



El futuro digital
es de todos

MinTIC

TUPLAS

+ Rogerio Orlando Beltrán Castro



Tuplas

En la conversión de un tiempo a horas, minutos y segundos, en la sección 3.6, usamos n-uplas (o tuplas) como una construcción que nos permitía que una función devolviera múltiples valores.

En programación, en general, al querer modelar objetos de la vida real, es muy común que queramos describir un objeto como un agrupamiento de datos de distintos tipos. Veamos algunos ejemplos:

Una fecha la podemos querer representar como la terna día (un número entero), mes (una cadena de caracteres), y año (un número entero), y tendremos por ejemplo: (25, "Mayo", 1810).

Como datos de los alumnos queremos guardar número de padrón, nombre y apellido, como por ejemplo (89766, "Alicia", "Hacker").

Es posible anidar tuplas: como datos de los alumnos queremos guardar número de padrón, nombre, apellido y fecha de nacimiento, como por ejemplo: (89766, "Alicia", "Hacker", (9, "Julio", 1988)).

Elementos y segmentos de tuplas

Las tuplas son secuencias, igual que las cadenas, y se puede utilizar la misma notación de índices que en las cadenas para obtener cada una de sus componentes.

El primer elemento de (25, "Mayo", 1810) es 25.

El segundo elemento de (25, "Mayo", 1810) es "Mayo".

El tercer elemento de (25, "Mayo", 1810) es 1810.

```
>>> t=(25, "Mayo", 1810)
```

```
>>> t[0]
```

```
25
```

```
>>> t[1]
```

```
'Mayo'
```

```
>>> t[2]
```

```
1810
```

Elementos y segmentos de tuplas

Todas las secuencias en Python comienzan a numerarse desde 0. Es por eso que se produce un error si se quiere acceder al n-ésimo elemento de un tupla:

```
>>> t[3]
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: tuple index out of rango

También se puede utilizar la notación de rangos, que se vio aplicada a cadenas para obtener una nueva tupla, con un subconjunto de componentes. Si en el ejemplo de la fecha queremos quedarnos con un par que sólo contenga día y mes podremos tomar el rango [:2] de la misma:

```
>>> t[:2]
```

```
(25, 'Mayo')
```

Ejercicio ¿Cuál es el cuarto elemento de (89766, "Alicia", "Hacker", (9, "Julio", 1988))?

Las tuplas son inmutables

Al igual que con las cadenas, las componentes de las tuplas no pueden ser modificadas:

```
>>> t[2] = 2008
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'tuple' object does not support item assignment

Longitud de tuplas

A las tuplas también se les puede aplicar la función `len()` para calcular su longitud. El valor de esta función aplicada a una tupla nos indica cuántas componentes tiene esa tupla.

```
>>> len(t)
```

```
3
```

Longitud de tuplas

A las tuplas también se les puede aplicar la función `len()` para calcular su longitud. El valor de esta función aplicada a una tupla nos indica cuántas componentes tiene esa tupla.

```
>>> len(t)
```

```
3
```

Ejercicio. ¿Cuál es la longitud de `(89766, "Alicia", "Hacker", (9, "Julio", 1988))`?

Longitud de tuplas

Una tupla vacía es una tupla con 0 componentes, y se la indica como ().

```
>>> z=()
```

```
>>> len(z)
```

```
0
```

```
>>> z[0]
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: tuple index out of range

Empaquetado y desempaquetado de tuplas

Si a una variable se le asigna una secuencia de valores separados por comas, el valor de esa variable será la tupla formada por todos los valores asignados. A esta operación se la denomina empaquetado de tuplas.

```
>>> a=125
```

```
>>> b="\#"
```

```
>>> c="Ana"
```

```
>>> d=a,b,c
```

```
>>> len(d)
```

```
3
```

```
>>> d
```

```
(125, '\#', 'Ana')
```

```
>>>
```

Empaquetado y desempaquetado de tuplas

Si se tiene una tupla de longitud k , se puede asignar la tupla a k variables distintas y en cada variable quedará una de las componentes de la tupla. A esta operación se la denomina desempaquetado de tuplas.

```
>>> x,y,z = d
```

```
>>> x
```

```
125
```

```
>>> y
```

```
'\#'
```

```
>>> z
```

```
'Ana'
```

Empaquetado y desempaquetado de tuplas

Advertencia: Si las variables no son distintas, se pierden valores. Y si las variables son exactamente k se produce un error.

```
>>> p,p,p = d
```

```
>>> p
```

```
'Ana'
```

```
>>> m,n = d
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: too many values to unpack

```
>>> m,n,o,p=d
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

ValueError: need more than 3 values to unpack

Comparación de tuplas

Dos tuplas son iguales cuando tienen el mismo tamaño y cada uno de sus elementos correspondientes tienen el mismo valor:

```
>>> (1, 2) == (3 / 2, 1 + 1)
```

```
True
```

```
>>> (6, 1) == (6, 2)
```

```
False
```

```
>>> (6, 1) == (6, 1, 0)
```

```
False
```

Comparación de tuplas

Para determinar si una tupla es menor que otra, se utiliza lo que se denomina orden lexicográfico. Si los elementos en la primera posición de ambas tuplas son distintos, ellos determinan el ordenamiento de las tuplas:

```
>>> (1, 4, 7) < (2, 0, 0, 1)
```

```
True
```

```
>>> (1, 9, 10) < (0, 5)
```

```
False
```

La primera comparación es True porque $1 < 2$. La segunda comparación es False porque $1 > 0$. No importa el valor que tengan los siguientes valores, o si una tupla tiene más elementos que la otra.

Comparación de tuplas

Si los elementos en la primera posición son iguales, entonces se usa el valor siguiente para hacer la comparación:

```
>>> (6, 1, 8) < (6, 2, 8)
```

```
True
```

```
>>> (6, 1, 8) < (6, 0)
```

```
False
```

La primera comparación es True porque $6 == 6$ y $1 < 2$. La segunda comparación es False porque $6 == 6$ y $1 > 0$.

Comparación de tuplas

Si los elementos respectivos siguen siendo iguales, entonces se sigue probando con los siguientes uno por uno, hasta encontrar dos distintos. Si a una tupla se le acaban los elementos para comparar antes que a la otra, entonces es considerada menor que la otra:

```
>>> (1, 2) < (1, 2, 4)
```

```
True
```

```
>>> (1, 3) < (1, 2, 4)
```

```
False
```

La primera comparación es True porque $1 == 1$, $2 == 2$, y ahí se acaban los elementos de la primera tupla. La segunda comparación es False porque $1 == 1$ y $3 < 2$; en este caso sí se alcanza a determinar el resultado antes que se acaben los elementos de la primera tupla.

Comparación de tuplas

Este método de comparación es el mismo que se utiliza para poner palabras en orden alfabético (por ejemplo, en guías telefónicas y diccionarios):

```
>>> 'auto' < 'auxilio'
```

```
True
```

```
>>> 'auto' < 'autos'
```

```
True
```

```
>>> 'mes' < 'mesa' < 'mesadas' < 'mesas' < 'meses' < 'mi'
```

```
True
```

Usos típicos de las tuplas

Las tuplas se usan siempre que es necesario agrupar valores. Generalmente, conceptos del mundo real son representados como tuplas que agrupan información sobre ellos. Por ejemplo, un partido de fútbol se puede representar como una tupla de los equipos que lo juegan:

```
partido1 = ('Milan', 'Bayern')
```

Usos típicos de las tuplas

Para representar puntos en el plano, se puede usar tuplas de dos elementos (x, y). Por ejemplo, podemos crear una función distancia que recibe dos puntos y entrega la distancia entre ellos:

```
def distancia(p1, p2):  
    x1, y1 = p1  
    x2, y2 = p2  
    dx = x2 - x1  
    dy = y2 - y1  
    return (dx ** 2 + dy ** 2) ** 0.5
```

Al llamar a la función, se le debe pasar dos tuplas:

```
>>> a = (2, 3)  
>>> b = (7, 15)  
>>> distancia(a, b)
```

```
13.0
```


Usos típicos de las tuplas

Las fechas generalmente se representan como tuplas agrupando el año, el mes y el día. La ventaja de hacerlo en este orden (el año primero) es que las operaciones relacionales permiten saber en qué orden ocurrieron las fechas:

```
>>> hoy = (2011, 4, 19)
```

```
>>> ayer = (2011, 4, 18)
```

```
>>> navidad = (2011, 12, 25)
```

```
>>> anno_nuevo = (2012, 1, 1)
```

```
>>> hoy < ayer
```

```
False
```

```
>>> hoy < navidad < anno_nuevo
```

```
True
```

Usos típicos de las tuplas

Una tupla puede contener otras tuplas. Por ejemplo, una persona puede ser descrita por su nombre, su rut y su fecha de nacimiento:

```
persona = ('Perico Los Palotes', '12345678-9', (1980, 5, 14))
```

En este caso, los datos se pueden desempaquetar así:

```
>>> nombre, rut, (a, m, d) = persona
```

```
>>> m
```

```
5
```

Usos típicos de las tuplas

A veces a uno le interesa sólo uno de los valores de la tupla. Para evitar crear variables innecesarias, se suele asignar estos valores a la variable `_`. Por ejemplo, si sólo nos interesa el mes en que nació la persona, podemos obtenerlo así:

```
>>> _, _, (_, mes, _) = persona
```

```
>>> mes
```

Usos típicos de las tuplas

Una tabla de datos generalmente se representa como una lista de tuplas. Por ejemplo, la información de los alumnos que están tomando un ramo puede ser representada así:

```
alumnos = [  
    ('Perico', 'Los Palotes', '201199001-5', 'Civil'),  
    ('Fulano', 'De Tal', '201199002-6', 'Electrica'),  
    ('Fulano', 'De Tal', '201199003-7', 'Mecanica'),  
]
```

Usos típicos de las tuplas

En este caso, se puede desempaquetar los valores automáticamente al recorrer la lista en un ciclo for:

```
for nombre, apellido, rol, carrera in alumnos:  
    print nombre, 'estudia', carrera
```

O, ya que el apellido y el rol no son usados:

```
for nombre, _, _, carrera in alumnos:  
    print(nombre, 'estudia', carrera)
```

En otros lenguajes, las tuplas reciben el nombre de registros. Este nombre es común, por lo que conviene conocerlo

Iteración sobre tuplas

Al igual que las listas, las tuplas son iterables:

```
for valor in (6, 1):  
    print valor ** 2
```

Además, se puede convertir una tupla en una lista usando la función `list`, y una lista en una tupla usando la función `tuple`:

```
>>> a = (1, 2, 3)  
>>> b = [4, 5, 6]  
>>> list(a)  
[1, 2, 3]  
>>> tuple(b)  
(4, 5, 6)
```

Ejercicios con tuplas

Ejercicio 1: Cartas como tuplas. Proponer una representación con tuplas para las cartas de la baraja francesa.

Escribir una función poker que reciba cinco cartas de la baraja francesa e informe (devuelva el valor lógico correspondiente) si esas cartas forman o no un poker (es decir que hay 4 cartas con el mismo número).

Ejercicio 2. El tiempo como tuplas. Proponer una representación con tuplas para representar el tiempo.

Escribir una función sumaTiempo que reciba dos tiempos dados y devuelva su suma.

Ejercicio 3. Escribir una función diaSiguienteE que dada una fecha expresada como la terna (Día, Mes, Año) (donde Día, Mes y Año son números enteros) calcule el día siguiente al dado, en el mismo formato.

Ejercicio 4 Escribir una función diaSiguienteT que dada una fecha expresada como la terna (Día, Mes, Año) (donde Día y Año son números enteros, y Mes es el texto Ene, Feb, ..., Dic, según corresponda) calcule el día siguiente al dado, en el mismo formato.