

Fundamentos de Programación con JavaScript

Estructuras

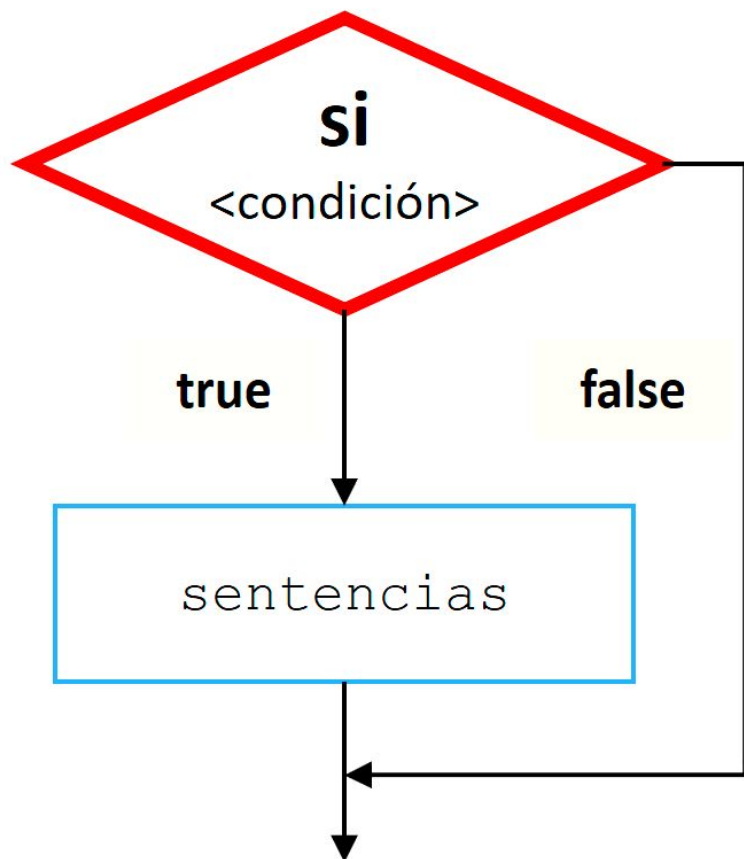
07

JS



Estructuras CONDICIONALES: sentencia **if**

Diagrama de flujo



Pseudocódigo

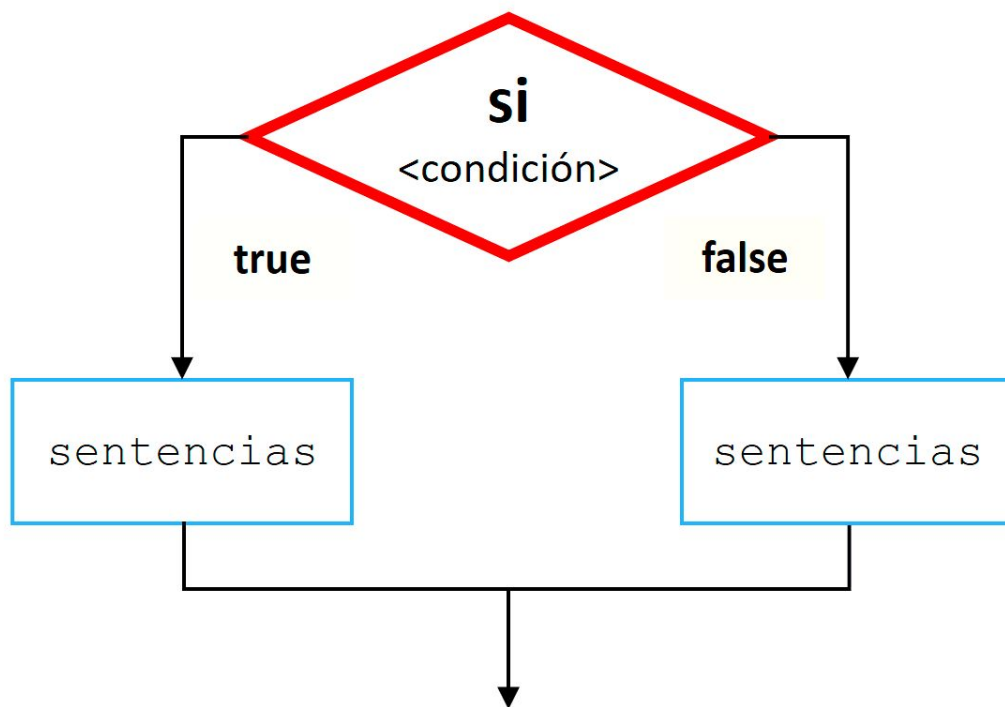
```
si <condición> entonces  
    [sentencia]  
fin_si
```

Javascript

```
if(condicion) {  
    // Bloque de código cuando se cumple  
    <condicion>...  
}
```

Estructuras CONDICIONALES: sentencia **if - else**

Diagrama de flujo



Pseudocódigo

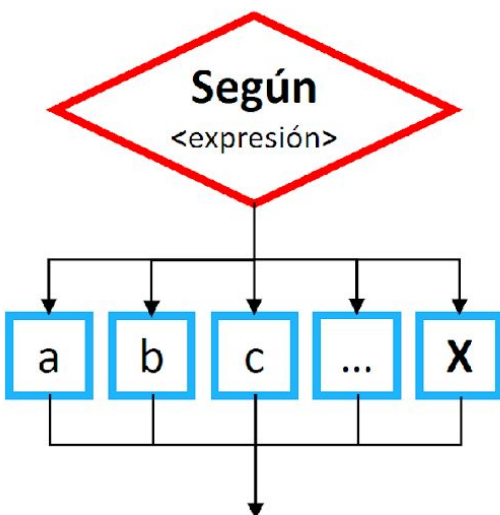
```
si <condición> entonces  
    [sentencia]  
sino  
    [sentencia]  
fin_si
```

Javascript

```
if(condicion) {  
    // Bloque de código cuando se cumple <condición>  
}  
else {  
    // Bloque cuando NO se cumple <condición>  
}
```

Estructuras CONDICIONALES: sentencia **switch**

Diagrama de flujo



Pseudocódigo

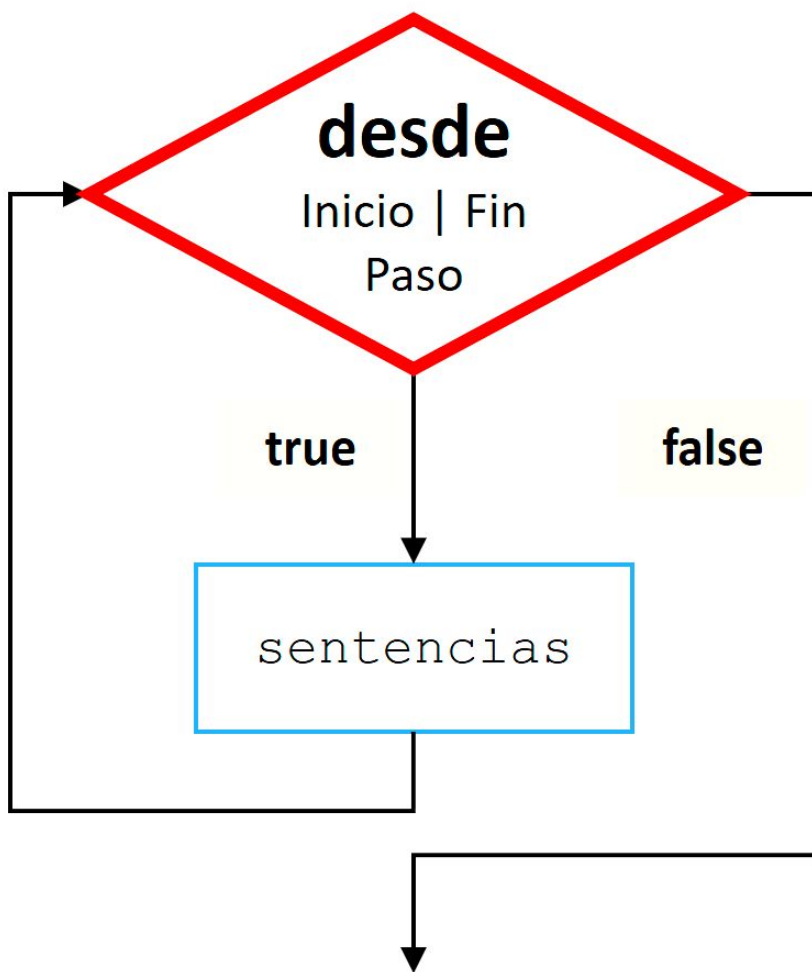
```
según_sea (expresión) hacer
  caso valor1:
    [sentencia]
    [sentencia]
    ...
    [ruptura]
  caso valor2:
    [sentencia]
    [sentencia]
    ...
    [ruptura]
  ...
  caso valorN:
    [sentencia]
    [sentencia]
    ...
    [ruptura]
  si_no:
    [sentencia]
    [sentencia]
    ...
fin_según
```

Javascript

```
switch (expresion) {
  case valor1:
    //Sentencias ejecutadas cuando el resultado de expresion
    coincide con valor1
    break;
  case valor2:
    //Sentencias ejecutadas cuando el resultado de expresion
    coincide con valor2
    break;
  ...
  case valorN:
    //Sentencias ejecutadas cuando el resultado de expresion
    coincide con valorN
    break;
  default:
    //Sentencias ejecutadas cuando no ocurre una coincidencia
    con los anteriores casos
}
```

Estructuras REPETITIVAS: sentencia **for**

Diagrama de flujo



Pseudocódigo

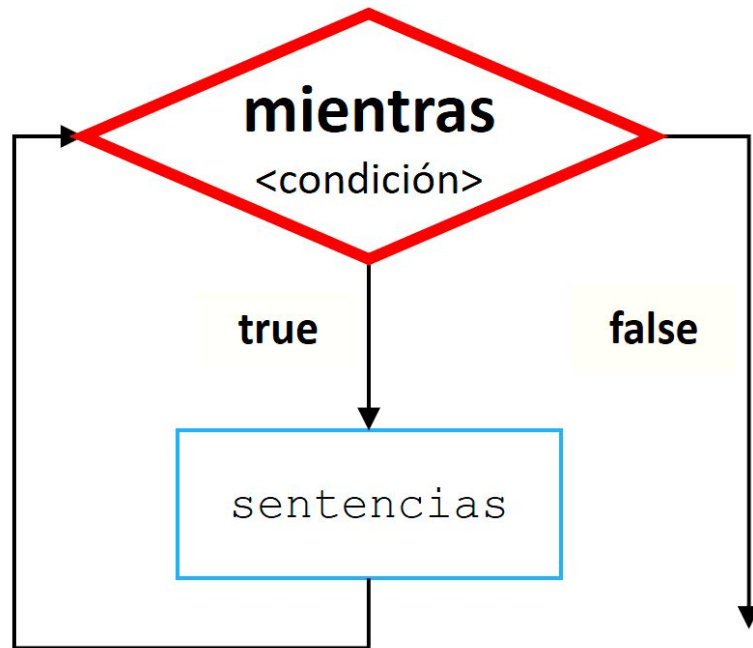
```
desde (inicio) hasta (fin) (paso) hacer  
    [sentencia  
    sentencia  
    ...]  
fin_desde
```

Javascript

```
for(inicializacion; condicion; actualizacion){  
    // Bloque de código que se repite n-veces...  
}
```

Estructuras REPETITIVAS: sentencia **while**

Diagrama de flujo



Pseudocódigo

```
mientras <condición> hacer  
    [sentencia  
    sentencia  
    ...]  
fin_mientras
```

Javascript

```
while(condicion) {
```

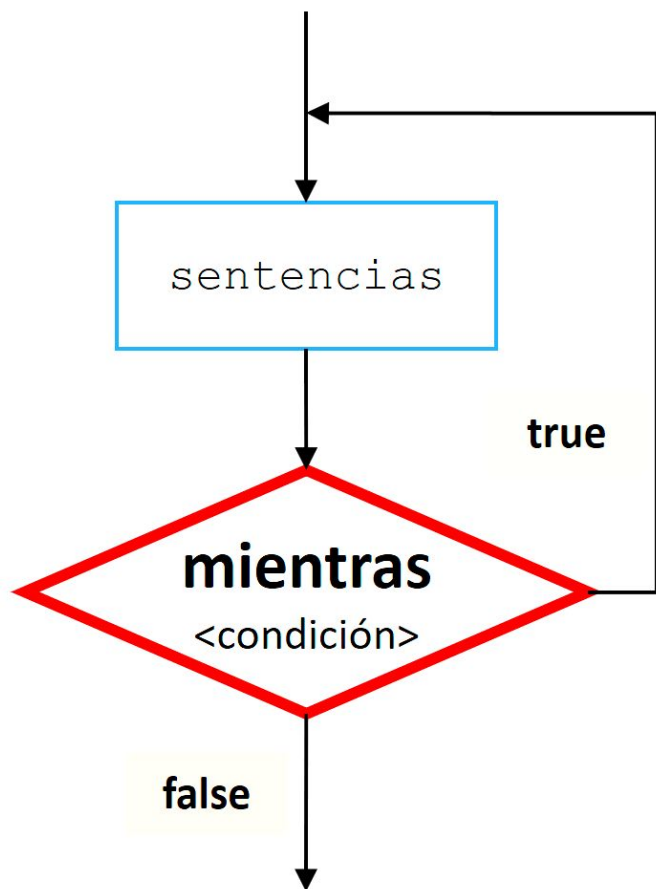
```
    /* Bloque de código que se ejecuta mientras se  
    cumple la <condición> ...
```

```
    Las variables que controlan la condición deben  
    modificarse dentro del propio bucle sino el bucle  
    se ejecuta indefinidamente y bloqueamos el  
    programa. */
```

```
}
```


Estructuras CONDICIONALES: sentencia **do**

Diagrama de flujo



Pseudocódigo

```
hacer  
    [sentencia  
    sentencia  
    ...]  
mientras <condicion>
```

Javascript

```
do {
```

```
    /* Bloque de código que se ejecuta una vez y luego  
    mientras se cumple la <condición> ...
```

```
    Las variables que controlan la condición deben  
    modificarse dentro del propio bucle sino el bucle  
    se ejecuta indefinidamente y bloqueamos el  
    programa. */
```

```
} while(condicion);
```

Fundamentos de Programación con JavaScript



JS