Condizioni in JS Programmazione

Le condizioni in JavaScript rappresentano uno dei costrutti fondamentali della programmazione, perché permettono di controllare il flusso di esecuzione del codice in base a determinate situazioni.

Attraverso le strutture condizionali, un programma può prendere decisioni diverse a seconda dei valori delle variabili o dei risultati di un'espressione logica.

L'istruzione più comune è if, che consente di eseguire un blocco di codice soltanto se la condizione specificata è vera.

A questa si possono aggiungere le clausole else e else if, che permettono di gestire i casi alternativi e di coprire scenari multipli con maggiore flessibilità.



In JavaScript, le condizioni si basano su espressioni booleane, ossia verifiche che restituiscono true o false.

Tali espressioni possono includere operatori di confronto (==, ===, !=, <, >, <=, >=) e operatori logici (&& per AND, || per OR, ! per NOT).

Grazie a questi strumenti, è possibile costruire verifiche anche molto articolate e precise.

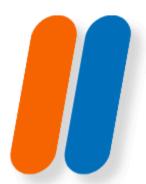
L'uso corretto delle condizioni è cruciale per rendere un programma interattivo e dinamico, dato che consente di adattarne il comportamento alle diverse situazioni che possono verificarsi durante l'esecuzione.



- Struttura base if → consente di eseguire un blocco di codice solo se la condizione risulta vera (true).
- Uso di else e else if → permettono di gestire i casi alternativi: else copre tutte le situazioni non specificate, mentre else if introduce controlli aggiuntivi.
- Operatori di confronto → come ==, ===, !=, <, >, <=, >= sono fondamentali per verificare uguaglianze o differenze tra valori.
- Operatori logici → && (AND), || (OR), ! (NOT) consentono di combinare più condizioni in un'unica espressione booleana.

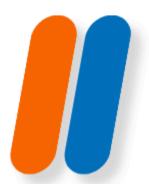


- Tipi di uguaglianza (== vs ===) → == confronta solo i valori con conversione di tipo automatica, mentre === richiede che siano uguali sia il valore che il tipo (più sicuro e consigliato).
- Truthy e Falsy → in JavaScript, non solo true e false, ma anche valori come 0, "", null, undefined e NaN sono considerati false (falsy), mentre tutti gli altri sono true (truthy).
- Operatore ternario ?: → rappresenta una forma compatta di if/else, utile per condizioni brevi e assegnazioni.
- Scope e leggibilità → usare sempre le parentesi graffe {} rende il codice più chiaro e riduce gli errori, anche quando il blocco contiene una sola istruzione.



```
1.let numero = 5;
2.
3.// Se la condizione è vera, il blocco viene eseguito
4.if (numero > 0) {
5. console.log("Il numero è positivo");
6.}
```

Qui l'istruzione if controlla se la variabile numero è maggiore di 0. Se la condizione è vera (true), viene eseguito il blocco di codice al suo interno.



```
1.let eta = 15;
2.
3.if (eta >= 18) {
4. console.log("Puoi entrare");
5.} else {
6. console.log("Accesso negato");
7.}
```

Con else gestiamo il caso in cui la condizione principale non sia verificata. In questo esempio, se l'età è minore di 18, il programma esegue il blocco alternativo.



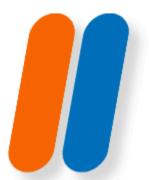
```
1.let voto = 7;
2.
3.if (voto >= 9) {
4. console.log("Ottimo");
5.} else if (voto >= 6) {
6. console.log("Sufficiente");
7.} else {
8. console.log("Insufficiente");
9.}
```

La clausola else if permette di aggiungere più controlli in sequenza. In base al valore della variabile voto, viene stampato un messaggio diverso.



```
1.let a = 10;
2.let b = "10";
3.
4.console.log(a == b); // true → confronta solo il valore
5.console.log(a === b); // false → confronta valore e tipo
```

In JavaScript esistono due tipi di uguaglianza: == fa una conversione implicita dei tipi e controlla solo i valori, mentre === controlla sia il valore che il tipo (più sicuro da usare).



```
1.let user = "admin";
2.let password = "1234";
3.
4.if (user === "admin" && password === "1234") {
5. console.log("Accesso consentito");
6.}
```

Gli operatori logici permettono di combinare condizioni: && richiede che entrambe siano vere, || almeno una, e ! ne inverte il risultato.

Qui l'accesso è concesso solo se username e password coincidono entrambi.



```
1.let valore = "";
2.
3.if (valore) {
4. console.log("Truthy");
5.} else {
6. console.log("Falsy"); // "" è considerato false
7.}
```

In JavaScript alcuni valori sono automaticamente interpretati come false (0, stringa vuota "", null, undefined, NaN), mentre tutti gli altri sono true. Questo rende possibile scrivere controlli rapidi.



```
1.let numero2 = 20;
2.
3.// Forma compatta di if/else
4.let risultato = (numero2 % 2 === 0) ? "Pari" : "Dispari";
5.console.log(risultato);
```

L'operatore ternario ? : è una scorciatoia per scrivere if/else in una sola riga.

Qui controlla se il numero è pari o dispari e assegna direttamente la stringa corrispondente.



```
1.let x = 3;
2.
3.if (x > 2) {
4. // usare le parentesi graffe {} rende chiaro il blocco
5. console.log("x è maggiore di 2");
6.}
```

Anche se JavaScript consente di omettere le graffe quando c'è una sola istruzione, è buona pratica usarle sempre.

Questo migliora la leggibilità ed evita errori in programmi più complessi.



```
1.// Programma che simula un controllo di accesso con bonus
 3.let nome = "Luca":
 4. let eta = 20:
 5.let voto = 7;
 6.let password = "1234";
 8.// 1. Controllo età con if/else
 9. if (eta < 18) {
10. console.log(nome + " non può accedere (minorenne).");
11.} else {
12. console.log(nome + " è maggiorenne.");
13.
14. // 2. Controllo password con operatore logico &&
15. if (password === "1234" && nome === "Luca") {
    console.log("Accesso consentito.");
17.
18. // 3. Valutazione del voto con else if
    if (voto >= 9) {
     console.log("Valutazione: Ottimo");
21. } else if (voto >= 6) {
      console.log("Valutazione: Sufficiente");
23. } else {
      console.log("Valutazione: Insufficiente");
26.
27. // 4. Uso operatore ternario per assegnare bonus
    let bonus = (voto >= 8) ? 10 : 5;
     console.log("Bonus ottenuto: " + bonus);
30.
    // 5. Dimostrazione di truthy/falsy
    let codicePromo = ""; // stringa vuota → falsy
     if (codicePromo) {
      console.log("Hai usato un codice promo.");
     } else {
      console.log("Nessun codice promo inserito.");
36.
37. }
38.
39. } else {
    console.log("Accesso negato: credenziali errate.");
41.
42.}
```

Spiegazione

- if/else controlla se l'utente è maggiorenne.
- Operatore logico && verifica che nome e password coincidano.
- else if valuta il voto in tre fasce diverse.
- Operatore ternario ? : assegna un bonus a seconda del voto.
- Truthy/Falsy viene mostrato con una stringa vuota che non passa la condizione.

