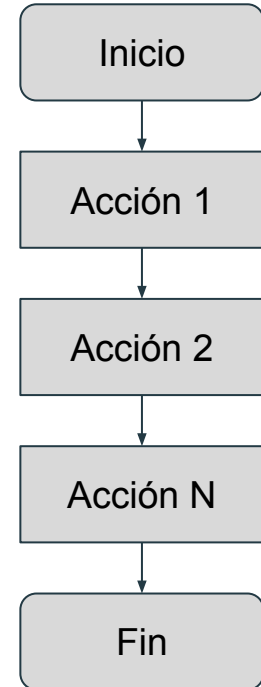




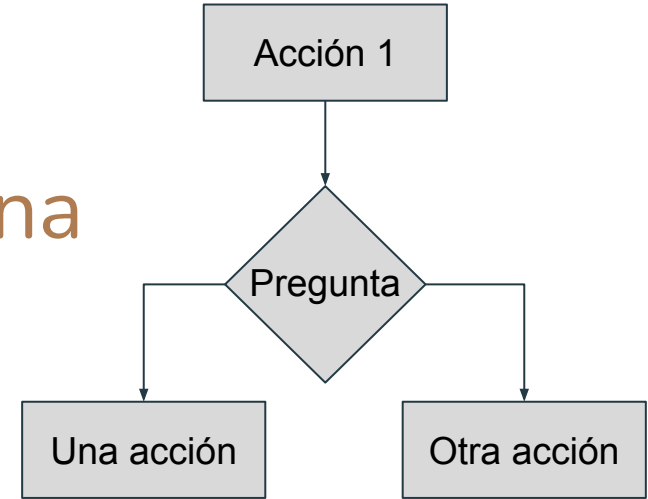
Estructuras de control

¿Cómo se ejecuta un programa?

- Un programa se ejecuta de manera secuencial
- Esto significa que primero se ejecuta la primera línea, después la segunda y así sucesivamente hasta llegar a la última



¿Qué pasa si quiero tomar una decisión?




Estructuras condicionales

- Ejecutan un bloque de código si se cumple una condición
- Se usan para tomar decisiones en tiempo de ejecución
- Vamos a ver dos diferentes: if y if-else

Estructuras condicionales: if

```
if ( condicion ):  
    accion1  
    accion2  
    accion3  
    .  
    .  
    .  
    accionN
```



La condición es una expresión lógica (verdadero o falso).
Si la condición es verdadera entonces se ejecuta el bloque.

Estructuras condicionales: if-else

```
if ( condicion ):  
    una_accion  
    ...  
else:  
    otra_accion  
    ...
```

Se comporta igual que el if pero si la condición es falsa entonces se ejecuta el bloque del else.

```
if ( condicion1 ):  
    accion1  
else if ( condicion2 ):  
    accion2  
else if ( condicion3 ):  
    accion3  
else:  
    accion4
```

Se pueden encadenar varios if-else con condiciones distintas

Estructuras condicionales: switch

```
1
2 switch( valor ) {
3     case 1:
4         accion_1;
5         ...
6         break;
7     case 2:
8         accion_2;
9         ...
10        break;
11    default:
12        accion_n;
13        ...
14        break;
15 }
16
```

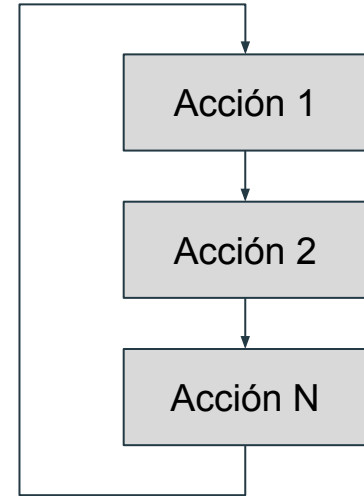
Este valor puede ser numérico o un carácter. Si el valor es igual a alguno de los casos entonces se ejecuta ese caso, si es diferente a todos los casos entonces se ejecuta el default.

No olvidarse de los break



¿Qué pasa si quiero ejecutar un
bloque de código varias veces?

¿Qué pasa si quiero ejecutar un bloque de acciones varias veces?

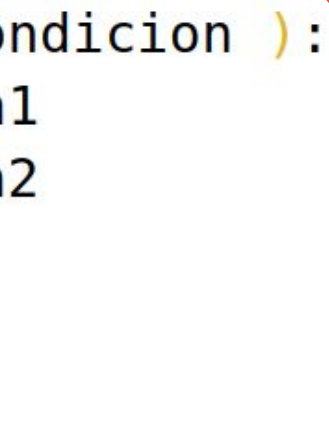


Estructuras iterativas

- Ejecuta un bloque de código varias veces
- Se comportan parecidas a las condicionales porque ejecuta el bloque siempre que se cumpla la condición
- Vamos a ver 2 diferentes: while y for

Estructuras iterativas: while

```
while ( condicion ):  
    accion1  
    accion2  
    .  
    .  
    .
```

A red curved arrow starts from a red dot at the bottom of the code block and points back to the 'while' keyword, illustrating the loop's repetition.

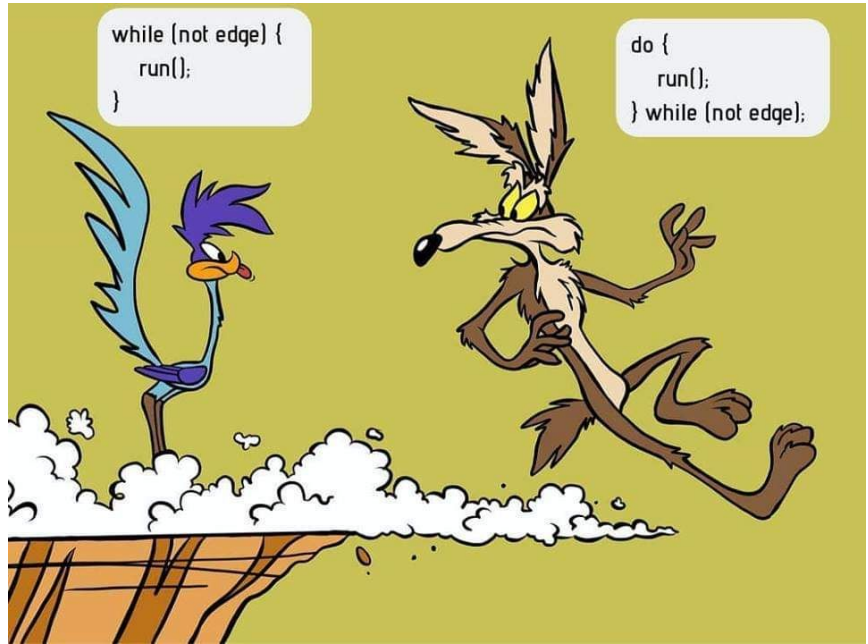
Mientras la condición sea verdadera se ejecuta el bloque. El ciclo no termina hasta que la condición deje de ser verdadera

No olvidarse que el valor de la condición debe cambiar en alguna iteración para no terminar con un ciclo infinito

Estructuras iterativas: do-while

```
1  
2 ▾ do{  
3     accion_1;  
4     accion_2;  
5     accion_3;  
6     ...  
7 } while( condicion );  
8
```

Funciona de la misma forma que el while pero ejecuta el bloque de código antes de chequear la condición



Estructuras iterativas: for

➔ Se ejecuta una vez antes de empezar con las iteraciones

➔ Se chequea antes de cada iteración de la misma forma que el while

```
1  
2 ▾ for(accion_unica ; condicion; accion_final){  
3     accion_1;  
4     accion_2;  
5     accion_3;  
6     ...  
7 }  
8
```

➔ Se ejecuta al final de cada iteración

Estructuras iterativas: for

```
for i in range( N ):
    accion1
    accion2
    .
    .
    .
```

Cuando queremos ejecutar una secuencia de acciones un número determinado de veces sin importar una condición entonces usamos la estructura for.



Cantidad de veces a ejecutar la secuencia

¿Cuántas veces se ejecuta la
accion_1?

```
1  
2 ▾ for(int i = 0; i < numero; i++){  
3     accion_1;  
4 }  
5
```

¿Cuándo se usa for y cuándo while?

- El for se usa siempre que sepamos exactamente cuántas veces vamos a iterar. Ya sea porque el valor es constante o porque está dentro de una variable a tiempo de ejecución
- El while se usa siempre que no sepamos de antemano cuántas veces hay que iterar