

# Programación Básica Clase 09

# Agenda

Arreglos unidimensionales

- Concepto y utilización
- Creación
- Escritura y lectura
- Ejemplos





# Arreglos unidimensionales

Imagínate una situación en la que se necesita guardar canciones en un reproductor de música, utilicemos variables para almacenar los nombres.

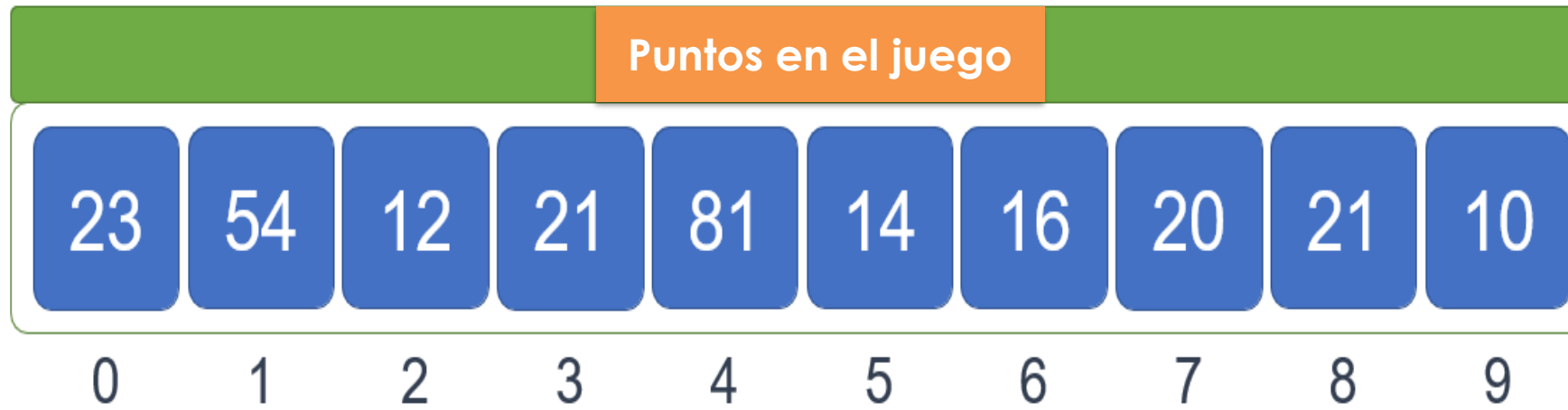


Requerimos solo 5 canciones y  
¿si fueran 50?



# Arreglos unidimensionales

Una estructura que permita almacenar múltiples valores de manera simultánea.



Cada valor es reconocido por su posición, iniciando el conteo en 0



En Python podemos crear un arreglo con sus valores ya predefinidos y hacer referencia a cada uno de ellos por medio de su posición.

**Arreglo con valores predefinidos**

```
edades = [23, 54, 12, 21, 81, 14, 16, 20, 21, 10]  
print(edades[1])  
print(edades[1] - 22)  
print(edades[edades[2]-3])
```

**Muestra el valor en la segunda posición**

**Aplica una operación sobre el valor almacenado**

**Utiliza un valor de otra posición, con una operación para identificar la posición que se desea mostrar**



También podemos asignar un nuevo valor a una de estas posiciones, con lo perderemos el valor almacenado anteriormente.

```
edades = [23, 54, 12, 21, 81, 14, 16, 20, 21, 10]
```

```
print(edades[1])
```

```
edades[1] = 100
```

```
print(edades[1])
```

Muestra el valor de la posición  
1 o sea 54

Almacena el número 100 en la  
posición 1

Imprime el contenido de la  
posición 1 en este caso un 100



**¿Qué sucede si asignamos  
un valor en una posición  
que no existe?**



**R/**

**Da un error y la ejecución  
del programa se detiene.**



En pocas ocasiones definiremos los arreglos y no se modificarán, será necesario agregar nuevos valores de la siguiente manera.

```
puntosJuego=[23,54,12,21,81,14,16,20,21,10]  
for x in (puntosJuego):  
    print(x)
```





**También se puede hacer con la función range.**



La función range devuelve una serie de números que van desde cero hasta el numero anterior al especificado entre paréntesis.



## Ejemplo:

```
puntosJuego=[23,54,12,21,81,14,16,20,21,10]  
for x in range(10):  
    print(puntosJuego[x])
```



**Si no conocemos la cantidad de elementos del arreglo podríamos utilizar la función `len()` para conocerlo.**



La función `len()` retorna un valor entero que corresponde a la cantidad de posiciones del arreglo.



También podríamos tomar en cada iteración del ciclo directamente el elemento que se encuentra en el arreglo, colocándolo como la fuente de los valores de la siguiente manera.

```
edades = [23, 54, 12, 21, 81, 14, 16, 20, 21, 10]
for x in edades:
    print(x)
```

Cualquiera de los dos métodos es válido, pero este nos permite tomar directamente los valores del arreglo en la definición del ciclo



Si quisiéramos agregar datos de manera vacía a un arreglo lo podríamos hacer de la siguiente manera.

```
puntosJuego={ }  
for x in range(0,10):  
    valor=int(input("Digite la puntuación obtenida:"))  
    puntosJuego[x]=valor  
for x in range(0,len(puntosJuego)):  
    print(puntosJuego[x],end=" ")
```



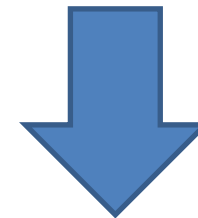


# Modificar un valor en un arreglo

Supongamos que tenemos un arreglo vacío donde almacenaremos las calificaciones de los cinco mejores estudiantes en un curso de Programación con el fin de nombrarlos estudiantes destacados al final del cuatrimestre.

Procedemos de la siguiente forma:

```
calificaciones=[0,0,0,0,0]  
calificaciones[0]=80  
calificaciones[1]=93  
calificaciones[2]=90  
calificaciones[3]=92  
calificaciones[4]=100
```



Supongamos que ahora requerimos modificar las calificaciones de los estudiantes almacenados en las posiciones 2 y 4, pues fueron ingresadas con un valor incorrecto. Realizamos lo siguiente:

```
calificaciones[2]=91  
calificaciones[4]=98
```



# 1 Ejercicio



Eduardo es un joven ciclista, cada semana debe reportar a su entrenador la cantidad de kilómetros recorridos en sus prácticas. Haga un programa que le permita almacenar para cada día los kilómetros recorridos y luego al final de la semana muestre por cada día, junto con el total de la semana. Para la solución, utilice arreglos y ciclos.

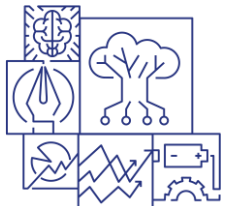
Tiempo aproximado 20 minutos



## 2 Ejercicio



En una sala de teatro se requiere almacenar los nombres de las personas que ocuparán las butacas de una fila, cada fila tiene 10 butacas. Cree un programa que almacena los nombres en las posiciones que le van indicando, por ejemplo: Ana Jiménez, 4 (el cuatro indica el número de butaca).



**Tiempo aproximado 20 minutos**



# 3 Ejercicio



Se requiere un programa que imprima una palabra al revés, debe funcionar para cualquier palabra, por lo cual debe utilizar instrucciones de lectura. En la solución utilice ciclos.

Tiempo aproximado 20 minutos





## 4 Ejercicio



Jimena es una joven que trabaja para pagarse sus estudios. Se ha inscrito en una de las plataformas de entrega de comida y otros. Haga un programa que le permita a Jimena registrar el monto ganado en cada día, al finalizar la semana le mostrará el total y le indicará el día que ganó menos dinero.

Tiempo aproximado 40 minutos





**¡Nos vemos la próxima semana!**

