TUPLAS

Actividades iniciadas el 20sept21

**Tuplas de Python**

mytuple = ("apple", "banana", "cherry")

**Tupla**

Las tuplas se utilizan para almacenar varios elementos en una sola variable.

Tupla es uno de los 4 tipos de datos integrados en Python que se utilizan para almacenar colecciones de datos, los otros 3 son Lista, Conjunto y Diccionario, todos con diferentes calidades y usos.

Una tupla es una colección ordenada e inmutable.

Las tuplas se escriben entre paréntesis.

### **Example**

Create a Tuple:

thistuple = ("apple", "banana", "cherry")  
print(thistuple)

**Elementos de tupla**

Los elementos de tupla están ordenados, no se pueden cambiar y permiten valores duplicados.

Los elementos de tupla están indexados, el primer elemento tiene índice [0], el segundo elemento tiene índice [1], etc.

**Ordenado**

Cuando decimos que las tuplas están ordenadas, significa que los elementos tienen un orden definido y ese orden no cambiará.

**Inmutable**

Las tuplas no se pueden cambiar, lo que significa que no podemos cambiar, agregar o eliminar elementos después de que se haya creado la tupla.

**Permitir duplicados**

Dado que las tuplas están indexadas, pueden tener elementos con el mismo valor:

### **Example**

Tuples allow duplicate values:

thistuple = ("apple", "banana", "cherry", "apple", "cherry")  
print(thistuple)

**Longitud de la tupla**

Para determinar cuántos elementos tiene una tupla, use la función len():

### **Example**

Print the number of items in the tuple:

thistuple = ("apple", "banana", "cherry")  
print(len(thistuple))

**Crear tupla con un elemento**

Para crear una tupla con un solo elemento, debe agregar una coma después del elemento; de lo contrario, Python no lo reconocerá como una tupla.

### **Example**

One item tuple, remember the comma:

thistuple = ("apple",)  
print(type(thistuple))  
  
#NOT a tuple  
thistuple = ("apple")  
print(type(thistuple))

**Elementos de tupla: tipos de datos**

Los elementos de tupla pueden ser de cualquier tipo de datos:

### **Example**

String, int and boolean data types:

tuple1 = ("apple", "banana", "cherry")  
tuple2 = (1, 5, 7, 9, 3)  
tuple3 = (True, False, False)

Una tupla puede contener diferentes tipos de datos:

### **Example**

A tuple with strings, integers and boolean values:

tuple1 = ("abc", 34, True, 40, "male")

**Type()**

Desde la perspectiva de Python, las tuplas se definen como objetos con el tipo de datos 'tupla':

<class 'tuple'>

### **Example**

What is the data type of a tuple?

mytuple = ("apple", "banana", "cherry")  
print(type(mytuple))

**El constructor tuple()**

También es posible usar el constructor tuple() para hacer una tupla.

### **Example**

Using the tuple() method to make a tuple:

thistuple = tuple(("apple", "banana", "cherry")) # note the double round-brackets  
print(thistuple)

**Colecciones de Python (Arrays)**

Hay cuatro tipos de datos de recopilación en el lenguaje de programación Python:

* List (lista) es una colección ordenada y modificable. Permite miembros duplicados.
* Tuple (tupla) es una colección ordenada e inmutable. Permite miembros duplicados.
* Set (conjunto) es una colección desordenada y no indexada. No hay miembros duplicados.
* Dictionary (diccionario) es una colección ordenada\* y modificable. No hay miembros duplicados.

\* A partir de la versión 3.7 de Python, los diccionarios están ordenados. En Python 3.6 y versiones anteriores, los diccionarios están desordenados.

Actividades del 21 de septiembre del 2021

Al elegir un tipo de colección, es útil comprender las propiedades de ese tipo. Elegir el tipo correcto para un conjunto de datos en particular podría significar la retención del significado y podría significar un aumento en la eficiencia o la seguridad.

**Python: acceder a elementos de tupla**

**Acceder a elementos de tupla**

Puede acceder a los elementos de tupla consultando el número de índice, entre corchetes:

### **Example**

Print the second item in the tuple:

thistuple = ("apple", "banana", "cherry")  
print(thistuple[1])

**Nota**: El primer elemento tiene índice 0.

**Indexación negativa**

La indexación negativa significa comenzar desde el final.

-1 se refiere al último elemento, -2 se refiere al penúltimo elemento, etc.

### **Example**

Print the last item of the tuple:

thistuple = ("apple", "banana", "cherry")  
print(thistuple[-1])

**Rango de índices**

Puede especificar un rango de índices especificando dónde comenzar y dónde terminar el rango.

Al especificar un rango, el valor de retorno será una nueva tupla con los elementos especificados.

### **Example**

Return the third, fourth, and fifth item:

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print(thistuple[2:5])

Esto devolverá los elementos de la posición 2 a la 5.

Recuerde que el primer elemento es la posición 0, y tenga en cuenta que el elemento en la posición 5 NO está incluido

**Nota**: La búsqueda comenzará en el índice 2 (incluido) y terminará en el índice 5 (no incluido).

Recuerde que el primer elemento tiene índice 0.

Al omitir el valor inicial, el rango comenzará en el primer elemento:

### **Example**

This example returns the items from the beginning to, but NOT included, "kiwi":

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print(thistuple[:4])

Al omitir el valor final, el rango continuará hasta el final de la lista:

### **Example**

This example returns the items from "cherry" and to the end:

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print(thistuple[2:])

**Rango de índices negativos**

Especifique índices negativos si desea iniciar la búsqueda desde el final de la tupla:

### **Example**

This example returns the items from index -4 (included) to index -1 (excluded)

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print(thistuple[-4:-1])

La indexación negativa significa comenzar desde el final de la tupla.

Este ejemplo devuelve los elementos del índice -4 (incluido) al índice -1 (excluido)

Recuerda que el último elemento tiene el índice -1,

**Compruebe si el elemento existe**

Para determinar si un elemento específico está presente en una tupla, use la palabra clave in:

### **Example**

Check if "apple" is present in the tuple:

thistuple = ("apple", "banana", "cherry")  
if "apple" in thistuple:  
  print("Yes, 'apple' is in the fruits tuple")

**Python - Actualizar tuplas**

Las tuplas no se pueden cambiar, lo que significa que no puede cambiar, agregar ni eliminar elementos una vez que se crea la tupla.

Pero existen algunas soluciones.

**Cambiar valores de tupla**

Una vez que se crea una tupla, no puede cambiar sus valores. Las tuplas son inmutables.

Pero hay una solución. Puede convertir la tupla en una lista, cambiar la lista y volver a convertir la lista en una tupla.

### **Example**

Convert the tuple into a list to be able to change it:

x = ("apple", "banana", "cherry")  
y = list(x)  
y[1] = "kiwi"  
x = tuple(y)  
  
print(x)

**Agregar elementos**

Dado que las tuplas son inmutables, no tienen un método append() incorporado, pero hay otras formas de agregar elementos a una tupla.

1. Convertir en una lista: al igual que la solución alternativa para cambiar una tupla, puede convertirla en una lista, agregar sus elementos y convertirlos nuevamente en una tupla.

### **Example**

Convert the tuple into a list, add "orange", and convert it back into a tuple:

thistuple = ("apple", "banana", "cherry")  
y = list(thistuple)  
y.append("orange")  
thistuple = tuple(y)

2. Agregue tupla a una tupla. Puede agregar tuplas a las tuplas, por lo que si desea agregar un elemento (o muchos), cree una nueva tupla con los elementos y agréguela a la tupla existente:

### **Example**

Create a new tuple with the value "orange", and add that tuple:

thistuple = ("apple", "banana", "cherry")  
y = ("orange",)  
thistuple += y  
print(thistuple)

**Nota**: Al crear una tupla con un solo elemento, recuerde incluir una coma después del elemento, de lo contrario, no se identificará como una tupla.

**Eliminar elementos**

**Nota**: no puede eliminar elementos en una tupla.

Las tuplas no se pueden cambiar, por lo que no puede eliminar elementos de ellas, pero puede usar la misma solución alternativa que usamos para cambiar y agregar elementos de tupla:

### **Example**

Convert the tuple into a list, remove "apple", and convert it back into a tuple:

thistuple = ("apple", "banana", "cherry")  
y = list(thistuple)  
y.remove("apple")  
thistuple = tuple(y)

O puede eliminar la tupla por completo:

### **Example**

The del keyword can delete the tuple completely:

thistuple = ("apple", "banana", "cherry")  
del thistuple  
print(thistuple) #this will raise an error because the tuple no longer exists

**Python - Desempaquetar tuplas**

**Desembalaje de una tupla**

Cuando creamos una tupla, normalmente le asignamos valores. Esto se denomina "empaquetar" una tupla:

### **Example**

Packing a tuple:

fruits = ("apple", "banana", "cherry")

Pero, en Python, también podemos extraer los valores nuevamente en variables. Esto se denomina "desembalaje":

### **Example**

Unpacking a tuple:

fruits = ("apple", "banana", "cherry")  
  
(green, yellow, red) = fruits  
  
print(green)  
print(yellow)  
print(red)

Actividades del 22 de sept de 2021

**Nota**: La cantidad de variables debe coincidir con la cantidad de valores en la tupla; de lo contrario, debe usar un asterisco para recopilar los valores restantes como una lista.

**Usando Asterisk \***

Si el número de variables es menor que el número de valores, puede agregar un \* al nombre de la variable y los valores se asignarán a la variable como una lista:

### **Example**

Assign the rest of the values as a list called "red":

fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")  
  
(green, yellow, \*red) = fruits  
  
print(green)  
print(yellow)  
print(red)

Si el asterisco se agrega a otro nombre de variable que no sea el anterior, Python asignará valores a la variable hasta que el número de valores restantes coincida con el número de variables restantes.

### **Example**

Add a list of values the "tropic" variable:

fruits = ("apple", "mango", "papaya", "pineapple", "cherry")  
  
(green, \*tropic, red) = fruits  
  
print(green)  
print(tropic)  
print(red)

**Python – Bucle en Tuplas**

**Bucle a través de una tupla**

Puede recorrer los elementos de la tupla utilizando un bucle for.

### **Example**

Iterate through the items and print the values:

thistuple = ("apple", "banana", "cherry")  
for x in thistuple:  
  print(x)

**Recorrer los números de índice**

También puede recorrer los elementos de la tupla consultando su número de índice.

Utilice las funciones range() y len() para crear un iterable adecuado.

### **Example**

Print all items by referring to their index number:

thistuple = ("apple", "banana", "cherry")  
for i in range(len(thistuple)):  
  print(thistuple[i])

**Usar un bucle while**

Puede recorrer los elementos de la lista utilizando un ciclo while.

Utilice la función len() para determinar la longitud de la tupla, luego comience en 0 y recorra los elementos de la tupla consultando sus índices.

Recuerde aumentar el índice en 1 después de cada iteración.

### **Example**

Print all items, using a while loop to go through all the index numbers:

thistuple = ("apple", "banana", "cherry")  
i = 0  
while i < len(thistuple):  
  print(thistuple[i])  
  i = i + 1

**Python: unir tuplas**

**Unir dos tuplas**

Para unir dos o más tuplas, puede usar el operador +:

### **Example**

Join two tuples:

tuple1 = ("a", "b" , "c")  
tuple2 = (1, 2, 3)  
  
tuple3 = tuple1 + tuple2  
print(tuple3)

**Multiplicar tuplas**

Si desea multiplicar el contenido de una tupla un número determinado de veces, puede usar el operador \*:

### **Example**

Multiply the fruits tuple by 2:

fruits = ("apple", "banana", "cherry")  
mytuple = fruits \* 2  
  
print(mytuple)

# Python - Tuple Methods

**Métodos de tupla**

Python tiene dos métodos integrados que puede usar en tuplas.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [count()](https://www.w3schools.com/python/ref_tuple_count.asp) | Devuelve el número de veces que ocurre un valor especificado  en una tupla. |
| [index()](https://www.w3schools.com/python/ref_tuple_index.asp) | Busca en la tupla un valor especificado y devuelve la posición  de donde se encontró. |

**Método Python Tuple count()**

### **Example**

Return the number of times the value 5 appears in the tuple:

thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
  
x = thistuple.count(5)  
  
print(x)

**Definición y uso**

El método count() devuelve el número de veces que aparece un valor especificado en la tupla.

**Sintaxis**

tuple.count(value)

**Valores paramétricos**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *value* | Requerido. El elemento a buscar. |

**Método Python Tuple index()**

### **Example**

Search for the first occurrence of the value 8, and return its position:

thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
  
x = thistuple.index(8)  
  
print(x)

**Definición y uso**

El método index() encuentra la primera aparición del valor especificado.

El método index() genera una excepción si no se encuentra el valor.

**Sintaxis**

tuple.index(value)

**Valores paramétricos**

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| value | Requerido. El elemento a buscar. |