

## 



## Hola:

Empezamos con la primera estructura de datos que vamos a ver esta semana. Las Tuplas nos van a permitir modelar entidades o elementos del mundo real del problema que queremos resolver

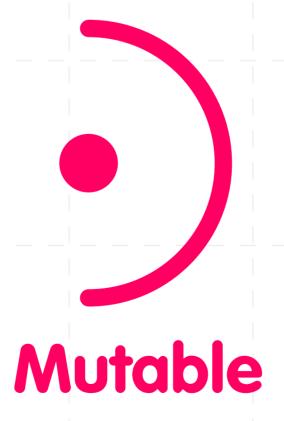
Vamos a dar un contexto a las estructuras de datos. El sitio en el cual nosotros estamos resolviendo un problema, o para el cual estamos realizando una aplicación, generalmente se conoce como el dominio del problema, el área del problema, o lo que sería el mundo real del problema. Ese mundo real es visto por dos protagonistas que ya conocemos: los Stakeholders (interesados) y los Developers (desarrolladores). Los Stakeholders ven esa realidad, ven el problema, ven las funcionalidades y las transmiten a los desarrolladores, estos, a su vez, convierten esas ideas de solución en algoritmos a los que le sumamos las estructuras de datos. Esto quiere decir que la representación de la información va de la mano en la solución de problemas, no solo son suficientes los algoritmos, ni tampoco son suficientes solo las estructuras de datos, necesitamos de ambos: algoritmos y estructuras de datos. Vamos a hacerle énfasis a una estructura de datos que llamamos Tuplas.

Antes hemos trabajado con estructuras de datos simples como las variables enteras y las variables reales; también hemos trabajado con unas estructuras de datos compuestas como son las cadenas y las listas. Además, hemos trabajado con estos dos tipos de estructuras de datos, los cuales sirven para representar un gran volumen y una gran cantidad de información. Vamos a ver de donde nace la necesidad de utilizar otras además de estas que ya tenemos, que son muy potentes, y nos han ayudado a solucionar problemas complejos.

Vamos a relacionar una palabra con un término que se llama inmutabilidad, eso quiere decir, no cambia. Cuando uno asigna un valor a una variable tipo cadena ya no puede cambiar los elementos que se encuentran allí. Podemos sacarle un clon, una copia, pero no podemos alterar la información que se almacena allí. A esas estructuras, una vez se le han asignado datos, los cuales no pueden ser cambiados, se llaman: estructuras de datos inmutables.



Ahora viene la contrapartida. Aquellas estructuras de datos que nos permiten cambiar su información, los datos que están allí almacenados, incluso su estructura en términos del número de elementos, se denominan: mutables, es decir, pueden cambiar. Algunas de las estructura de datos nuevas que vamos a ver esta semana serán mutables y otras serán inmutables. No olvidemos que las Strings son inmutables y las listas son mutables.





posX =[ 3,	4,	6,	8	]
posY =[ 2,	6,	4,	3	]
Radio =[ 2,	1,	1.5,	0.5	]
Color =[ "Cian"	, "Green",	"Chocolate",	"blue"	]

¿Qué pasa cuando se necesita representar un elemento del mundo real, así sea un elemento muy simple? Pensemos que esos círculos que están allí están en un videojuego y están ubicados en una pantalla de cierta manera o que están rebotando. ¿Qué información necesito de esos círculos que tienen una posición en el eje X y en el eje Y? Sabemos que tienen un radio porque son círculos y que tienen un color, por analogía y por facilidad podemos asumir que la representación de uno solo de estos círculos necesita de un valor de cada una de las listas.

Estamos diciendo que en la posición cero (0) de cada una de estas listas tengo la posicion en X, la posición en Y, el radio y el color, entonces, esa información corresponde o representa a los diferentes círculos que tenemos allí como se muestra en el siguiente diagrama:





¿Lo podemos representar con las estructuras que tenemos? Si, no hay problema. También es cierto que la manipulación se vuelve un poco compleja y truculenta porque lo que tenemos es que la información es independiente de cada uno de esos elementos que están allí, están separadas en cuatro (4) listas diferentes, pero aún así, los podemos manejar. Ahora, es posible también hacerlo de otra manera:



círculos=[[3,2,2,"Cian"], [4,6,1,"Green"], [6,4,1.5,"Chocolate"], [8,3,0.5"Blue"]]

Esta forma también es válida. Puedo tener una lista de listas, por ejemplo: la lista que se llama "círculos", y ahí tengo una lista que tiene; posición en X, posición en Y, radio y color, lo mismo para la circunferencia verde, la chocolate y la azul, entonces, puedo utilizar cualquiera de esos dos elementos para agrupar la información de las figuras que aparecen allí. Cada una tendrá sus ventajas y sus desventajas, y puede ser más claro entender la información porque está agrupada en elementos, puede ser que sea más fácil adicionar elementos en la parte superior o en la anterior, pero puedo manipularla. Sin embargo, estas dos estructuras que utilizamos aquí para representar una serie de círculos en ese videojuego, tienen un gran problema, y es que son mutables.

Pensemos en que se nos pueda borrar alguna de esas listas, o que haya un desplazamiento hacia la izquierda o hacia la derecha de cualquiera de las listas en la posición uno (1) o en la posición dos (2). Además, podríamos no ingresar una lista de cuatro (4) elementos, sino de dos (2) o tres (3), eso enloquecería absolutamente al sistema, porque son mutables.

Podemos agrupar información y ser solucionado con listas, pero el control y el manejo se podría complicar y mantener la coherencia de esta información que corresponde a cada elemento, se volvería muy complejo, porque son mutables, y pueden cambiar. Entonces, allí ya aparece un concepto, que es el de Tuplas, que es muy similar al de listas.

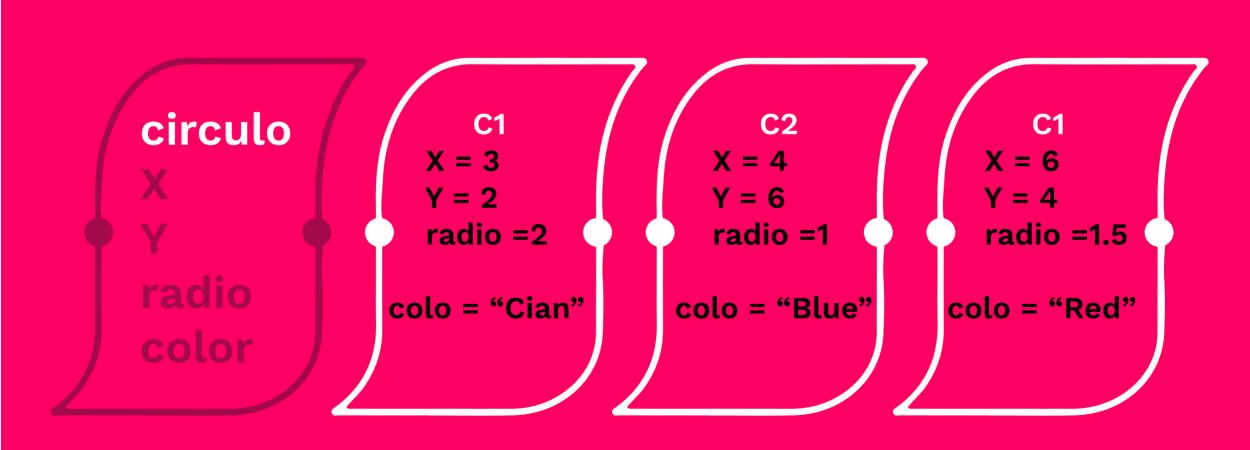
Una Tupla es una secuencia o colección de valores, del mismo o de diferente tipo, y se diferencian de las listas por su inmutabilidad, es decir, una vez le asignamos valores a una lista, no los podemos cambiar, ni podemos cambiar su valor, ni el número de elementos que tiene, lo que ayudará más a controlar la información que se maneja de una estructura de datos compuesta, como la que tenemos que hacer aquí. Esto es muy importante tenerlo en cuenta, porque en el mundo real, generalmente los números no van solos en la lista, ni los colores, sino que encontramos entidades con mucha información, como por ejemplo:

>>>circulo_1 = (3,	2,	2,	"Cian" )
>>>circulo_2 = (4,	6,	1,	"Green")
>>>circulo_3 = (6,	4,	1.5,	"Chocolate")
>>>circulo_4 = (8,	3,	0.5,	"Blue" )

El círculo uno (1) es: (3, 2, 2, cían"), lo que nos está diciendo es: la posición en X, en Y, el radio y el color, es decir, tenemos unas características de un círculo, lo que sí debemos identificar es: cada una de esas Tuplas que aparecen allí, tiene el mismo número de elementos, y que el elemento que está en cada posición, corresponde con el mismo en cada una. Aquí ya tenemos cuatro (4) elementos identificados.

¿Qué diferencia tienen las Tuplas con las listas? Además de la mutabilidad e inmutabilidad, es que en la sintaxis cambiamos los corchetes por paréntesis e incluso, podríamos definir Tuplas sin paréntesis. Esta característica de la inmutabilidad hace de gran utilidad a las Tuplas, pero aún más, es esa visión de que una Tupla, nos presenta varias características de una entidad del mundo real, por ejemplo: los sensores de una casa inteligente, los empleados de una empresa, es decir, que realmente las entidades que quiero-manejar-en-un sistema, no-solamente se representan por una información, sino por varias, ejemplo: un estudiante no solo tiene la edad, tiene número de documento, nombres, fecha de nacimiento, correo electrónico.

Cuando deseamos agrupar todo los elementos que conceptualmente tienen una cohesión entre ellos las Tuplas son una buena versión.



¿Qué otras ventajas tienen las Tuplas? Existe un subconjunto que se llama: "Tuplas con nombre". Esto indica que podemos crear una Tupla que se llame "círculo", estoy diciendo que ese círculo tiene X, Y, radio y color, es decir, ya estoy dando las características de ese círculo. Mientras que las otras el c1, c2 y c3, lo que hacen es darle valores a esas características, es decir, c1, c2 y c3, son círculos con valores predefinidos y específicos.

De esta manera podemos definir que c2 por ejemplo, es el círculo que está en la posición 4, 6 de radio uno (1) y color azul, así que, esas Tuplas que nosotros les asignamos un nombre, se llaman namedtuple o Tuplas con nombre, y para utilizarlas, debemos importar una librería que se llama Collections, específicamente importo la que necesito.

```
from collections import namedtuple

Circulo = namedtuple ( 'Circulo', 'x y radio color')

pt1 = Circulo (3, 2, 2, "Cian")

pt2 = Circulo (4, 6, 1, "Blue")

print (pt1.x)
```

estamos diciendo que círculo es una Tupla con nombre, que se llama círculo, y que tiene X, Y, radio y color. Aquí estoy creando esa Tupla, le estoy dando un nombre, ¿Qué tiene de interesante? Que ya puedo decir a PT1: asígnele círculo de posición X 3, Y 2, de radio dos (2), y color cían. nosotros definiendo Tuplas podemos darle un nombre, unas características al nombre círculo y después puedo crear elementos o entidades de esas Tuplas.

Esto es muy importante para lo que hemos visto, hemos visto estructuras de datos simples, que representan un solo valor; estructura de datos compuestas, que lo que hacen es permitirnos una lista de elementos, pero de una manera única. Aquí ya hemos dado un gran paso con estas Tuplas con nombres, porque estamos dándole una relación aislando estructuralmente una identidad, en este caso círculos, pero pensémoslo con personas, con estudiantes, con un sensor, con una cartografía, con lo que queramos ver modelado. Este es un ejemplo básico que nos vala ser de mucha utilidad manejarlo en el reto de esta semana y en alguno de los talleres que vamos a ver.

En conclusión hemos visto que existen unos datos compuestos como son las Tuplas que son de la familia de las listas, pero que tiene unas características adicionales como su inmutabilidad. Además, vimos una excelente estructura de datos llamada Tuplas con nombre que nos van a permitir agrupar en un nombre general a un conjunto de características de un grupo de objetos. — —

## Muchas gracias.

## 77.000 - TIG20

400000



