

# Sentencias de Selección

# Sentencias de Selección

Selección múltiple (`switch`)

# Sentencias de Selección

## Selección múltiple (switch)

Se usa en casos donde hay muchos posibles valores para una variable con distintas acciones para cada valor.

# Sentencias de Selección

## Selección múltiple (`switch`)

Se usa en casos donde hay muchos posibles valores para una variable con distintas acciones para cada valor.

Se usa en lugar de anidar múltiples `if-else`

# Sentencias de Selección

## Selección múltiple (`switch`)

Se usa en casos donde hay muchos posibles valores para una variable con distintas acciones para cada valor.

Se usa en lugar de anidar múltiples `if-else`

Solo se pueden comparar enteros (pueden ser caracteres pero no flotantes) y solo por igualdad (no relación)

# Sentencias de Selección

# Sentencias de Selección

## Sintaxis

```
switch (expresión) {  
    case valor1:  
        sentencia 1; // pueden ser muchas, no hace falta {}  
        break; // opcional  
    case valor2:  
        sentencia 2;  
        break; // opcional  
    // tantos case como se quiera, mientras sean diferentes  
    default:  
        sentencia n;  
        break; // opcional  
}
```

# Sentencias de Selección

# Sentencias de Selección

Se compara la expresión (puede ser una variable, operación, etc.) con las *etiquetas* en los **case**

# Sentencias de Selección

Se compara la expresión (puede ser una variable, operación, etc.) con las *etiquetas* en los **case**

Las etiquetas deben ser diferentes. Cuando haya una coincidencia, el flujo del programa salta hasta esa ubicación

# Sentencias de Selección

# Sentencias de Selección

```
i = 3;
switch (i) {
    case 1:
        printf("Primera opción\n");
        break;
    case 2:
        printf("Segunda opción\n");
        break;
    case 3:
        printf("Tercera opción\n");
        break;
    default:
        printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
  case 1:
    printf("Primera opción\n");
    break;
  case 2:
    printf("Segunda opción\n");
    break;
  case 3:
    printf("Tercera opción\n");
    break;
  default:
    printf("Ninguna opción\n");
}
```

# Sentencias de Selección

```
i = 3;
switch (i) {
    case 1:
        printf("Primera opción\n");
        break;
    case 2:
        printf("Segunda opción\n");
        break;
    case 3:
        printf("Tercera opción\n");
        break;
    default:
        printf("Ninguna opción\n");
}
```

# Sentencias de Selección

Selección múltiple (`switch`)

# Sentencias de Selección

## Selección múltiple (switch)

### Ejemplo con if-else

```
if (nota == 10) {  
    printf("A\n");  
} else {  
    if (nota == 9) {  
        printf("B\n");  
    } else {  
        if (nota == 8) {  
            printf("C\n");  
        } else {  
            if (nota == 7) {  
                printf("D\n");  
            } else {  
                printf("F\n");  
            }  
        }  
    }  
}
```

# Sentencias de Selección

## Selección múltiple (switch)

### Ejemplo con if-else

```
if (nota == 10) {
    printf("A\n");
} else {
    if (nota == 9) {
        printf("B\n");
    } else {
        if (nota == 8) {
            printf("C\n");
        } else {
            if (nota == 7) {
                printf("D\n");
            } else {
                printf("F\n");
            }
        }
    }
}
```

### Ejemplo con switch

```
switch (nota) {
    case 10:
        printf("A\n");
        break;
    case 9:
        printf("B\n");
        break;
    case 8:
        printf("C\n");
        break;
    case 7:
        printf("D\n");
        break;
    default:
        printf("F\n");
        break;
}
```

# Sentencias de repetición (o iterativas)

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

# Sentencias de repetición (o iterativas)

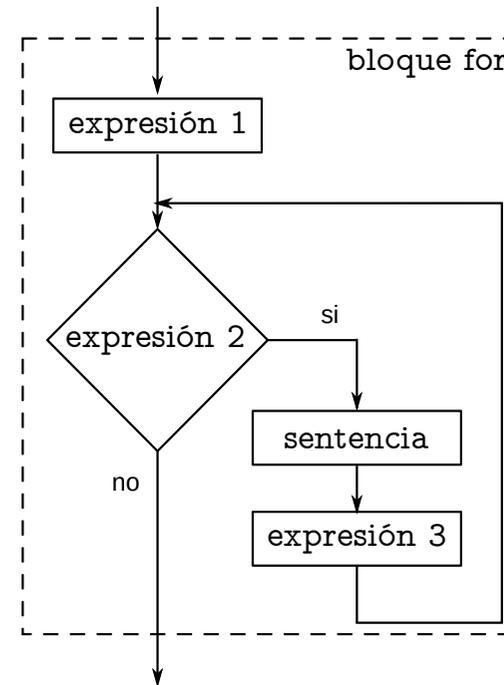
Sentencia repetitiva `for`

Tiene un funcionamiento semejante  
al `while` controlado por contador

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

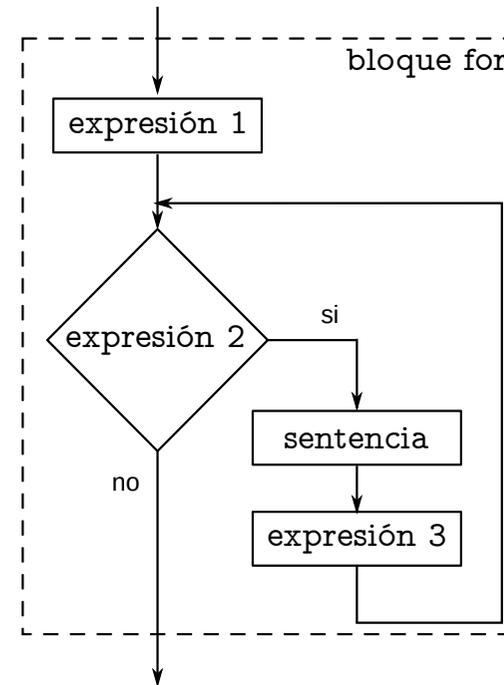


# Sentencias de repetición (o iterativas)

## Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

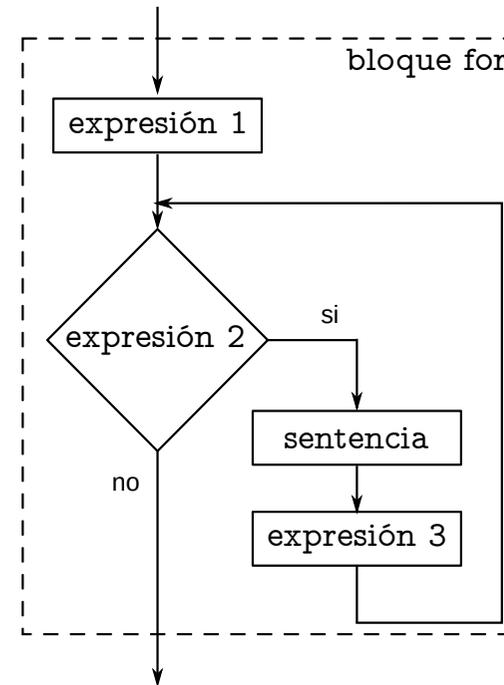
La expresión 1 se ejecuta una vez, al principio



# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

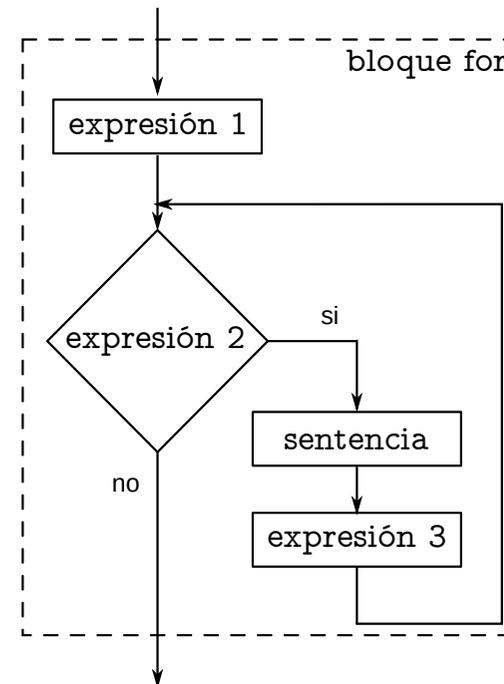


# Sentencias de repetición (o iterativas)

## Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

La expresión 2 sirve de condición para repetir el ciclo



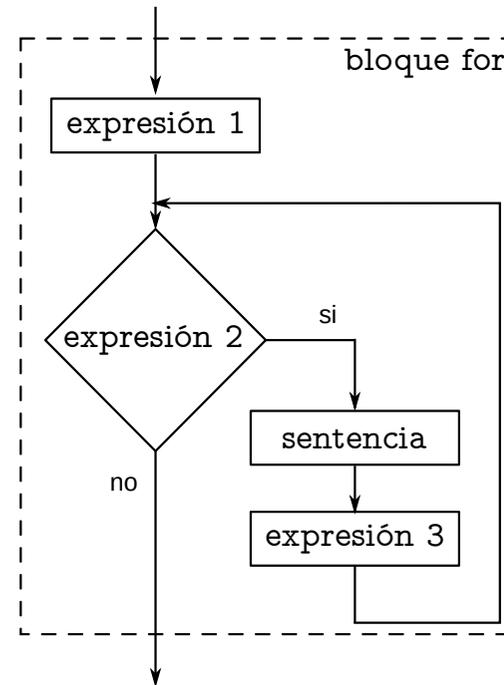
# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

La expresión 2 sirve de condición para repetir el ciclo

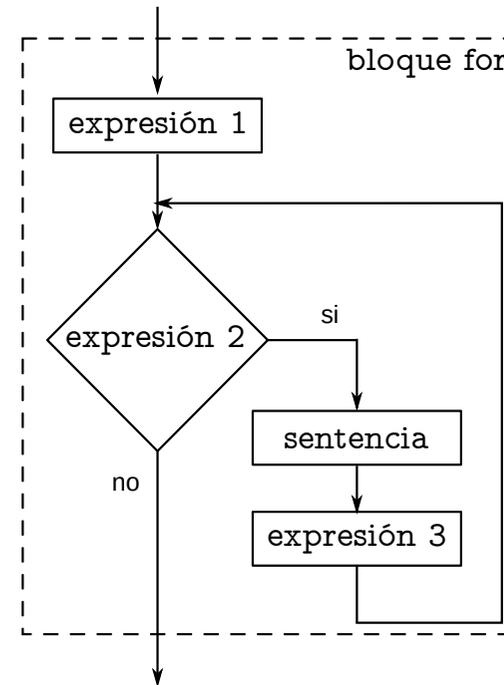
Se evalúa todos los ciclos



# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

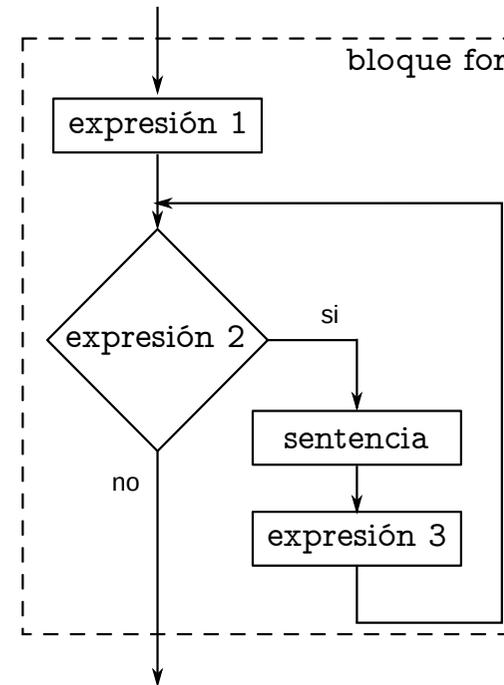


# Sentencias de repetición (o iterativas)

## Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

La sentencia puede ser simple, compuesta, condicional o iterativa.



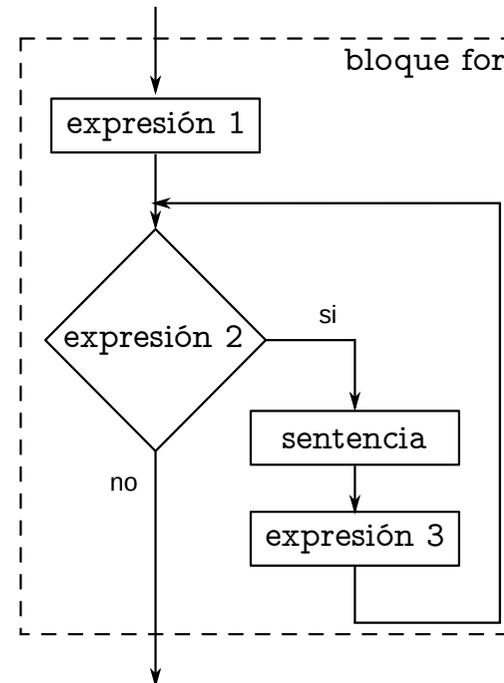
# Sentencias de repetición (o iterativas)

## Sentencia repetitiva for

Tiene un funcionamiento semejante al while controlado por contador

La sentencia puede ser simple, compuesta, condicional o iterativa.

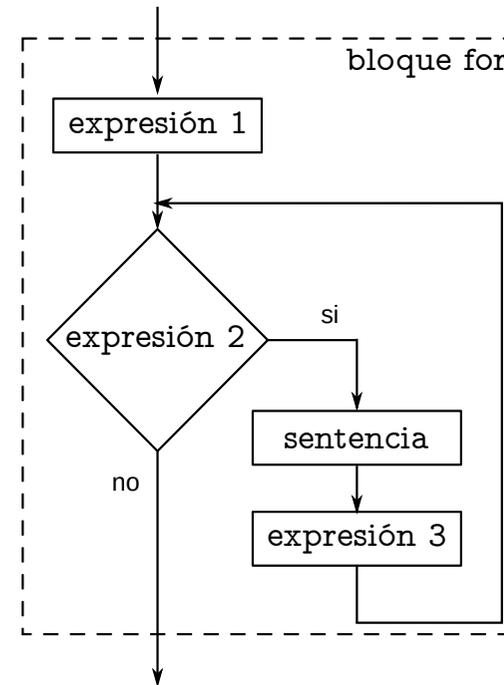
Se ejecuta todos los ciclos



# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Tiene un funcionamiento semejante al `while` controlado por contador

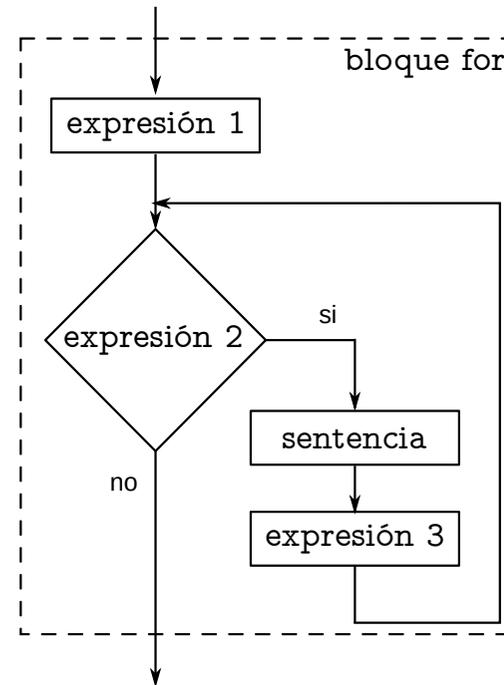


# Sentencias de repetición (o iterativas)

## Sentencia repetitiva for

Tiene un funcionamiento semejante al while controlado por contador

La expresión 3 se ejecuta todos los ciclos al finalizar la sentencia



# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

```
for ( expresión 1; expresión 2; expresión 3 )  
    sentencia
```

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

```
for ( expresión 1; expresión 2; expresión 3 )  
    sentencia
```

Las expresiones son opcionales

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

```
for ( expresión 1; expresión 2; expresión 3 )  
    sentencia
```

Las expresiones son opcionales

Si falta la expresión 2 se considera que la condición es verdad y el bucle continúa indefinidamente

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

```
for ( i = 0; i < 5; i++ )  
    printf("vuelta %d\n", i);
```

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

```
for ( i = 0; i < 5; i++ )  
    printf("vuelta %d\n", i);
```

```
vuelta 0  
vuelta 1  
vuelta 2  
vuelta 3  
vuelta 4
```

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

A partir del estándar C99 se puede declarar la variable de control en la expresión 1

# Sentencias de repetición (o iterativas)

## Sentencia repetitiva `for`

A partir del estándar C99 se puede declarar la variable de control en la expresión 1

```
#include <stdio.h>
// u5-for-init-1.c

int main (void) {

    for ( int i = 0; i < 10 ; i++ ) {
        printf("%d\n", i);
    }

    return 0;
}
```

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

# Sentencias de repetición (o iterativas)

Sentencia repetitiva `for`

Pero la variable solo existe en el bloque del `for`

# Sentencias de repetición (o iterativas)

## Sentencia repetitiva for

Pero la variable solo existe en el bloque del for

```
#include <stdio.h>
// u5-for-init-2.c

int main (void) {

    for ( int i = 0; i < 10 ; i++ ) {
        printf("%d\n", i);
    }
    printf("%d\n", i);

    return 0;
}
```

# Sentencias de repetición (o iterativas)

## Sentencia repetitiva for

Pero la variable solo existe en el bloque del for

```
#include <stdio.h>
// u5-for-init-2.c

int main (void) {

    for ( int i = 0; i < 10 ; i++ ) {
        printf("%d\n", i);
    }
    printf("%d\n", i);

    return 0;
}
```

```
$ gcc -Wall -std=c99 -pedantic-errors u5-for-init-2.c
u5-for-init-2.c: In function 'main':
u5-for-init-2.c:9:18: error: 'i' undeclared (first use in this function)
    printf("%d\n", i);
                   ^
```