



## Clase 6 – Listas + for avanzado + Turtle con colores

**Módulo:** Repeticiones y Turtle

### Fundamentación

Las listas permiten almacenar múltiples valores en una sola variable, y combinarlas con ciclos for abre la posibilidad de crear programas más flexibles y dinámicos. Integrar Turtle con listas de colores hace que los estudiantes vean cómo la estructura de datos influye en el resultado visual, reforzando la lógica y fomentando la creatividad.

### Objetivo general

- Comprender el uso de listas en Python y aplicar ciclos for para recorrerlas, utilizando sus elementos en dibujos con Turtle.

### Objetivos particulares

1. Declarar y manipular listas en Python.
2. Recorrer listas con for para acceder a sus elementos.
3. Usar los valores de una lista para cambiar atributos de Turtle (colores).
4. Crear figuras visuales que combinen secuencias de colores y formas.

### Contenidos

- **Conceptuales**
  - Definición de lista: sintaxis y ejemplos.
  - Acceso a elementos por índice.
  - Recorrido de listas con for.
- **Procedimentales**
  - Crear y modificar listas.
  - Iterar sobre listas para aplicar colores a figuras.
  - Dibujar patrones usando múltiples colores.



## PYTHON INICIAL

---

- **Actitudinales**

- Experimentación con paletas personalizadas.
- Valoración del trabajo visual como producto de la lógica.

### Iniciamos con la siguiente pregunta:

“¿Cómo guardarías varios colores para usarlos sin tener que escribirlos uno por uno cada vez?”

¿Qué es una lista?

Una lista es una colección ordenada de elementos que se pueden **modificar**.

Siguiendo el uso de colores de la clase 5, podemos crear una lista colores.

```
colores = ["red", "blue", "green"]  
print(colores[0]) # Muestra 'red'
```

**El primer elemento tiene índice 0.**

### Operaciones básicas con listas

Agregar un elemento:

```
colores.append("black")  
print(colores)
```

resultado:

```
['red', 'blue', 'green', 'black']
```

Eliminar elementos:

```
colores.remove("blue")  
print(colores)
```

resultado:

```
['red', 'green']
```

Contar elementos:

```
print(len(colores))
```

Verificar si un elemento esta en la lista:

```
print("black" in colores)
```

Cambiar un elemento:

```
colores[1] = "white"  
print(colores)
```



## PYTHON INICIAL

---

### Introducción a listas + for sobre listas

```
colores = ["red", "blue", "green", "purple"]
```

```
for color in colores:  
    print("El color es:", color)
```

- for recorre directamente cada elemento.

### 3. Ejemplo con Turtle – Cuadrados de colores

**Va a dibujar un cuadrado donde cada lado tiene un color diferente: rojo, azul, verde y púrpura.**

```
import turtle as t
```

```
colores = ["red", "blue", "green", "purple"]
```

```
for color in colores:  
    t.color(color)  
    t.forward(100)  
    t.right(90)
```

```
t.done()
```

#### Cómo funciona paso a paso:

1. `import turtle as t` → Importa la librería turtle y la llama t para escribir menos.
2. `colores = [...]` → Lista de 4 colores.
3. `for color in colores:` → Recorre cada color de la lista.
4. `t.color(color)` → Cambia el color de la línea.
5. `t.forward(100)` → Avanza 100 píxeles.
6. `t.right(90)` → Gira 90° a la derecha.
7. `t.done()` → Mantiene abierta la ventana de dibujo.

- Cambiar lista y observar efecto.

### 4. Ejercicio – Flor simple con colores

**Ese código dibuja seis círculos de colores diferentes, girando 60° cada vez, de forma que todos se superponen en el centro formando una especie de flor o mandala.**



## PYTHON INICIAL

---

```
import turtle as t

t.speed(0)
colores = ["red", "blue", "green", "orange", "purple", "pink"]

for color in colores:
    t.color(color)
    t.circle(50)
    t.right(60)

t.done()
```

- Cada giro genera un pétalo.

### Paso a paso:

1. `t.speed(0)` → La velocidad máxima de dibujo, para que se vea instantáneo.
2. Lista colores → 6 colores distintos.
3. `for color in colores:` → Recorre cada color.
4. `t.color(color)` → Cambia el color del trazo.
5. `t.circle(50)` → Dibuja un círculo con radio de 50 píxeles.
6. `t.right(60)` → Gira 60°, de forma que 6 giros completan 360°.
7. `t.done()` → Mantiene la ventana abierta.

### 5. Desafío en parejas

- Crear una figura libre usando:
  - Una lista de al menos 5 colores.
  - `for` para recorrerla.
  - Combinación de figuras (`circle`, `forward`, `right`).