Operatori Aritmetici

Eseguono operazioni matematiche.

Operatore	Descrizione	Esempio	Risultato
`+`	Addizione	`5 + 2`	`7`
`-`	Sottrazione	`5 - 2`	`3`
`*`	Moltiplicazione	`5 * 2`	`10`
`/`	Divisione	`5 / 2`	`2.5`
`%`	Modulo (resto)	`5 % 2`	`1`
`++`	Incremento	`let x = 5; x++;`	`x` diventa `6`
`--`	Decremento	`let x = 5; x--;`	`x` diventa `4`

Operatori di Assegnazione

Assegnano valori alle variabili.

Operatore	Esempio	Equivalente a
`=`	`x = 5`	`x = 5`
`+=`	`x += 2`	`x = x + 2`
`-=`	`x -= 2`	`x = x - 2`
`*=`	`x *= 2`	`x = x * 2`
`/=`	`x /= 2`	`x = x / 2`
`%=`	`x %= 2`	`x = x % 2`

Operatori di Confronto

Confrontano due valori e restituiscono un valore booleano (`true` o `false`).

N.B. È generalmente raccomandato usare `===` e `!==` per confronti più rigorosi che tengono conto anche del tipo di dato.

Operatore	Descrizione	Esempio	Risultato (se x=5, y=3)
`==`	Uguale a (valore)	`x == 5`	`true`
`===`	Uguale a (valore e tipo)	`x === "5"`	`false`
`!=`	Diverso da (valore)	`x != 3`	`true`
`!==`	Diverso da (valore e tipo)	`x !== "5"`	`true`
`>`	Maggiore di	`x > y`	`true`
`<`	Minore di	`x < y`	`false`
`>=`	Maggiore o uguale a	`x >= 5`	`true`
`<=`	Minore o uguale a	`x <= 3`	`false`

Operatori Logici

Combinano o modificano espressioni booleane.

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

Logica Decisionale: L'istruzione if

L'istruzione if permette di eseguire un blocco di codice solo se una determinata condizione è vera.

La **condizione** all'interno delle parentesi tonde deve essere un'espressione che valuta a `true` o `false`.

```
let età = 20;

if (età >= 18) {
  console.log("Sei maggiorenne.");
}
```

Logica Decisionale: L'istruzione if … else

L'istruzione `if...else` permette di eseguire un blocco di codice se la condizione è vera e un altro blocco di codice se la condizione è falsa.

```
let voto = 65;

if (voto >= 60) {
  console.log("Promosso!");
} else {
  console.log("Bocciato.");
}
```

Logica Decisionale: L'istruzione if … else is … else

L'istruzione `if...else` permette di eseguire un blocco di codice se la condizione è vera e un altro blocco di codice se la condizione è falsa.

```
let punteggio = 75;

if (punteggio >= 90) {
  console.log("Ottimo!");
} else if (punteggio >= 70) {
  console.log("Buono.");
} else if (punteggio >= 60) {
  console.log("Sufficiente.");
} else {
  console.log("Insufficiente.");
}
```

La Struttura di Controllo switch

Alternative a `if...else if...else` per Scelte Multiple

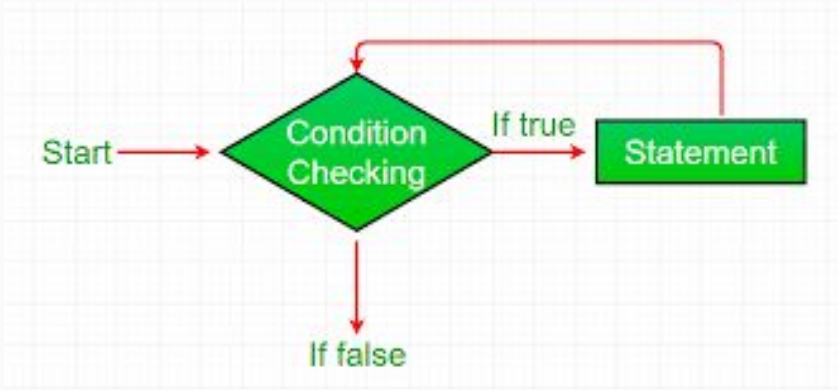
- L'istruzione `switch` valuta un'**espressione** e confronta il suo valore con diversi **casi** (`case`).
- Se il valore dell'espressione corrisponde al valore di un `case`, viene eseguito il blocco di codice associato a quel `case`.

```
let giorno = "Lunedì";

switch (giorno) {
  case "Lunedì":
    console.log("Inizio settimana!");
    break;
  case "Venerdì":
    console.log("Quasi weekend!");
    break;
  case "Sabato":
  case "Domenica":
    console.log("Weekend!");
    break;
  default:
    console.log("Giorno feriale.");
}
```

Cosa Sono i Cicli?

- I **cicli** (o loop) sono strutture di controllo fondamentali nella programmazione che permettono di eseguire ripetutamente un blocco di codice (il "corpo" del ciclo) fino a quando una determinata **condizione** è vera (o falsa, a seconda del tipo di ciclo).
- Consentono di automatizzare compiti ripetitivi, evitando di scrivere lo stesso codice più volte.



Il Ciclo for

For loop

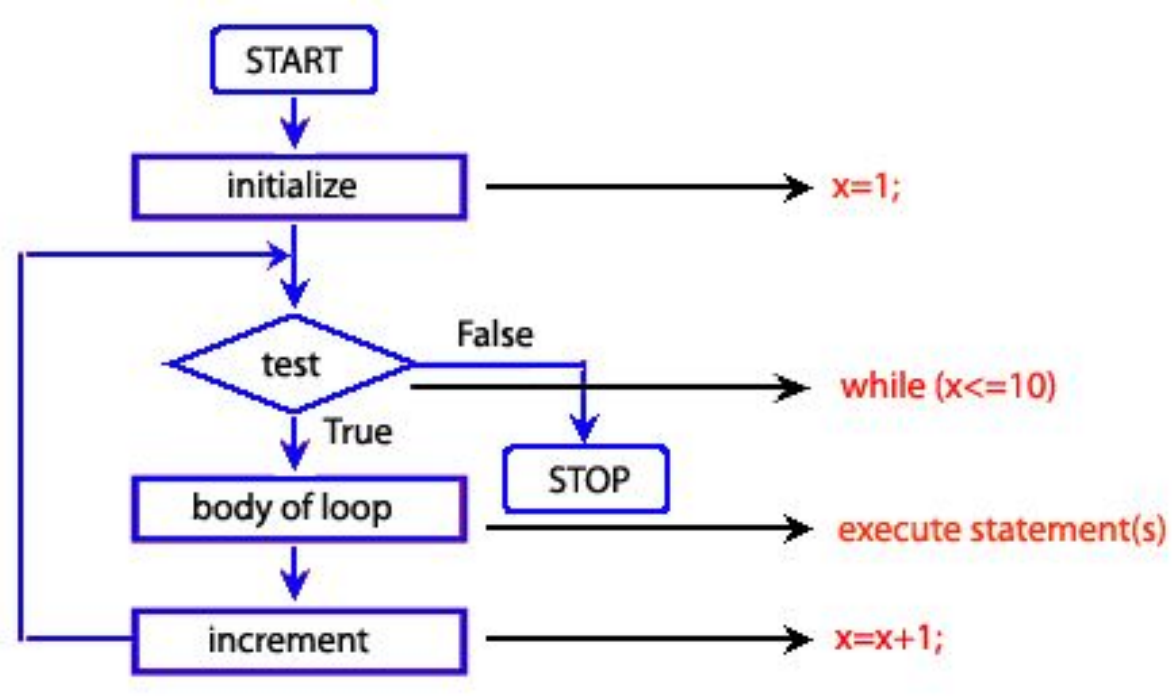
Condition:
- initialisation
- condition
- updater

for keyword

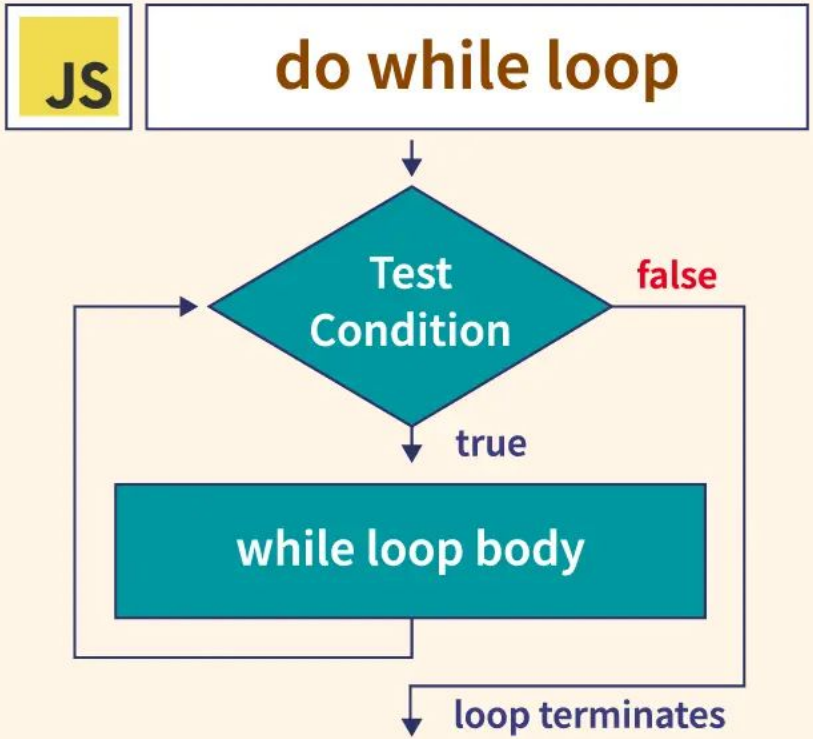
```
for (let i = 0; i < 10; i++) {
    console.log(i);
}
```

Code to execute when the loop runs

Il Ciclo while



Il Ciclo do … while



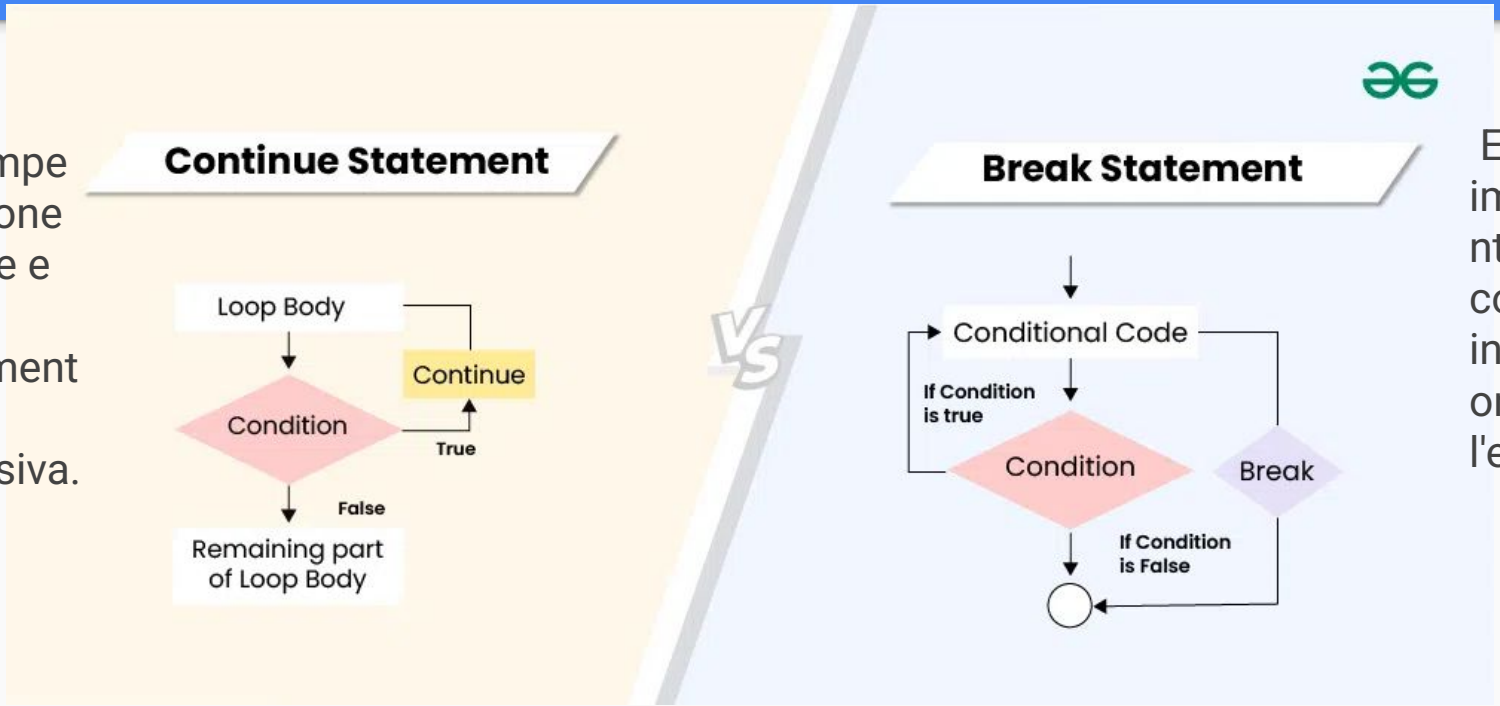
Iterare su Array con for .. of

```
for (let n of x)

for (let n in y)
```

Istruzioni break e continue nei Cicli

Interrompe l'iterazione corrente e passa direttamente alla successiva.



Esce immediatamente dal ciclo corrente, interrompendone l'esecuzione.