costruttore *tuple* permette di definire una tupla partendo da
 - una sequenza *vuota*
 - una *stringa*
 - una *lista*

```
t_vuota = tuple()
print(type(t_vuota), t_vuota)
#output <class 'tuple'> ()
```

```
vocali = tuple('aeiou')
print(vocali)
#output ('a', 'e', 'i', 'o', 'u')
```

```
l_giorni = ['lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom']
print(type(l_giorni), l_giorni)
#output <class 'list'> ['lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom']
```

```
t_giorni = tuple(l_giorni)
print(type(t_giorni), t_giorni)
#output <class 'tuple'> ('lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom')
```

- `[n]` selezione valore di indice `n`
- **slicing** (analogo a liste e stringhe)
- operatori di **confronto** (operano in sequenza sugli elementi)

```
giorno = t_giorni[0]
print(giorno)           #output lun
festivi = t_giorni[5:]
print(festivi)         #output ('sab', 'dom')

t1 = (2,5,8)
t2 = (2,4,9)
print(t1 < t2)        #output false
```

```

punto3d = (100,25,80)
x, y, z = punto3d
print(x,y,z)                #output 100, 25, 80
print('x = {0} y = {1} z = {2}'.format(x, y, z))
#output x = 100 y = 25 z = 80

alfa = 2
beta = 3
beta , alfa = alfa, beta
print('alfa = {0} beta = {1}'.format(alfa, beta))
# output alfa = 3 beta = 2

```

```

mail = 'alberto.ferrari@unipr.it'
nome, dominio = mail.split('@')
print(nome, dominio)       #output alberto.ferrari unipr.it

```