



Coding Bootcamp

Web fundamentals

Javascript - Parte III

Programma

Le funzioni

Le costanti

Object

Object - Proprietà

Object - Metodi

Le Funzioni

Le funzioni sono un blocco di istruzioni che eseguono un compito e facilitano notevolmente il compito di scrivere del codice sintetico e riutilizzabile.

```
function somma(num1, num2) {  
  return num1 + num2;  
}
```

Il codice contenuto all'interno delle parentesi graffe non viene eseguito subito da javascript ma solo quando la funzione viene chiamata:

```
let totale = somma(3, 2);    // alla variabile 'totale' viene assegnato 5
```

Le Costanti

const è il metodo per dichiarare le costanti all'interno del nostro file js.

Sintassi:

```
const listChars = [ ];
```

```
const cssFolder = 'myprojects/myapp/build/assets/css/';
```

- non si può assegnare un nuovo valore ad una costante una volta dichiarata (accessibile solo in lettura);
- può essere dichiarata con qualsiasi tipologia di dato;
- un oggetto o un array assegnato via reference possono essere modificati ma non si possono assegnare nuove reference alla costante.

NOTA:

Da un punto di vista prettamente convenzionale, molti sviluppatori sono soliti definire i valori costanti all'interno di comuni variabili caratterizzate da un nome in maiuscolo.

Object

Un oggetto è un contenitore di valori che messi insieme formano una struttura dati unica.

Normalmente, un oggetto è utilizzato per rappresentare un'entità specifica come ad esempio una persona, un ordine, una fattura, una prenotazione, etc. tramite un'aggregazione di dati (**proprietà**) e di funzionalità (**metodi**).

Sintassi:

```
let person = {  
  name: 'Arthur',  
  surname: 'Fleck',  
  age: '37',  
  job: 'Comedian'  
};
```

- Per dichiarare una proprietà utilizziamo i due punti (:) e non il segno uguale (=)
- La **virgola** tra le chiavi è indispensabile (simile al CSS ma con la “,” invece del “;”);

Object - Proprietà

Le **proprietà** sono rappresentate da coppie di nome-valore.

Per i nomi delle proprietà di un oggetto non abbiamo le restrizioni dei nomi delle variabili. I doppi apici intorno al nome della proprietà sono opzionali, ma diventano obbligatori quando il nome contiene dei caratteri speciali.

```
let person = {  
  name: 'Arthur',  
  surname: 'Fleck',  
  age: '37',  
  job: 'Comedian',  
  "previous-job": 'Clown'  
};
```

Object - Proprietà

Accedere alle proprietà

Per accedere ai valori memorizzati in una proprietà abbiamo due approcci:

```
let name = person.name;
```

```
let name = person[ " name " ];
```

- Nel primo approccio indichiamo l'oggetto e la proprietà a cui siamo interessati separandoli con un punto
- Nel secondo approccio specifichiamo la proprietà di un oggetto indicando il suo nome sotto forma di stringa tra parentesi quadre

Object - Proprietà

Creazione per definizione

Per accedere ai valori memorizzati in una proprietà abbiamo due approcci:

```
let person = {};  
person.name = ' ';  
person[ " name " ];
```

- Se proviamo ad assegnare un valore ad una proprietà non definita creiamo di fatto questa proprietà inizializzandola con il valore assegnato:

Object - Metodi

A differenza delle proprietà di un oggetto che rappresentano dati, i **metodi** rappresentano attività che un oggetto può compiere.

```
let person = {  
  name: 'Arthur',  
  surname: 'Fleck',  
  age: '37',  
  job: 'Comedian',  
  "previous-job": 'Clown',  
  getFullName: function()  
  {  
    return this.name    // ← this parola chiave che rappresenta l'oggetto a cui appartiene il metodo invocato.  
  }  
};
```