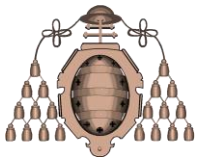


Otras instrucciones para bucles

`"break" y "continue"`



Otras instrucciones para bucles

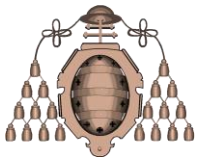
Hay un par de instrucciones que permiten controlar el flujo de control en los bucles. Son "**break**" y "**continue**".

"**break**" permite finalizar el bucle (salir de él) inmediatamente.

"**continue**" permite volver inmediatamente al inicio del bucle para realizar el siguiente ciclo del mismo.

Las dos pueden usarse tanto en los bucles "**for**" como en los bucles "**while**". En general "**continue**" se usa menos.

Casi siempre van ambas dentro de un condicional interno al bucle y existen también en otros lenguajes como Java o C++.



La instrucción "break"

El siguiente ejemplo ilustra el uso de la primera:

```
for i in range(5):  
    if i==3:  
        break  
    print(i)
```

La salida de este programa es 0, 1 y 2 uno en cada línea. Al ejecutarse la instrucción "**break**" el bucle se interrumpe y finaliza en vez de seguir con los valores 3 y 4.

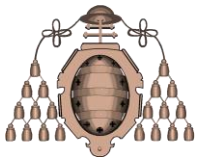


La instrucción "continue"

El siguiente ejemplo ilustra el uso de la primera:

```
for i in range(5):  
    if i==3:  
        continue  
    print(i)
```

La salida de este programa es 0, 1, 2 y 4 uno en cada línea. Al ejecutarse la instrucción "**continue**" cuando "**i**" vale 3, no se ejecuta el resto del bucle en ese caso y se vuelve al principio para seguir con el siguiente valor de "**i**".



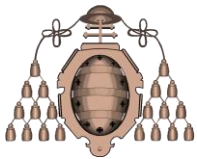
Un programa para ver si un número es primo

```
# se pide el número entero por teclado
n=int(input("número a comprobar"))

# suponemos inicialmente que es primo pero en cuanto
# encontremos un divisor exacto anotaremos que no lo es
es_primo=True

# bucle (poco eficiente por cierto) para buscar divisores exactos
for i in range(2,n):
    if n%i==0:
        es_primo=False
        # ya sabemos que no es primo luego no hace falta continuar
        break

# ya hemos salido del bucle bien porque acabó con su valor "i" superior
# o bien porque se ejecutó la instrucción "break"
if es_primo:
    print(n,"es primo")
else:
    print(n,"no es primo")
```



Para saber más

El "**exit()**" permite terminar la ejecución de un programa. Funciona como si se hubiera llegado al final del mismo y ya no hubiera más instrucciones para ejecutar. Se emplea cuando sabemos que se va a producir un error y que no debe continuar la ejecución del programa.

Ejemplo:

```
...  
# calculamos el cociente si se puede  
if divisor==0:  
    print("no se puede dividir por cero, fin del programa")  
    exit()  
# se puede calcular, prosigue el programa  
cociente=valor/divisor  
...
```