

Control de flujo I: Estructuras de selección

Flujo del programa

- Llamamos flujo al recorrido del programa por las diferentes instrucciones.
- Hasta ahora solo hemos realizado programas de flujo secuencial, es decir, programas que ejecutan sus instrucciones de una en una.
- Es interesante que nuestro programa decida en función de los datos que maneja si ejecutar o no una instrucción y cuántas veces.

Necesidad de controlar el flujo

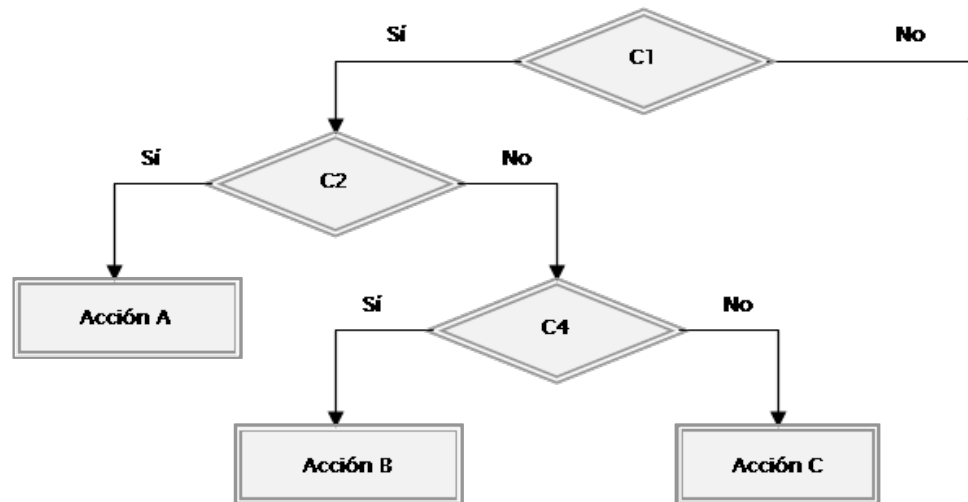
- El control del flujo del programa es útil porque nos permite desarrollar cualquier algoritmo.
- Un algoritmo es el conjunto de instrucciones, reglas y lógica que permiten resolver un problema. Saber programar no es más que crear algoritmos.

Teorema del programa estructurado

- Todo algoritmo* puede ser implementado usando únicamente 3 estructuras lógicas:
 - Secuencia.
 - Selección.
 - Iteración.
- * Siempre que sea computable, es decir, que tenga final.
- [Info](#) formas de representar el flujo.

Estructuras de selección

- Las estructuras de selección permiten tomar decisiones.
- Usando estructuras de selección se consigue que el flujo del programa forme árboles de decisión.



if

- En Python la única estructura de selección es `if`.
- `if` ejecuta el código a continuación cuando la expresión que le acompaña no se evalúa como `false`.
- La sintaxis es:

```
if expresión:  
    # código
```

Indentación

- En Python no hay símbolos para encerrar los bloques de código.
- En su lugar usamos la indentación (o según la RAE “sangrado” o “sangría” LOL) para definir qué líneas están en un mismo bloque. Uno de los errores más comunes al empezar a programar en Python es tener problemas con la indentación.
- Según el manual de estilo, se deben de dejar 4 espacios en blanco en cada indentación.

pass

- `pass` es una palabra reservada cuyo único uso es evitar errores mientras desarrollamos un programa.
- Previene el error de indentación cuando no queremos poner nada en un bloque de código. Después de una estructura de control Python espera encontrar un bloque indentado.
- Ej:

```
if a > 5:  
    pass
```


if-else-elif

- Podemos ampliar la estructura if con else y elif. Ambas son opcionales.
- Si la expresión tras if se evalúa como False, se evaluarán las expresiones elif en orden. Si ninguna de las expresiones se evalúan diferente a False entonces se ejecutará el bloque de código de else.
- Ej:

```
if a > b:
    print('Mayor')
elif a < b:
    print('Menor')
else:
    print('Igual')
```

Ejercicio

- En la UPCT existe una beca de comedor para comer gratis todos los días en la cantina. Sus condiciones son:
 - Tener más de un 8 de media en el expediente.
 - Cumplir una o más de estas condiciones: renta familiar <20.000€ al año, ser huérfano o tener una discapacidad de más del 75%.
 - Haber solicitado la beca MEC.

Elabora un formulario y determina si el usuario puede recibir la beca. Asegúrate de que la entrada del usuario tiene sentido.