



PROGRAMACIÓN DE COMPUTADORAS III

Abdel G. Martínez L.

CONCEPTO Y DEFINICIÓN

- ⦿ Un ciclo es una secuencia de instrucciones que se van a repetir continuamente hasta que se cumpla cierta condición
- ⦿ Debemos evitar los ciclos infinitos, los cuales son ciclos que carecen de una condición válida de salida
- ⦿ Cada pase a través del ciclo se conoce como iteración
- ⦿ Existe varios tipos de ciclos: controlados, rangos numéricos y el basado en un iterador.

CATEGORÍAS

⦿ *While*

- Ejecuta una sentencia hasta que se cumpla una condición

```
>>> x = 0
```

```
>>> while x < 3:
```

```
...     print("Sigo en el ciclo")
```

```
...     x = x + 1
```

```
...
```

```
>>> print("Hasta luego")
```

CATEGORÍAS

⦿ *While-Else*

- Las sentencias del `else` se ejecuta después de salir del ciclo

```
>>> x = 0
```

```
>>> while x < 3:
```

```
...     print("Sigo en el ciclo")
```

```
...     x = x + 1
```

```
... else:
```

```
...     print("Hasta luego")
```

CATEGORÍAS

⦿ *For*

- Ejecuta una sentencia un número específico de veces

```
>>> languages = ["C", "C++", "Perl", "Python"]
```

```
>>> for l in languages:
```

```
...     print(l)
```

```
...
```

```
>>>
```

CATEGORÍAS

◉ *For*

- Puede utilizar la función `range()` para manejarse por rangos

```
>>> for x in range(5):
```

```
...     print(x)
```

```
...
```

```
>>>
```

CATEGORÍAS

⦿ *For-Else*

- Las sentencias del `else` se ejecuta después de salir del ciclo

```
>>> languages = ["C", "C++", "Perl", "Python"]
```

```
>>> for l in languages:
```

```
...     print(l)
```

```
... else:
```

```
...     print("Se acabaron")
```

```
...
```

CONSIDERACIONES ESPECIALES

- ⦿ La sentencia `break` rompe el ciclo en ejecución
- ⦿ La sentencia `pass` no hace ninguna acción
- ⦿ La sentencia `else` no se ejecuta luego de un `break`