



PROGRAMACIÓN 2

Guía 5

DAVID VALENCIA SANDOVAL
INGENIERO EN SISTEMAS
david.valencia@ciaf.edu.co

Listas, Tuplas y Range

Vamos a aprender a crear listas en Python y no estoy hablando de las listas del mercado ni de los invitados.



Aunque su funcionalidad nos permita realizar una!. Las Listas y Tuplas en python se refieren a una zona de almacenamiento contiguo donde podemos guardar numerosos elementos.

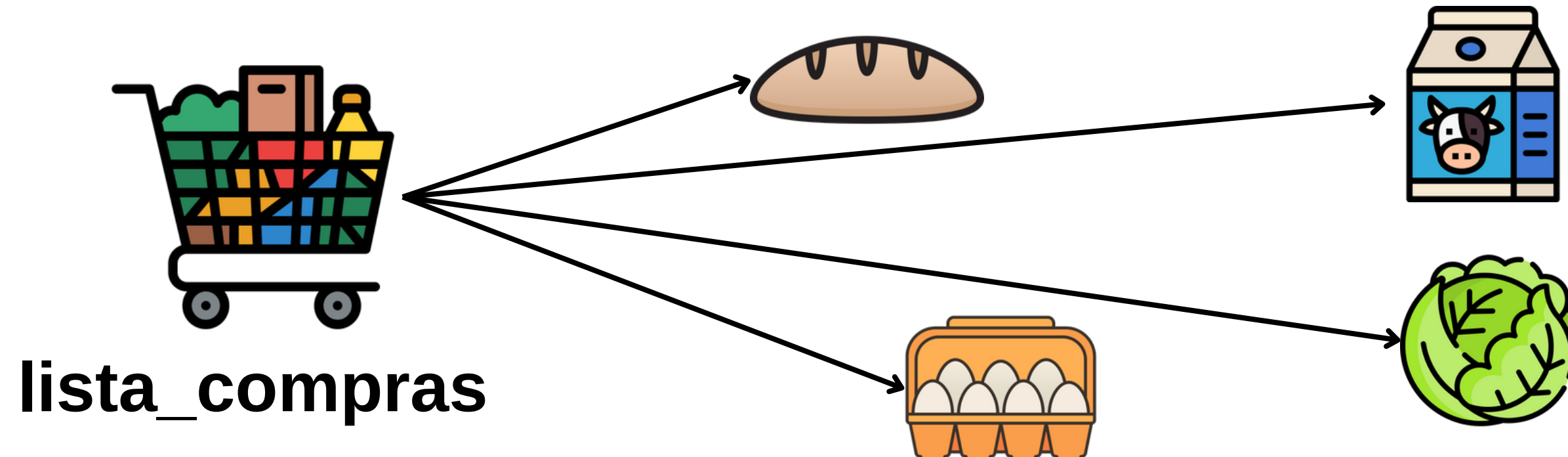
“Simple y sencillo es como si almacenamos varios datos dentro de una misma variable”

Pueden ser también **mutables** o **inmutables** (en el caso de las tuplas).

Listas en python

Como dijimos anteriormente podemos definir una lista como una zona de almacenamiento contigua donde podemos almacenar numerosos y diferentes tipos de datos, la misma puede editarse, modificarse mediante métodos siendo mutable.

Por lo que decimos también que las listas son datos estructurados almacenados en una variable. Un conjunto de datos que pueden ser simples o estructurados agrupados dentro de una variable a la que Python le asigna el tipo lista.



Listas en python

Sintaxis de Listas[]

Para crear un lista simplemente lo hacemos como si declaramos una variable pero almacenando en ella dentro de corchetes los elementos o ítems:

```
lista_compras = ["leche", "huevos", "pan", "lechuga"]  
numeros_loteria = [2114, 0728, 0965, 4932, 4114, 3666]
```

Como vemos mas arriba creamos dos listas «lista_compras» y «numeros_loteria», la primera contiene elementos de tipo String (texto). Por lo tanto deben ir entre comillas y separados por coma.

Listas en python

```
lista_compras = ["leche", "huevos", "pan", "lechuga"]  
numeros_loteria = [2114, 0728, 0965, 4932, 4114, 3666]
```

Pero también pueden ser creadas sin especificar los elementos que contienen, porque al ser mutables los podemos agregar después. **No es así en el caso de las tuplas que obligadamente debemos especificar al momento de crearla cuáles serán los elementos que vamos a almacenar en ella!**

Para crear una vacía lo hacemos de igual modo pero sin elementos, siempre respetando la sintaxis y colocando los debidos «Corchetes»

```
estudiantes = []
```

Listas en python

```
lista_compras = ["leche", "huevos", "pan", "lechuga"]  
numeros_loteria = [2114, 0728, 0965, 4932, 4114, 3666]
```

Pero también pueden ser creadas sin especificar los elementos que contienen, porque al ser mutables los podemos agregar después. **No es así en el caso de las tuplas que obligadamente debemos especificar al momento de crearla cuáles serán los elementos que vamos a almacenar en ella!**

Para crear una vacía lo hacemos de igual modo pero sin elementos, siempre respetando la sintaxis y colocando los debidos «Corchetes»

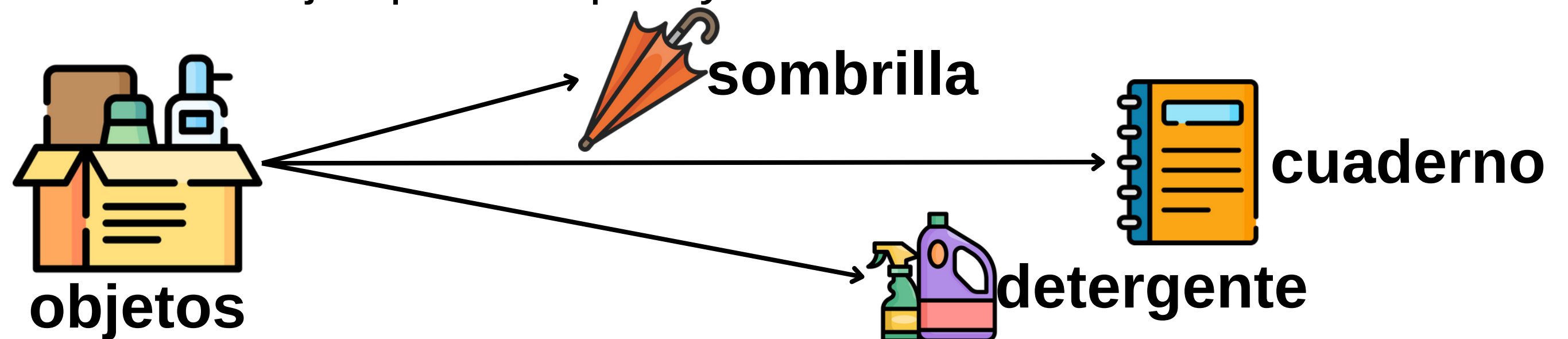
```
estudiantes = []
```

Tuplas en python

Una Tupla es la igual que una lista, una zona de almacenamiento contigua donde podemos almacenar diferentes y numerosos elementos. Pero con algunas similitudes y diferencias muy marcadas que siempre debes recordar.

Las tuplas son variables donde se almacenan diferentes tipos de datos estructurados que deben estar ordenados desde el comienzo.

Las tuplas son inmutables, no se pueden modificar durante la ejecución del programa. Por ende al crearla debes colocar si o si los elementos que se van a almacenar. Veamos ahora un ejemplo de tuplas y su sintaxis:



Tuplas en python

Sintaxis de Tuplas():

Para crear una tupla simplemente lo hacemos como si declaramos una variable pero almacenando en ella dentro de paréntesis los elementos o ítems:

```
objetos = ("sombrilla", "cuaderno", "detergente")
```

```
numeros_loteria = (187, 215, 363, 447, 559)
```

Como vemos más arriba creamos dos Tuplas «objetos» y «numeros_loteria», la primera contiene elementos de tipo String (texto). Por lo tanto deben ir entre comillas y separados por coma.

De esta forma hemos creados dos tuplas!

Recuerda que en las tuplas debes obligadamente especificar al momento de crearla cuáles son los elementos que vamos a almacenar!



Diferencias y similitudes entre listas y tuplas:

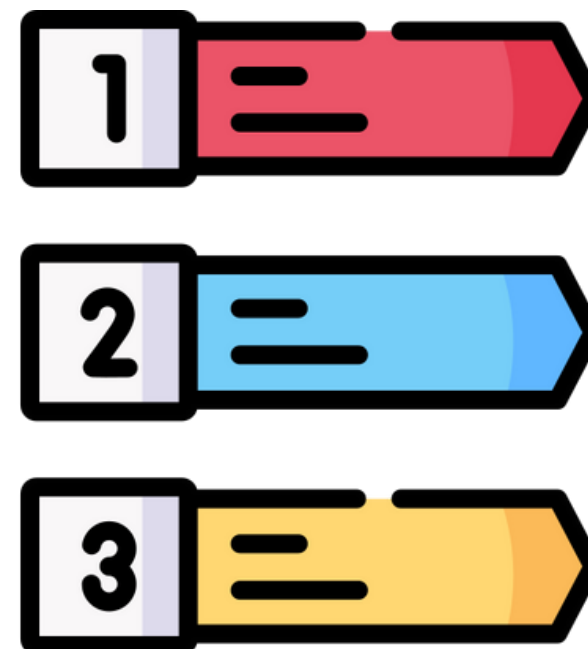
<u>Listas en Python</u>	<u>Tuplas en Python</u>
Son Dinámicas	Son Estáticas
Utilizamos Corchetes []	Utilizamos Paréntesis ()
Elementos separados por coma	Elementos separados por coma
List = [1, 2]	Tupla = (1, 2)
Acceso a elementos [Índice]	Acceso a elementos [Índice]

Accediendo a Listas y Tuplas:

¿Cómo accedemos a estos elementos que almacenamos dentro de ellas?

Aquí es donde comprenderás su utilidad, para acceder a los elementos debemos hacerlo teniendo en cuenta que a cada elemento almacenado se le asigna un Índice.

Un índice es un número que identifica el elemento dentro del grupo de elementos o ítems, siempre comenzando por cero.



Accediendo a Listas y Tuplas:

Aunque modifiquemos los elementos siempre el primero será el cero y a partir de allí irá en aumento.

A la lista que creamos anteriormente le corresponden los siguientes:

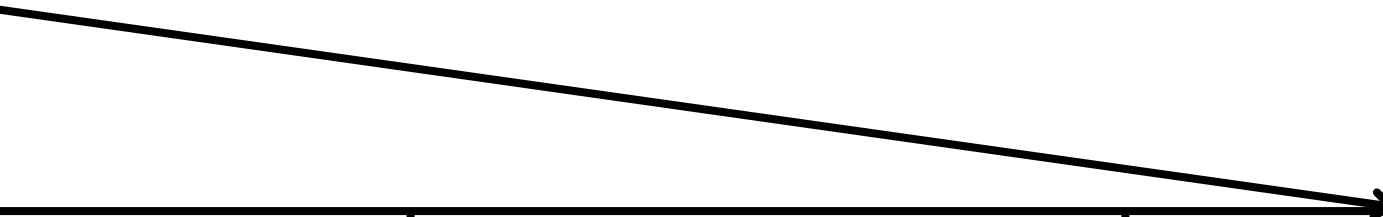
```
lista_compras = ["leche", "huevos", "pan", "lechuga"]
```

lista_compras =	["leche,	"huevos"	"pan"	"lechuga"]
Indice	0	1	2	3

Accediendo a Listas y Tuplas:

Entonces simplemente si yo quiero acceder al objeto «**pan**» de la Lista «**lista_compras**» debo especificar la lista y seguido entre «Corchetes», el número del índice que le corresponde:

lista_compras[2]



lista_compras =	["leche,	"huevos"	"pan"	"lechuga"]
Indice	0	1	2	3

Y esto debería mostrar el objeto «pan» correspondiente al Índice 2.



Actividad

Investigar que hace la función **slice** y aplíquelo en un ejemplo



Métodos de listas en Python (modificarlas)

Vamos a ver como añadir o quitar elementos mediante los métodos **append**, **insert**, **remove**!! Porque por supuesto puedes añadir todos los elementos que quieras en el código fuente. Pero en este caso vamos a agregar código para que cambie mientras nuestro programa se ejecuta.

Vamos a comenzar!! Supongamos que luego de crear una lista queremos imprimirla y luego modificar elementos con una instrucción para volver a imprimirla ¿Como lo hacemos?



¿Qué es un Método?

Un método puede ser una serie de sentencias para llevar a cabo una acción, pero en este caso podemos ver al método como el pedido a un objeto para que realice una tarea o en este caso un cambio!

Método Append e Insert (Añadir Elementos)

Append (Que se traduce Adjuntar en español)

Sirve para agregar un elemento al final de una lista en python.

Queremos agregar el objeto «Azucar» a la lista y como no nos importa el orden de los elementos utilizamos el método append que lo ubicara a lo último. Vamos a imprimirla completa antes y después.

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.append("Azucar")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```


Método Append e Insert (Añadir Elementos)

Método Extend (Que se traduce Ampliar en español)

Es útil para agregar **varios elementos de un iterable al final de una lista o también añadir una nueva lista** en python. Usando el **método extend** nos permite extenderla, veamos el ejemplo:

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.extend("Carne")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

Método Extend (Que se traduce Ampliar en español)

```
Lista1 = [1, 2, 3]  
Lista2 = [4, 5, 6]  
Lista1.extend(Lista2) #Añadimos la lista2 como extensión de Lista1  
print (Lista1)
```

```
[1, 2, 3, 4, 5, 6]
```

Método Append e Insert (Añadir Elementos)

Método Insert (Que se traduce Insertar en español)

Para añadir un elemento en otra parte podemos usar el método insert al igual que lo hicimos anteriormente: Primero indicamos el nombre de ella y seguido de un punto el método (Insert en este caso) y luego entre paréntesis indicamos el índice y seguido de una coma el elemento. Así:

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.insert(1, "Arroz")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

Método Insert (Que se traduce Insertar en español)

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.insert(1, "Arroz")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

```
Esta es tu lista de compra:  ['Leche', 'Huevos', 'Pan', 'Lechuga']  
Esta es tu nueva lista de compra:  ['Leche', 'Arroz', 'Huevos', 'Pan', 'Lechuga']
```

Métodos para eliminar elementos de una lista

Método Pop (Que se traduce al español como pequeña explosión ^^)

Al igual que los demás solo debemos ingresar el nombre, seguido del punto y el nombre del método (pop) y entonces entre paréntesis indicamos el índice del elemento a eliminar.

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.pop(0)  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

Método Pop (Que se traduce al español como pequeña explosión ^^)

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.pop(0)  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

```
Esta es tu lista de compra: ['Leche', 'Huevos', 'Pan', 'Lechuga']  
Esta es tu nueva lista de compra: ['Huevos', 'Pan', 'Lechuga']
```

Métodos para eliminar elementos de una lista

Método Remove (Que se traduce como remover)

El método Remove nos va a servir para eliminar un elemento de listas en python llamándolo por su valor, o por su índice, aunque a veces teniendo una larga sería molesto ponerse a contar palabras.

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.remove("Huevos")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

Método Remove (Que se traduce como remover)

```
lista_compras = ["Leche", "Huevos", "Pan", "Lechuga"]  
print("Esta es tu lista de compra: ", lista_compras)  
  
lista_compras.remove("Huevos")  
  
print("Esta es tu nueva lista de compra: ", lista_compras)
```

```
Esta es tu lista de compra:  ['Leche', 'Huevos', 'Pan', 'Lechuga']  
Esta es tu nueva lista de compra:  ['Leche', 'Pan', 'Lechuga']
```