



# JavaScript

**Ing. Simone Giustetti**  
**[www.giustetti.net](http://www.giustetti.net)**

# JavaScript

Linguaggio di programmazione orientato al web.

Il nome ufficiale del linguaggio è **ECMAScript**.

Progettazione e sviluppo incominciano nel 1995.

Il primo standard è del 1997 (ECMA-262).

Versioni minori rilasciate tra il 1997 ed il 1999 (ES2, ES3).

La prima revisione sostanziale è del 2009 (ES5).

La seconda revisione completa nel 2015 (ES6).

Dal 2016 le nuove versioni vengono rilasciate su base annuale (2016, 2017, 2018, ...).



# JavaScript

JavaScript è utilizzato per conferire dinamicità alle pagine web permettendo, ad esempio, di:

- Modificare il contenuto delle pagine.
- Caricare elementi di una pagina separatamente dalla pagina stessa.
- Mostrare finestre di pop-up.
- Nascondere e mostrare parti di una pagina.
- Modificare le immagini.
- Controllare la riproduzione di flussi multimediali.
- Scrivere videogiochi.
- ...



# JavaScript

È un **linguaggio interpretato** compilato/eseguito al volo dal browser con cui si carica la pagina. L'interprete può essere un programma separato dal browser (FireFox / SpiderMonkey).

È un **linguaggio imperativo**: Il programmatore deve scrivere codice dettagliato per tutte le operazioni eseguite dall'interprete.

Può essere combinato con CSS e HTML per creare interfacce dinamiche.



Supporta sia il paradigma di programmazione **imperativo strutturato**, che **la programmazione ad oggetti**.

È un linguaggio a tipizzazione debole. La tipologia dei dati può essere cambiata arbitrariamente dallo interprete in base alle operazioni che esegue.

Il formato “predefinito” è: **stringa di testo**.  
Bisogna fare attenzione quando si eseguono operazioni matematiche sui dati.



# JavaScript e DOM

JavaScript dipende dall'interprete, generalmente il browser, per i metodi e gli oggetti attraverso cui interagire con l'ambiente.

Tutti i browser esportano la struttura delle pagine attraverso un **Document Object Model**.

Il DOM tratta una pagina HTML o XML come un albero gerarchico ove ogni nodo rappresenta parte del documento.

È possibile modificare una pagina con JavaScript attraverso il DOM.



# Dichiarare Codice JavaScript

Il codice JavaScript può essere aggiunto nelle pagine web attraverso una coppia di tag **<script>... </script>**.

I tag possono essere inseriti nella sezione `<head>` di una pagina, nel `<body>` oppure in entrambe.

È possibile inserire più di uno script in una singola pagina HTML.

È possibile inserire direttamente il codice, oppure un collegamento ad un file esterno.



# Dichiarare Codice JavaScript – Esempi

```
<html>  
  <head>  
    <title>Esempio 1</title>  
    <script src="script.js"></script>  
  </head>
```

...

```
<html>  
  <head>  
    <title>Esempio 2</title>  
    <script type="application/javascript">  
      document.write( "Ciao Mondo !" );  
    </script>  
  </head>
```

...



# I Commenti

Esistono due tipologie di commenti in JavaScript:

- I commenti di una singola riga

Ogni riga che incomincia con due slash (Shift+7).

```
Codice... // Commento  
// Commento
```

- I commenti inclusi in un blocco di testo

```
Codice... /* Commento  
Commento  
Commento */ Codice...
```



# Le Variabili

Una variabile è una porzione di memoria in cui vengono salvate informazioni per la durata in esecuzione di uno script.

Le variabili devono essere dichiarate esplicitamente oppure implicitamente mediante assegnazione di un valore.

```
var variabile;           // Dichiarazione esplicita  
var variabile = "test"; // Dichiarazione esplicita  
variabile = 5;          // Dichiarazione implicita
```



# Le Costanti

Una costante è una porzione di memoria in cui vengono salvate informazioni che non possono essere successivamente modificate.

Le costanti devono essere dichiarate con l'istruzione **const**.

```
const PI = 3.14159;  
PI = 3.14;           // Rende un errore  
PI = PI + 10;       // Rende un errore  
const PI;           // Errore. L'assegnazione è  
                    obbligatoria
```



# Tipologie di Dato

Le variabili JavaScript possono contenere numeri, stringhe di testo e strutture complesse come array ed oggetti.

```
var xd = 34.00; // Numero frazionario
```

```
var xi = 34; // Numero intero
```

```
var xs1 = 'Marco'; // Stringa di testo
```

```
var xs2 = "Matteo"; // Stringa di testo
```

```
xi2 = xi1; // Assegna a xi2 una copia del valore  
di xi1
```

```
var xb = true; // Valore booleano. Sono  
ammessi solo i valori false o true.
```



# Gli Array

Un array è una variabile che contiene molti dati dello stesso tipo.

Per dichiarare un array è necessario il ricorso alle parentesi quadre ed alla virgola.

```
// Array di 4 elementi
```

```
ar_01 = [ 34.00, 45.00, 67.12, 99.45 ];
```

Per accedere ad un elemento è necessario il ricorso alle parentesi quadre.

```
ar_01[ 3 ]; // 99.45. Le componenti partono da 0
```



# Gli Oggetti

Strutture dati composte da variabili e codice (Membri e Metodi).

Per dichiarare un oggetto è necessario il ricorso a parentesi graffe, due punti ed alla virgola.

// Oggetto di 3 elementi

```
obj_01 = { nome: "Mario", cognome: "Rossi",  
          anni: 45 };
```

Per accedere ad un elemento è necessario il ricorso al punto.

```
obj_01.cognome; // "Rossi"
```



# Gli Operatori

Gli operatori eseguono operazioni elementari sui dati.

Il risultato di un'operazione può variare in base alla tipologia del dato.

Esistono operatori specifici per ogni tipologia di dato.

Alcuni operatori hanno la precedenza su altri.  
Usare le parentesi tonde per imporre la precedenza tra le operazioni.



# Gli Operatori Matematici

Operano sui numeri e rendono sempre un valore numerico.

- `+`, `-` (Addizione e sottrazione algebrica)
- `*`, `/` (Moltiplicazione e divisione algebrica)
- `%` Il resto di una divisione intera (modulo)
- `++` Incremento (+ 1)
- `--` Decremento (- 1)
- `+=`, `-=`, `*=`, `/=`, `%=` Eseguono un'operazione ed assegnano il risultato



# Gli Operatori Stringa

Operano su stringhe di testo rendono sempre una stringa.

- + (Concatenazione)



# Gli Operatori Booleani

Gli operatori booleani sono usati per eseguire dei confronti e rendono sempre un valore **false** oppure **true**.

- >, >=
- <, <=
- == (Uguale)
- != (Diverso)
- && (And. Vero se entrambe le espressioni sono vere)
- || (Or. Vero quando almeno una espressione lo è)
- ! (Not. Nega un'espressione)



# L'Operatore typeof

L'operatore **typeof** può essere usato per determinare la tipologia di una variabile.

```
typeof "John" // Rende "string"  
typeof 0      // Rende "number"  
typeof 314    // Rende "number"  
typeof 3.14   // Rende "number"
```

Una variabile non inizializzata ha tipo e valore **undefined**.

```
var car; // Valore undefined. Tipo undefined
```



# Eliminare una Variabile

Una variabile può essere "pulita" assegnandole il valore **undefined**.

```
var car = "Volvo"; // Tipo "string"
```

```
var car = ""; // Tipo "string"
```

```
var car = undefined; // Valore undefined. Tipo  
"undefined"
```

Assegnando il valore **null**, si pulisce la variabile, che però assume la tipologia oggetto.

```
var persona = { nome: "Mario",  
cognome: "Rossi", anni: 40 };
```

```
persona = null; // Valore null. Tipo "object"
```



# Ciclo if

Un blocco di codice è eseguito solo se è verificata una condizione.

```
if( <condizione> ) {  
    <Blocco 1 di istruzioni>  
    ...  
} else {  
    <Blocco 2 di istruzioni>  
    ...  
}
```

Il blocco **else** è opzionale.



# Ciclo if

È possibile annidare le istruzioni **if**, oppure usare il costrutto **else if**, quando bisogna verificare molte condizioni in cascata.

```
if( <condizione 1> ) {  
    <Blocco 1 di istruzioni>  
} else if( <condizione 2> ) {  
    <Blocco 2 di istruzioni>  
} else {  
    <Blocco 3 di istruzioni>  
}
```



# Ciclo switch

Costrutto condizionale che permette di verificare tutti i valori resi da una espressione.

```
switch( <espressione> ) {  
  case valore_01:  
    <Blocco 1 di istruzioni>  
  break;  
  case val2:  
    <Blocco 2 di istruzioni>  
  break;  
  ...  
  default:  
    <Blocco n di istruzioni>  
}
```



# Operatore Condizionale Ternario

L'operatore condizionale ternario lavora con tre valori. La sintassi è:

`<condizione> ? <espr01> : <espr02>`

È una sintassi compatta per un ciclo if, then, else.

Es:

`anni >= 18 ? "maggiorenne" : "minorenne";`



# Ciclo while

Esegue un blocco di codice un numero molteplice di volte in base alla verifica di una condizione.

```
while( <condizione> ) {  
    <Blocco 1 di istruzioni>;  
}
```

È possibile che il blocco non sia mai eseguito

Il blocco può essere eseguito all'infinito, se verifica sempre la condizione.



# Ciclo do..while

La verifica della condizione di esecuzione è posta in coda al ciclo invece che in testa.

```
do {  
    <Blocco di istruzioni>;  
} while( <condizione> );
```

Il ciclo viene sempre eseguito almeno una volta.



# Ciclo for

I cicli **for** usano un contatore che viene incrementato ad ogni iterazione.

```
for( <condizione_iniziale>; <condizione_finale>;  
    <incremento> ) {  
    <Blocco di istruzioni>;  
}
```

Viene eseguito sempre un numero determinato di volte.



# Iterare gli Elementi di un Array

In JavaScript esiste una versione semplificata del ciclo **for** che opera sul contenuto di un array.

```
for( let <indice> of <array> ) {  
    <Blocco di istruzioni>;  
}
```

Viene eseguito un numero di volte corrispondente alla cardinalità dello array.



# Cambiare il Flusso di un Ciclo

Le istruzioni **break** e **continue** cambiano il normale flusso di esecuzione di un ciclo.

- **break**: Causa l'uscita da un ciclo spostando il flusso del programma alla prima riga successiva il ciclo stesso.
- **continue**: Interrompe l'iterazione in corso e riprende il ciclo dall'iterazione successiva. Consente di saltare iterazioni.



## Canali Web

- [www.youtube.com/watch?v=dDekOIgzxPA&list=PL0qAPtx8YtJceyk5\\_NpNvLbbkrmfX9kkw](http://www.youtube.com/watch?v=dDekOIgzxPA&list=PL0qAPtx8YtJceyk5_NpNvLbbkrmfX9kkw)
- <https://www.youtube.com/watch?v=84TYC44ezIU>
- <https://www.youtube.com/watch?v=-eTn9DYc674>
- <https://www.youtube.com/watch?v=jayqQsljrGg>

## Manuali

- [www.w3schools.com/js/default.asp](http://www.w3schools.com/js/default.asp)
- [developer.mozilla.org/en-US/docs/Learn/JavaScript](http://developer.mozilla.org/en-US/docs/Learn/JavaScript)
- [it.wikibooks.org/wiki/JavaScript](http://it.wikibooks.org/wiki/JavaScript)
- [learnjavascript.online/](http://learnjavascript.online/)



# Informazioni & Licenze

## LICENZA

Salvo dove altrimenti specificato grafica, immagini e testo della presente opera sono © Simone Giustetti. L'opera può essere ridistribuita per fini non commerciali secondo i termini della licenza:

Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale



È possibile richiedere versioni rilasciate sotto diversa licenza scrivendo all'indirizzo: [studiosg@giustetti.net](mailto:studiosg@giustetti.net)

## TRADEMARK

- FreeBSD è un trademark di The FreeBSD Foundation.
- Linux è un trademark di Linus Torvalds.
- Macintosh, OS X e Mac OS X sono tutti trademark di Apple Corporation.
- MariaDB è un trademark di MariaDB Corporation Ab.
- MySQL è un trademark di Oracle Corporation.
- UNIX è un trademark di The Open Group.
- Windows e Microsoft SQL Server sono trademark di Microsoft Corporation.
- Alcuni algoritmi crittografici citati nella presente opera potrebbero essere protetti da trademark.

Si prega di segnalare eventuali errori od omissioni al seguente indirizzo: [studiosg@giustetti.net](mailto:studiosg@giustetti.net)

