Taller de Programación Web

Estructuras de Control Condicionales

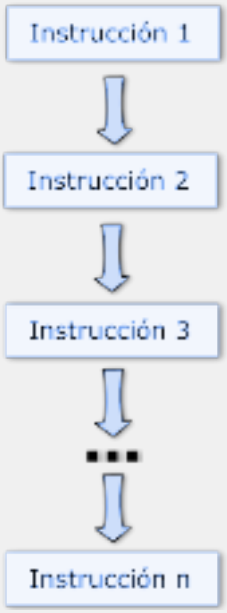
Subsecretaría de Empleo
Chaco Gobierno de todos

Ministerio de Producción, Industria y Empleo
Chaco Gobierno de todos

CHACO
Gobierno de todos

INFORMATORIO

INTRODUCCIÓN



Si un programa sólo ejecutara instrucciones planas unas tras otras no servirían de mucho.
Pues, el hecho que las acciones se ejecuten secuencialmente implica que nunca se ejecutara más de una acción al mismo tiempo

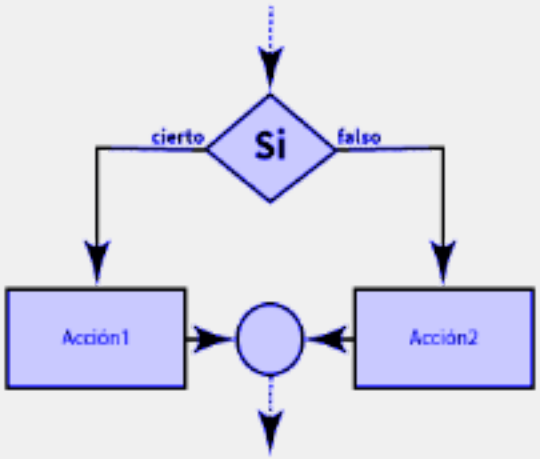
Por suerte ahí es donde aparecen las estructuras de control, las cuales van a permitir que *el flujo del programa se adapte y sepa cómo actuar ante determinadas situaciones e incluso repetir una tarea si es necesario*.

¿Qué tipos de estructuras veremos en el curso?

CONDICIONALES	REPETITIVAS

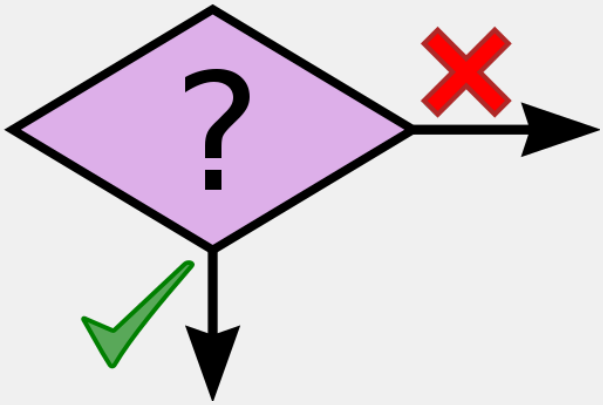
INFORMATORIO

Estructuras de Control Condicionales



No todos los problemas pueden resolverse empleando estructuras secuenciales. Cuando hay que tomar una decisión aparecen las estructuras condicionales.

De hecho, en



nuestra vida diaria se nos presentan situaciones donde debemos decidir.

- *¿Elijo la carrera A o la carrera B?*
- *¿Me pongo este pantalón?*
- *Para ir al trabajo, ¿elijo el camino A o el camino B?*
- *Hoy veamos una peli, ¿cuál la 1 o la 2?*

Cómo funciona?

En las estructuras condicionales, se realiza una evaluación de una condición y de acuerdo al resultado, el programa realiza una determinada acción.
Las condiciones son especificadas utilizando expresiones lógicas.

A partir de aquí te recomendamos ir probando las sentencias de ejemplo Para que te sea más fácil comprobar su funcionamiento

Condicional simple: Sentencia if (si)

La sentencia if se ejecuta siempre que la expresión que comprueba devuelve True:

Código

```
if True:  # equivale a if not False
    print("Se cumple la condición")
    print("También se muestre este print")
```

Resultado

```
Se cumple la condición
También se muestre este print
```

Puede ocurrir que necesites encadenar condicionales pues:

- Si se verifica una determinada condición, ejecutar una serie de instrucciones (bloque 1).
- Si no, esto es, si la condición NO se verifica, ejecutar otra serie de instrucciones (bloque 2).

Código

```
a = 5
if a == 2:
    print("a vale 2")
if a == 5:
    print("a vale 5")
```

Resultado

```
a vale 5
```

O también anidar If dentro de If:

Código

```
a = 5
b = 10
if a == 5:
    print("a vale",a)
    if b == 10:
        print("y b vale",b)
```

Resultado

```
a vale 5
y b vale 10
```

Como condición podemos evaluar múltiples expresiones, siempre que éstas devuelvan True o False:

Código

```
if a==5 and b == 10:
    print("a vale 5 y b vale 10")
```

Resultado

```
a vale 5 y b vale 10
```

Condicional alternativo o doble: Sentencia else (sino)

Se encadena a un If para comprobar el caso contrario (en el que no se cumple la condición):

Código

```
n = 11
if n % 2 == 0:
    print(n,"es un número par")
else:
    print(n,"es un número impar")
```

Resultado

```
11 es un número impar
```

Condicional multiple: Sentencia elif (sino si)

Con frecuencia es necesario que existan más de dos elecciones posibles.
Este problema se podría resolver por estructuras selectivas simples o alternativas, encadenadas. Pero si el número de alternativas es grande puede plantear serios problemas de escritura y de legibilidad.
Por este motivo, en estos casos, se recomienda el uso de la estructura condicional alternativo múltiple que permite *evaluar una variable que puede tomar de 1 a n valores* y según ocurra uno de esos valores, se realizará *una de las n acciones*; es decir, que el programa seguirá sólo un determinado camino entre varios.

En Python se encadena a un if u otro elif para comprobar múltiples condiciones, siempre que las anteriores no se ejecuten:

Código

```
nota = float(input("Ingresar una nota: "))

if nota >= 9:
    print("Sobresaliente")
elif nota >= 7:
    print("Notable")
elif nota >= 6:
    print("Bien")
elif nota >= 5:
    print("Suficiente")
else:
    print("Insuficiente")
```

Resultado

```
Introduce una nota: 10
Sobresaliente
```