TANKERINO.com

Sintassi di base di JavaScript

JavaScript, come molti linguaggi di programmazione, ha una sua sintassi unica. Comprendere questa sintassi è fondamentale per poter scrivere codice corretto ed efficace. In questa lezione, esploreremo i concetti fondamentali della sintassi di JavaScript.

1. Dichiarazione di variabili

In JavaScript, possiamo dichiarare variabili usando le parole chiave `var`, `let`, `const`. Ognuna di queste ha le sue caratteristiche specifiche.

```
var nome = "Mario";
```

```
let cognome = "Rossi";
```

```
const eta = 30;
```

In JavaScript, le variabili sono contenitori utilizzati per conservare dati che possono variare nel tempo. La dichiarazione di una variabile avviene mediante alcune parole chiave specifiche. Vediamole in dettaglio:

var: Introdotto originariamente con JavaScript, il comando `var` è utilizzato per dichiarare una variabile. Tuttavia, ha alcune limitazioni e peculiarità, in particolare riguardo al suo "hoisting" (sollevamento) e al suo ambito di funzione. [Vai all'approfondimento]

```
var nome = "Mario"; // Una variabile chiamata "nome" con il valore "Mario"
```

let: Introdotto con ES6 (ECMAScript 2015), `let` è simile a `var`, ma ha un ambito di blocco, il che significa che è limitato all'interno del blocco, dell'istruzione o dell'espressione in cui viene utilizzato. È il modo raccomandato per dichiarare variabili in molti casi.

```
let cognome = "Rossi"; // Una variabile chiamata "cognome" con il valore "Rossi"
```

const: Anchor esso introdotto con ES6, `const` viene utilizzato per dichiarare variabili il cui valore non dovrebbe cambiare nel tempo, rendendole di fatto costanti. Una volta assegnato un valore a una variabile dichiarata con `const`, non è possibile riassegnarle un nuovo valore.

È una buona pratica utilizzare `const` per default, a meno che non tu sappia che il valore della variabile dovrà cambiare in futuro. In quel caso, usa `let`.

```
const etaMinima = 18; // Una costante chiamata "etaMinima" con il valore 18
```

In generale, l'uso di `let` e `const` è preferito rispetto a `var` nella programmazione moderna in JavaScript grazie alla loro maggiore prevedibilità e chiarezza nell'ambito.

2. Tipi di dati

In JavaScript, ogni valore ha un tipo associato che determina le sue caratteristiche e le operazioni che si possono eseguire su di esso. Ecco una panoramica dei principali tipi di dati:

1. Number

Il tipo `Number` rappresenta sia i numeri interi sia quelli a virgola mobile.

```
let intero = 42; // Numero intero
```

```
let decimale = 3.14; // Numero a virgola mobile
```

2. String

Il tipo `String` rappresenta una sequenza di caratteri e viene solitamente dichiarato tra virgolette singole, doppie o backticks.

```
let saluto = 'Ciao!'; // Con virgolette singole
```

```
let nome = "Mario"; // Con virgolette doppie
```

`let presentazione = `Il mio nome è ${nome}`;` // Con backticks, permettendo l'interpolazione di stringhe

3. Boolean

Il tipo `Boolean` ha solo due valori possibili: `true` (vero) o `false` (falso). È spesso utilizzato in operazioni logiche e condizioni.

`let acceso = true;` // Il valore è vero

`let maggiorenne = false;` // Il valore è falso

4. Object

Il tipo `Object` rappresenta una collezione di chiavi e valori. Gli oggetti possono contenere qualsiasi tipo di dato, incluso altri oggetti, e sono estremamente versatili.

```
let persona = {
nome: "Anna",
eta: 25,
hobby: ["leggere", "cucinare"]
};
```

5. Array

Gli `Array` sono elenchi ordinati di valori e possono contenere qualsiasi tipo di dato. Gli elementi di un array sono accessibili tramite il loro indice, che inizia da 0.

`let frutti = ["mela", "banana", "arancia"];`

`let primoFrutto = frutti[0];` // Accede al primo elemento dell'array, "mela"

6. Undefined e Null

Il tipo `undefined` rappresenta una variabile che non è stata ancora assegnata, mentre `null` rappresenta un valore nullo o assente intenzionalmente.

`let nonDefinito;` // Questa variabile è undefined

```
let vuoto = null; // Questa variabile ha un valore null
```

Ci sono altri tipi e sottotipi di dati in JavaScript, ma questi sono i più comuni e fondamentali da conoscere quando si inizia. Comprendere questi tipi e come lavorare con essi è essenziale per scrivere codice JavaScript efficace.

3. Operatori

JavaScript offre una serie di operatori per eseguire operazioni matematiche, di confronto e logiche.

```
let somma = 5 + 3; // Matematico

let maggiore = 5 > 3; // Confronto

let entrambi = true && false; // Logico
```

4. Controllo del flusso

Per controllare il flusso di esecuzione del nostro programma, possiamo utilizzare istruzioni condizionali e cicli.

```
if (condizione) {
// esegui se la condizione è vera
} else {
// esegui se la condizione è falsa
}

for (let i = 0; i < 5; i++) {
// ripeti per 5 volte
}
```

5. Funzioni

Le funzioni sono blocchi di codice riutilizzabili che possono accettare input (parametri) e restituire un output.

```
function saluta(nome) {
return "Ciao " + nome + "!";
}
```

```
let messaggio = saluta("Marco");
```

Conclusione

Questa è solo una panoramica molto basilare della sintassi di JavaScript. Man mano che continuerai a imparare, incontrerai molti altri concetti e dettagli sintattici.

Ma comprendere questi fondamenti ti fornirà una solida base su cui costruire le tue conoscenze future. Buona programmazione!

Questa lezione e' stata realizzata grazie al contributo di:

Risorse per la scuola
https://www.baobab.school

Siti web a Varese
https://www.francescobelloni.it