

Estructuras condicionales

```
n=input('Escribe un número: ')
n=eval(n)
if n%2==0:
    div2=n/2
    print('El número',n,'es divisible por 2.')
    print('Al dividir',n,'entre 2 se obtiene',div2)
elif n%3==0:
    div3=n/3
    print('El número',n,'es divisible por 3.')
    print('Al dividir',n,'entre 3 se obtiene',div3)
elif n%5==0:
    div5=n/5
    print('El número',n,'es divisible por 5.')
    print('Al dividir',n,'entre 5 se obtiene',div5)
```


Todos los lenguajes de programación disponen de estructuras que permiten evaluar expresiones cuyo resultado es True o False y dependiendo del resultado llevar a cabo un determinado bloque de instrucciones u otro.

A este tipo de estructuras se le llaman estructuras condicionales, evalúan una condición y cuando la condición resulta True ejecutan un determinado código y cuando resultan False pueden ejecutar otro bloque de código.

En Python estas estructuras vienen definidas con las palabras reservadas: **if**, **else** y **elif**.

Podemos encontrarnos con las siguientes construcciones:

Tipo 1

```
If condicion:
    instruccion1
    instruccion2
    ...
```

Tipo 2

```
If condicion:
    bloque1-instruccion1
    bloque1-instruccion2
    ...
else:
    bloque2-instruccion1
    bloque2-instruccion2
    ...
```

Tipo 3

```
If condicion1:
    bloque1-instruccion1
    bloque1-instruccion2
    ...
elif condicion2:
    bloque2-instruccion1
    bloque2-instruccion2
    ...
elif condicion3:
    bloque3-instruccion1
    bloque3-instruccion2
    ...
else:
    bloque4-instruccion1
    bloque4-instruccion2
    ...
```

En todas las estructuras anteriores, la condición representa una expresión cuyo resultado será True o False.

Vamos a poner algunos ejemplos. Supongamos que en un programa tenemos una variable numérica, podríamos hacer condiciones de las formas:

n<10 Esta expresión sólo dará como resultado True cuando n contenga un valor inferior a 10.

n>=1000 Esta expresión sólo dará como resultado True cuando n contenga un valor mayor o igual que 1000.

n>10 and n<100 Sólo resultará True cuando n esté comprendido entre 10 y 100.

n<1 or n>=5 Será True cuando n sea menor que 1 ó mayor o igual a 5

En la estructura **tipo 1**, el bloque de instrucciones sólo se ejecutará cuando el resultado de evaluar la condición sea True, si el resultado es False, no se ejecutará ningún código asociado a dicho caso.

Programa: condicionaltipo1.py

Vamos a hacer un programa que nos pida un número y a continuación nos diga si dicho número es múltiplo de 3 y nos muestre la división de ese número entre 3.

1º El programa mostrará unos mensajes informando lo que hace.

2º Solicitará un valor numérico.

3º Sólo si el valor introducido es múltiplo de 3, nos mostrará el resultado de dividir dicho número entre 3.

Veamos el código que hace esta funcionalidad. Como ya dijimos en ejemplos anteriores, existen varias formas de hacer el programa, nos limitaremos a hacerlo de una de ellas.

```

1  print('Programa para saber si un número es múltiplo de 3')
2  print('Si el número que escribes es divisible por 3, te mostraré')
3  print('el resultado de dividir dicho número entre 3.')
4  print('En caso contrario, no mostraré nada más.')
5  n = input('Escribe un número: ')
6  n = eval(n)
7  if n%3==0:
8      print('Enhorabuena, has escrito un número múltiplo de 3')
9      div = n/3
10     print('El resultado de dividir',n,'entre 3 es',div)

```

Nota: Existen dos formas de partir líneas:

1. **Dentro de un paréntesis**, como en el caso anterior, se podrá partir la línea (lo deseable después de una coma o un operador matemático) y **tabular** la continuación **a la derecha** (habitualmente cuatro espacios)
2. En líneas que no contienen paréntesis, lo hacemos igual que arriba, pero utilizando la **barra invertida ** al final de la primera línea

Observa la línea 7: te recuerdo que el operador % no calcula el resto de la dividir n entre 3 (en este caso).

Vamos a ver 2 posibles ejecuciones del programa, una introduciendo 21 (múltiplo de 3) y otra introduciendo 31 (que no es divisible por 3).

>>> %Run condicionaltipo1.py

```

Programa para saber si un número es múltiplo de 3
Si el número que escribes es divisible por 3, te mostraré
el resultado de dividir dicho número entre 3.
En caso contrario, no mostraré nada más.
Escribe un número: 21
Enhorabuena, has escrito un número múltiplo de 3
El resultado de dividir 21 entre 3 es 7.0

```

>>> |

>>> %Run condicionaltipo1.py

```

Programa para saber si un número es múltiplo de 3
Si el número que escribes es divisible por 3, te mostraré
el resultado de dividir dicho número entre 3.
En caso contrario, no mostraré nada más.
Escribe un número: 31

```

>>> |

En la estructura condicional **tipo 2**, el bloque 1 de instrucciones sólo se ejecutará si la condición resulta True, pero a diferencia de la estructura condicional tipo 1, en el caso de que dicha condición resulte False se ejecutaría el bloque 2 de instrucciones.

Programa: condicionaltipo2.py

Vamos a hacer un programa que nos pida un número y a continuación nos diga si dicho número es múltiplo de 3 y nos muestre la división de ese número entre 3.

- 1º El programa mostrará un mensaje informando lo que hace.
- 2º Solicitará un valor numérico.
- 3º Sólo si el valor introducido es múltiplo de 3, nos mostrará el resultado de dividir dicho número entre 3.
- 4º Si el valor introducido no es múltiplo de 3, el programa nos informará de ello y mostrará la división de dicho número entre 3.

Realizaremos algunas modificaciones más sobre el código anterior que debes comprender sin mayor dificultad

```

1  print('Programa para saber si un número es múltiplo de 3')
2  n = input('Escribe un número: ')
3  n = eval(n)
4  div = n/3

5  if n%3==0:
6      print('Enhorabuena, has escrito un número múltiplo de 3')
7      print('El resultado de dividir',n,'entre 3 es',div)
8  else:
6      print('El número que has escrito NO es múltiplo de 3')
7      print('Al dividir',n,'entre 3 se obtiene',div)
9      print('La división no es exacta.')
```

Vamos a ver nuevamente las 2 ejecuciones anteriores con este nuevo programa:

```
>>> %Run condicionaltipo2.py
```

```
Programa para saber si un número es múltiplo de 3
Escribe un número: 21
Enhorabuena, has escrito un número múltiplo de 3
El resultado de dividir 21 entre 3 es 7.0
```

```
>>> %Run condicionaltipo2.py
```

```
Programa para saber si un número es múltiplo de 3
Escribe un número: 31
El número que has escrito NO es múltiplo de 3
Al dividir 31 entre 3 se obtiene 10.333333333333334
La división no es exacta.
```

```
>>> |
```

```
>>> |
```

En el **tipo 3** de estructuras condicionales podemos comprobar que entran en juego **varias condiciones**, tantas como consideremos oportuno, pero el programa sólo ejecutará el bloque de instrucciones de la primera condición que resulte True, es decir:

Si la condición 1 resulta True, ejecutará el bloque 1 de instrucciones y no seguirá evaluando las siguientes condiciones. Sólo seguirá evaluando la condición 2 en el caso de que la condición 1 resulte False.

Si la condición 2 resulta True, ejecutará el bloque 2 de instrucciones y no seguirá evaluando las siguientes condiciones. Por lo tanto, la condición 3 sólo será evaluada en el caso de que las condiciones 1 y 2 resulten False.

Siguiendo el mismo razonamiento, el bloque 4 de instrucciones sólo será ejecutado en el caso de que todas las condiciones evaluadas con anterioridad resulten False.

Programa: condicionaltipo3.py

Vamos a hacer un programa que nos pida un número y a continuación nos diga si dicho número es múltiplo de 2, en cuyo caso mostrará la división entre 2. Sólo en el caso de que el número no sea múltiplo de 2, comprobará si es múltiplo de 3 y sólo en caso afirmativo, nos mostrará la división de ese número entre 3. Si el número tampoco es múltiplo de 3, el programa comprobará si es múltiplo de 5, en cuyo caso mostrará la división del número introducido entre 5.

En el caso de que no sea múltiplo de 2, ni de 3, ni de 5, nos informará de ello.

- 1º El programa mostrará unos mensajes informando lo que hace
- 2º Solicitará un valor numérico
- 3º Si el valor introducido es múltiplo de 2, nos mostrará el resultado de dividir dicho número entre 2.
- 4º Si el valor introducido no es múltiplo de 2, pero sí es múltiplo de 3, el programa nos informará de ello y mostrará la división de dicho número entre 3.
- 5º Sólo si el número introducido no es múltiplo de 2, ni de 3, pero sí lo es de 5, en dicho caso, nos mostrará el resultado de dividirlo entre 5.
- 6º Si el número introducido no es múltiplo de 2, ni de 3, ni de 5, el programa nos informará de dicha circunstancia.

Antes de continuar con este ejemplo, vamos a aclarar que el objetivo es mostrar el mecanismo de las estructuras condicionales. Si deseamos encontrar los divisores de un número, esta manera de hacerlo no es la adecuada, precisamente ese ejercicio de calcular los divisores de un número lo resolveremos en el capítulo siguiente cuando estudiemos los bucles.

Vamos a escribir el código de nuestro programa para comprobar el funcionamiento de las estructura condiciones cuando hay varias condiciones:

```

1  print('Este programa te informará si un número es divisible por 2')
2  print('Si no es divisible por 2, comprobará si lo es por 3')
3  print('Si tampoco es divisible por 3, comprobará si lo es por 5')
4  print('Si no es divisible por 2, ni por 3, ni por 5, informará de ello')
5  n=input('Escribe un número: ')
6  n=eval(n)
7  if n%2==0:
8      div2=n/2
9      print('El número',n,'es divisible por 2.')
10     print('Al dividir',n,'entre 2 se obtiene',div2)
11  elif n%3==0:
12     div3=n/3
13     print('El número',n,'es divisible por 3.')
14     print('Al dividir',n,'entre 3 se obtiene',div3)
15  elif n%5==0:
16     div5=n/5
17     print('El número',n,'es divisible por 5.')
18     print('Al dividir',n,'entre 5 se obtiene',div5)
19  else:
20     print('El número no es múltiplo de 2, ni de 3, ni de 5.')
21     print('Prueba otra vez con otro número.')
```

Veamos el resultado de algunas ejecuciones del programa:

```

>>> %Run condicionaltipo3.py

Este programa te informará si un número es divisible por 2
Si no es divisible por 2, comprobará si lo es por 3
Si tampoco es divisible por 3, comprobará si lo es por 5
Si no es divisible por 2, ni por 3, ni por 5, informará de ello
Escribe un número: 30
El número 30 es divisible por 2.
Al dividir 30 entre 2 se obtiene 15.0
```

```

>>> |
```

```
>>> %Run condicionaltipo3.py
```

```
Este programa te informará si un número es divisible por 2
Si no es divisible por 2, comprobará si lo es por 3
Si tampoco es divisible por 3, comprobará si lo es por 5
Si no es divisible por 2, ni por 3, ni por 5, informará de ello
Escribe un número: 45
El número 45 es divisible por 3.
Al dividir 45 entre 3 se obtiene 15.0
```

```
>>> |
```

```
>>> %Run condicionaltipo3.py
```

```
Este programa te informará si un número es divisible por 2
Si no es divisible por 2, comprobará si lo es por 3
Si tampoco es divisible por 3, comprobará si lo es por 5
Si no es divisible por 2, ni por 3, ni por 5, informará de ello
Escribe un número: 61
El número que has introducido no es múltiplo de 2, 3 o 5.
Prueba otra vez con otro número.
```

```
>>>
```

```
>>> %Run condicionaltipo3.py
```

```
Este programa te informará si un número es divisible por 2
Si no es divisible por 2, comprobará si lo es por 3
Si tampoco es divisible por 3, comprobará si lo es por 5
Si no es divisible por 2, ni por 3, ni por 5, informará de ello
Escribe un número: 125
El número 125 es divisible por 5.
Al dividir 125 entre 5 se obtiene 25.0
```

```
>>>
```

Es muy probable que se te haya ocurrido hacer un programa que nos diga si un número es divisible por 2, si también lo es por 3 y si también lo es por 5, o alguna variante del él.

Vamos a hacer exactamente eso, un programa que nos pedirá un número y después nos informará si el número es divisible por 2, si es divisible por 3 y si es divisible por 5. Para ello vamos a usar varias estructuras condicionales. Es muy fácil.

Programa: multiplos235.py

```

1  print('Este programa te dice si un número es múltiplo de 2, 3 o 5')
2  n=input('Escribe un número: ')
3  n=eval(n)
4
5  if n%2==0:
6      div2=n/2
7      print('El número',n,'es múltiplo de 2')
8      print('Al dividir',n,'entre 2 se obtiene',div2)
9  else:
10     print('El número', n, 'no es múltiplo de 2')
11
12  if n%3==0:
13     div3=n/3
14     print('El número',n,'es múltiplo de 3')
15     print('Al dividir',n,'entre 3 se obtiene',div3)
16  else:
17     print('El número', n, 'no es múltiplo de 3')
18
19  if n%5==0:
20     div5=n/5
21     print('El número',n,'es múltiplo de 5')
22     print('Al dividir',n,'entre 5 se obtiene',div5)
23  else:
24     print('El número', n, 'no es múltiplo de 5')

```

En este ejemplo hemos dejado algunas líneas vacías únicamente para que código resulte más fácil de leer, pero no es necesario dejar dichas líneas en blanco. De todas formas, cuando los programas crecen en tamaño, verás que es conveniente dejar líneas en blanco y añadir comentarios para comprender mejor el código en un futuro.

Veamos una posible salida del programa:

```
>>> %Run multiplo235.py
```

```

Este programa te dice si un número es múltiplo de 2, 3 o 5
Escribe un número: 12
El número 12 es múltiplo de 2
Al dividir 12 entre 2 se obtiene 6.0
El número 12 es múltiplo de 3
Al dividir 12 entre 3 se obtiene 4.0
El número 12 no es múltiplo de 5

```

```
>>>
```

Vamos a terminar este capítulo dedicado a las estructuras condicionales haciendo un último ejemplo en que vamos a **anidar** varias estructuras condicionales.

Anidar estructuras significa incluir unas estructuras dentro de otras.

Programa: anidarcondicionales.py

Vamos a hacer un programa que nos pida el nombre del usuario y compruebe la longitud en caracteres del nombre introducido. No dirá si el nombre es largo o se trata de un nombre corto y además en el caso de que el nombre sea largo, nos debe decir si el nombre es compuesto o no.

El programa mostrará un mensaje indicando lo que hace.
Solicitará el nombre del usuario.
Contará el número de caracteres del nombre.
Si el nombre tiene 7 caracteres o menos, sólo contará mostrará el numero de caracteres del nombre.
Si el nombre tiene más de 7 caracteres, comprobará además si contiene algún espacio.
En el caso de que contenga algún espacio dirá que el nombre es compuesto.
Si no contiene ningún espacio, dirá que el nombre no es compuesto.

El código del programa es el siguiente:

```
1 print('Este programa contará los caracteres de tu nombre')
2 nombre = input('Por favor, escribe tu nombre: ')
3 longitud = len(nombre)
4 print('Hola', nombre, 'encantado de saludarte')
5
6 if longitud<=7:
7     print('Tu nombre tiene', longitud, 'caracteres')
8 else:
9     print('Tienes un nombre largo,', longitud, 'caracteres')
10    if ' ' in nombre:
11        print('Además, tienes un nombre compuesto')
12    else:
13        print('Además, no se trata de un nombre compuesto')
```

Vamos a explicar algunas líneas:

En la línea 3, con la función **len()** contamos el número de caracteres que tiene la variable de texto nombre, y además asignamos dicho número a la variable **longitud**.

En la línea 6 comparamos la longitud con el número 7.

En la línea 10, comprobamos si el carácter **Espacio** (' ') se encuentra dentro de la variable nombre, para ello hemos usado la palabra clave **in** que hace exactamente eso.

Observa que el condicional de la línea 10, se encuentra dentro del else de la línea 8, en esto consiste la anidación de bloque de estructuras condicionales, en incluir unos bloques dentro de otros.

Ejercicios:

Ejercicio 1:

Escribe un programa que genere las siguientes salidas en su ejecución.

```
Programa sobre números pares
Escribe un número que sea par: 24
!!! CORRECTO !!!
El número 24 es un número par
24 / 2 = 12.0
```

```
Programa sobre números pares
Escribe un número que sea par: 31
!! Oh, lo siento !!
El número 31 no es un número par
31 / 2 = 15.5
```

Ejercicio 2:

Escribe un programa que genere las siguientes salidas en su ejecución.

```
Programa para comprobar si un número es
divisible o no por 2, 3, 5, 7 y 11
Escribe un número: 132
El número 132 SI es divisible por 2
El número 132 SI es divisible por 3
El número 132 NO es divisible por 5
El número 132 NO es divisible por 7
El número 132 SI es divisible por 11
```

Ejercicio 3:

El programa comprobará si la palabra tiene 5 letras o menos, en cuyo caso mostrará la primera y la última letra.

En el caso de que tenga más de 5 letras, mostrará las 4 primeras letras por un lado y las 4 últimas letras por otro, aunque haya letras coincidentes en las 2 particiones.

El programa debe generar las siguientes salidas en su ejecución.

```
Programa que analiza una palabra
Escribe una palabra: Hola
La palabra Hola tiene 4 letras
La primera letra es: H
La última letra es: a
```

```
Programa que analiza una palabra
Escribe una palabra: Saludos
La palabra Saludos tiene 7 letras
Las 4 primeras letras son: Salu
Las 4 últimas letras son: udos
```

Ejercicio 4:

Realizar un programa que calcule la letra del Número de Identificación Fiscal (N.I.F.).

El programa debe solicitar el número del Documento Nacional de Identidad y a partir de dicho número calcule la letra correspondiente.

Consultar la página: <http://www.interior.gob.es/web/servicios-al-ciudadano/dni/calculo-del-digito-de-control-del-nif-nie>

Nota: Sólo calcular el caso de NIF españoles.