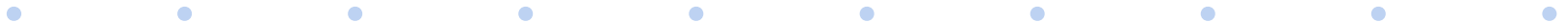


# Condicionales



# Clase 4 - Temario

Repaso Clase Anterior

Condicionales



# Clase 4 - Bibliografía

**Aprender a Pensar como un  
Programador**

Capítulo 4 (Hasta 4.8 Inclusive)



# Repaso Clase Anterior



# Estructuras de un programa



# Estructura de un programa

**Sentencia:** orden ejecutable que compone un programa.

**Flujo de ejecución:** forma en que se ejecutan las sentencias.

**Estructuras de control:**

Son el medio por el cual los programadores pueden determinar el flujo de ejecución en un programa

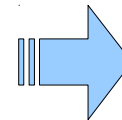
-

# Secuencias de instrucciones

Si se debe calcular el sueldo con el 20% de aumento de un empleado...

Con el intérprete

```
antigüedad = input ('Ingrese su antigüedad en el trabajo: ')\nsueldo= input('Ingrese su sueldo ')\nsueldo= sueldo + sueldo * 0.20\nprint ('Su sueldo con el aumento es: ', sueldo)
```



El flujo de ejecución es “**secuencial**”.  
Se ejecuta una sentencia y luego la siguiente

....Pero .. ¿Si el aumento hubiese sido SOLO para los que tenían 10 o más años de antigüedad?

Se debería haber usado una estructura de control que permitiese testear si la antigüedad era mayor o igual que 10

# Estructura de control

Y.. ¿si se tendría que calcular el sueldo de 100 empleados?

¿Repetiríamos 100 veces el siguiente conjunto de sentencias?

```
antiguedad = input ('Ingrese su antiguedad en el trabajo: ')\nsueldo= input('Ingrese su sueldo ')\nsueldo= sueldo + sueldo * 0.20\nprint ('Su sueldo con el aumento es: ', sueldo)
```

**NO!!!!**

Se debería haber usado una estructura de control que permitiese REPETIR 100 veces esas 4 sentencias..



# Estructura de control

Las estructuras de control sirven para que el programador pueda cambiar y manejar el flujo de control de acuerdo a ciertas condiciones.

## Condicionales:

- if
- if .. else
- if .. elif.. elif.. else
- A if C else B

## Bucles:

- while
- for .. in

## Decisiones ....

**Sentencias condicionales:** Permiten comprobar condiciones y que el programa se comporte de una manera u otra, de acuerdo a esa condición.

```
if  
If .. else  
If .. elif.. elif.. else
```

## Las condiciones...

Las “condiciones” se evalúan. Esa evaluación pueden arrojar dos posibles valores:

“Verdadero” o

“Falso”

¿A qué tipo de dato pertenecen esos valores?

Al tipo de dato **Boolean**

**Valores booleanos:** True y False

**Operadores lógicos:** and, or, not....

# Condiciones y resultados....

Ejemplos:

$a=(3>1) \rightarrow \text{print } a \rightarrow \text{True}$

$b=(3>3) \rightarrow \text{print } b \rightarrow \text{False}$

**Operadores lógicos:**

<b>and</b>	True	False
True	<b>True</b>	False
False	False	False

<b>or</b>	True	False
True	True	True
False	True	<b>False</b>

## Sentencia if

**Sentencia if:** Sentencia condicional más simple. Permite tomar decisiones sencillas.

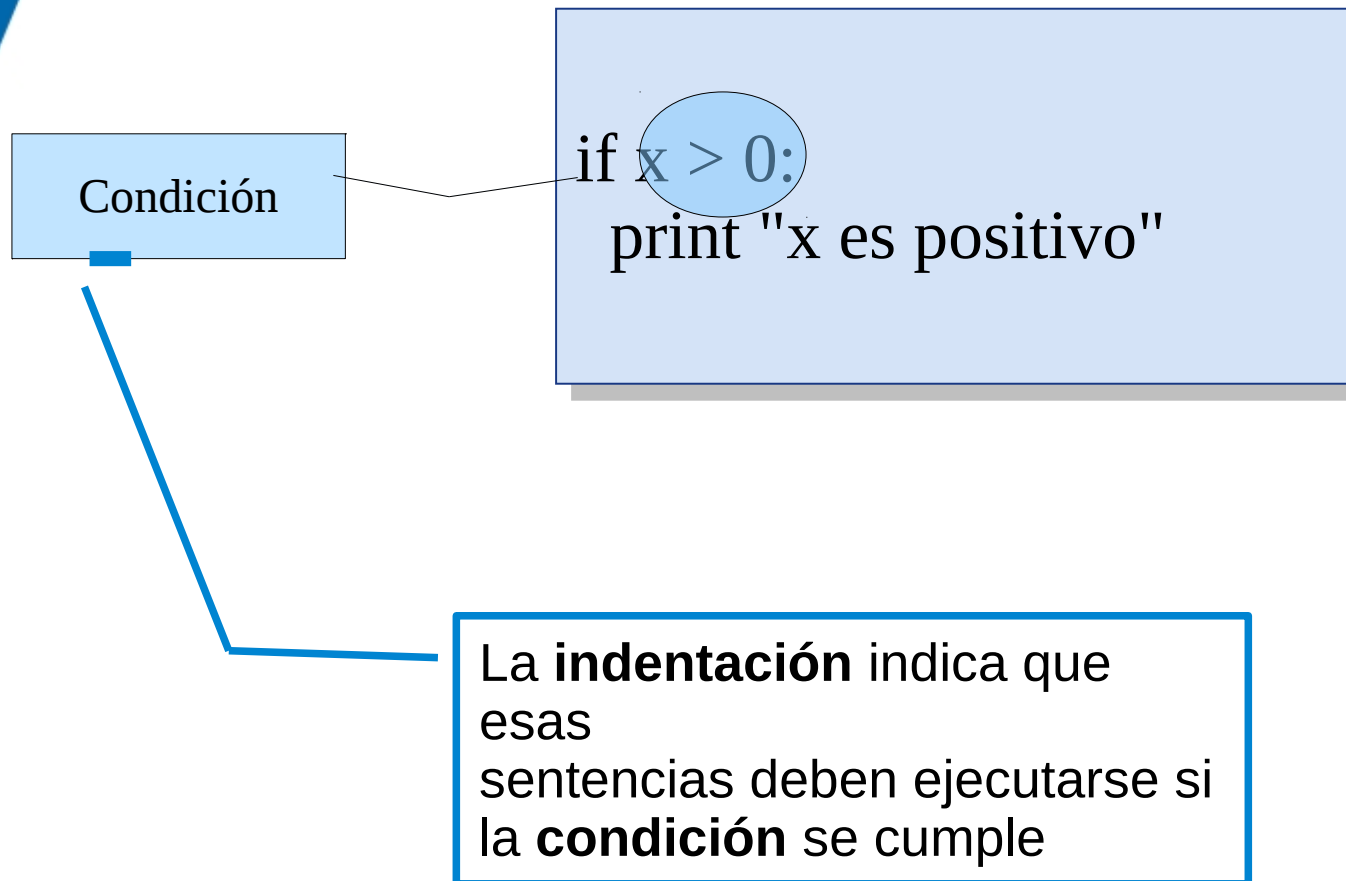
**Ejemplo:**

“Si un número es mayor que cero, entonces es positivo”

En Python:

```
if x > 0:  
    print "x es positivo"
```

# Sentencia if



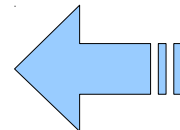
# Sentencia if

## Ejemplo

```
>>>  
>>>  
>>> dia="1 de enero"  
>>> if dia == "1 de enero":  
    print "¡Feliz Año Nuevo!!"
```

```
¡Feliz Año Nuevo!!  
>>> |
```

```
>>>  
>>> dia="2 de enero"  
>>> if dia == "1 de enero":  
    print "¡Feliz Año Nuevo!!"
```



¿Qué pasó aquí?

```
>>>  
>>> |
```

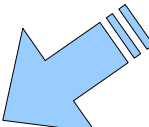
# Sentencia if

## Ejemplo

```
>>>  
>>> numero=1  
>>> if numero==1:
```

```
    print "Estamos mostrando una sentencia if con varias lineas"  
    print "En este caso, estas lineas se imprimen si la variable numero es 1"  
    print "Luego de imprimir estas lineas, pasmaos a la siguiente instruccion"
```

Atención a  
la indentación



```
Estamos mostrando una sentencia if con varias lineas  
En este caso, estas lineas se imprimen si la variable numero es 1  
Luego de imprimir estas lineas, pasmaos a la siguiente instruccion
```

```
>>>  
\\
```





# Decisiones ....

## Sentencia if-else

Permite establecer las acciones necesarias cuando la condición NO se cumple.

Ejemplo:

```
>>>
>>> dia="2 de enero"
>>> if dia == "1 de enero":
        print "¡Feliz Año nuevo!"
else:
        print "Hoy no es año nuevo..."
```

```
Hoy no es año nuevo...
```

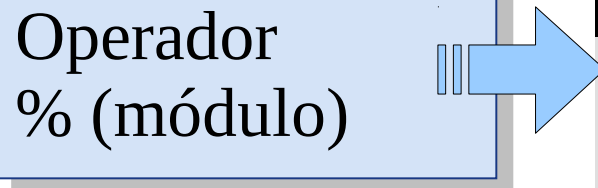
```
>>>
~~~
```

# Decisiones ....

## Sentencia if-else

Ejemplo: Algo más útil:

Operador  
% (módulo)



```
if x%2==0:  
    print x, "es par"  
else:  
    print x, "es impar"
```

# Decisiones ....

## Sentencia if-elif

¿Qué pasa cuando hay más de dos condiciones?

Ejemplo:

Aquí aparecen varias condiciones excluyentes.

Si la edad de una persona es menor que 2,  
entonces es un bebé,  
sino, si es mayor que 2 y menor que 13  
es un/a niño/a,  
Sino, si es mayor que 13 pero menor que 20,  
es adolescente,  
Sino, es adulto.

# Decisiones ....

## Sentencia if-elif

En Python:

```
if edad > 0 and edad < 2:  
    print "Bebe"  
elif edad > 2 and edad < 13:  
    print "Niño/a"  
elif edad > 13 and edad < 20:  
    print "Adolescente"  
else:  
    print "Adulto"
```

Acá hay cosas que  
no están bien

No están cubiertos todos los  
posibles valores de la variable  
edad.

# Decisiones ....

## Sentencia if-elif

Ahora bien....

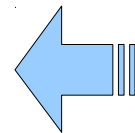
```
if edad >= 0 and edad < 2:  
    print "Bebe"  
elif edad >= 2 and edad < 13:  
    print "Niño/a"  
elif edad >= 13 and edad < 20:  
    print "Adolescente"  
else:  
    print "Adulto"
```

# Decisiones ....

## Sentencia if-elif

Otro ejemplo:

```
if ruedas <2:  
    print "Monopatin"  
elif ruedas <3:  
    print "Moto"  
elif ruedas<4:  
    print "Triciclo"  
elif ruedas <5:  
    print "Auto"  
else:  
    print "Demasiadas ruedas"
```



Acá hay cosas  
que no están bien

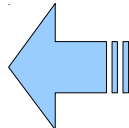
La condiciones no son  
excluyentes.

# Decisiones ....

## Sentencia if-elif

Otro ejemplo: Ahora bien escrito

```
if ruedas ==1:
    print "Monopatin"
elif ruedas ==2:
    print "Moto"
elif ruedas==3:
    print "Triciclo"
elif ruedas ==4:
    print "Auto"
elif ruedas >4:
    print "Demasiadas ruedas"
else:
    print "No tenemos ruedas"
```



¿A qué valores  
referencia el else?