

3.10. Tipo tuplas

Las tuplas son objetos de tipo *secuencia*, específicamente es un tipo de dato [lista](#) inmutable. Esta no puede modificarse de ningún modo después de su creación.

3.10.1. Métodos

Son muy similares a las [listas](#) y comparten varias de sus funciones y métodos integrados, aunque su principal diferencia es que son inmutables. El objeto de tipo *tupla* integra una serie de métodos integrados a continuación:

3.10.1.1. count()

Este método recibe un elemento como argumento, y cuenta la cantidad de veces que aparece en la tupla.

```
>>> valores = ("Python", True, "Zope", 5)
>>> print("True ->", valores.count(True))
True -> 1
>>> print("'Zope' ->", valores.count("Zope"))
'Zope' -> 1
>>> print("5 ->", valores.count(5))
5 -> 1
```

3.10.1.2. index()

Comparte el mismo método [index\(\)](#) del tipo [lista](#). Este método recibe un elemento como argumento, y devuelve el índice de su primera aparición en la [tupla](#).

```
>>> valores = ("Python", True, "Zope", 5)
>>> print(valores.index(True))
1
>>> print(valores.index(5))
3
```

El método devuelve un excepción [ValueError](#) si el elemento no se encuentra en la tupla, o en el entorno definido.

```
>>> valores = ("Python", True, "Zope", 5)
>>> print(valores.index(4))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: tuple.index(x): x not in tuple
```

 v: 3.7 ▼

3.10.2. Convertir a tuplas

Para convertir a *tipos tuplas* debe usar la función `tuple()`, la cual [está integrada](#) en el interprete Python.

Truco

Para más información consulte las funciones integradas para [operaciones de secuencias](#).

3.10.3. Ejemplos

A continuación, se presentan algunos ejemplos de su uso:

Definir simple de tupla

```
1 tupla = 12345, 54321, "hola!"
```

Definir tuplas anidadas

```
1 otra = tupla, (1, 2, 3, 4, 5)
```

Operación asignar de valores de una tupla en variables

```
1 x, y, z = tupla
```

Cuidar seguimiento del número de la numeración

Una tarea común es iterar sobre una secuencia mientras cuidas el seguimiento de la numeración de un elemento.

Podría usar un bucle `while` con un contador o un bucle `for` usando la función `range()` y la función `len()`:

```
>>> tecnologias = ("Zope", "Plone", "Pyramid")
>>> for i in range(0, len(tecnologias)):
...     print(i, tecnologias[i])
...
0 Zope
```

```
1 Plone
2 Pyramid
```

Pero, Python provee la palabra reservada `enumerate` para esto:

 v: 3.7 ▼

```
>>> tecnologias = ("Zope", "Plone", "Pyramid")
>>> for contador, tecnologia in enumerate(tecnologias, start=1):
...     print(contador, tecnologia)
...
1 Zope
2 Plone
3 Pyramid
>>>
```

Caso real de conexión a BD

A continuación, un ejemplo más apegado a la realidad que busca establecer una conexión a una BD:

```
1  conexion_bd = (
2      "127.0.0.1",
3      "root",
4      "qwerty",
5      "nomina",
6  )
7  print("Conexión típica:", conexion_bd, type(conexion_bd))
8
9  conexion_completa = (
10     conexion_bd,
11     "3307",
12     "10",
13 )
14 print(
15     "\nConexión con parámetros adicionales:",
16     conexion_completa,
17     type(conexion_completa),
18     "\n",
19 )
20
21 print("IP de la BD:", conexion_completa[0][0])
22 print("Usuario de la BD:", conexion_completa[0][1])
23 print("Contraseña de la BD:", conexion_completa[0][2])
24 print("Nombre de la BD:", conexion_completa[0][3])
25 print("Puerto de conexión:", conexion_completa[1])
26 print("Tiempo de espera en conexión:", conexion_completa[2])
```

Este script da como resultado el siguiente mensaje:

```
Definiendo conexión a BD MySQL
=====

Conexión típica: ('127.0.0.1', 'root', 'qwerty', 'nomina') <class 'tuple'>
```

```
Conexión con parámetros adicionales: (('127.0.0.1', 'root', 'qwerty',  
'nomina'), '3307', '10') <class 'tuple'>
```

```
IP de la BD: 127.0.0.1  
Usuario de la BD: root  
Contraseña de la BD: qwerty  
Nombre de la BD: nomina  
Puerto de conexión: 3307  
Tiempo de espera en conexión: 10
```

Más información acerca de MySQL y Python <http://mysql-python.sf.net/MySQLdb.html>

 v: 3.7 ▼

3.10.4. Ayuda integrada

Usted puede consultar toda la documentación disponible sobre las **tuplas** desde la [consola interactiva](#) de la siguiente forma:

```
>>> help(tuple)
```



Importante

Usted puede descargar el código usado en esta sección haciendo clic [aquí](#).



Truco

Para ejecutar el código `tipo_tuplas.py`, abra una consola de comando, acceda al directorio donde se encuentra el mismo, y ejecute el siguiente comando:

```
$ python tipo_tuplas.py
```



Ver también

Consulte la sección de [lecturas suplementarias](#) del entrenamiento para ampliar su conocimiento en esta temática.

¿Cómo puedo ayudar?

¡Mi soporte está aquí para ayudar!

Mi horario de oficina es de lunes a sábado, de 9 AM a 5 PM. [GMT-4 - Caracas, Venezuela](#).

La hora aquí es actualmente 7:35 PM GMT-4.

Mi objetivo es responder a todos los mensajes dentro de un día hábil.

 v: 3.7 ▼

[Contáctenos en la sección de soporte](#)



What do you think?

1 respuesta



Upvote



Funny



Love



Surprised



Angry



Sad

0 Comentarios

 1 Acceder ▼

G

Sé el primero en comentar...

INICIAR SESIÓN CON

O REGISTRARSE CON DISQUS 

Nombre