

Introduzione a Javascript

Premessa

Il presente mini-tutorial è necessario per mettere in condizioni gli studenti di affrontare in modo semplice il linguaggio di programmazione "javascript".

Ciò che trovate scritto è ampiamente descritto sui vari tutorial on-line a cui ho fatto riferimento e ai quali vi chiedo di accedere per approfondire.

Il linguaggio JS, assieme all'HTML e al CSS (in realtà metalinguaggi) compone il bagaglio culturale di riferimento di ogni sviluppatore in ambito web-client.

Logica

Il linguaggio JS è detto ad "eventi", ovvero non si aziona da solo ma necessita di un evento esplicito od implicito. JS è anche ad oggetti sulla cui logica ci soffermeremo in un'altra parte del nostro programma, ovvero quando avrete affrontato la logica ad oggetti anche in informatica.

Nella vita, ad esempio, un incontro può far accadere qualcosa di piacevole (si spera!), in programmazione il click del mouse, la pressione di un pulsante, il resize della finestra, ecc. possono scatenare una determinata routine.

In poche parole, possiamo definire un evento Javascript come qualcosa che accade all'interno del documento HTML che è il nostro contesto. Nella maggior parte dei casi a scatenare l'evento è un comportamento dell'utente (ad esempio un click su un dato elemento della pagina) altre volte, invece, è una situazione contingente che prescinde da una qualsivoglia azione dell'utente (come ad esempio il caricamento della pagina).

Javascript supporta moltissimi eventi che, con l'evolversi del linguaggio, sono andati via via arricchendosi. (<https://www.mrwebmaster.it/>)

Per riferirsi ad un elenco aggiornato: https://www.w3schools.com/jsref/dom_obj_event.asp (non spaventatevi, non tutti si utilizzano!)

JavaScript è un linguaggio interpretato. Questo vuol dire che non viene compilato, dunque ogni comando viene eseguito direttamente dal browser.

Per utilizzare JS occorre conoscere la logica del DOM (Document Object Model) che sottende tutta l'ingegneria del software lato client. In questo caso faremo una piccola digressione nei paragrafi che seguono

Il linguaggio JS può essere inserito in qualsiasi punto della pagina HTML tramite il tag `<script></script>`

Cosa si può fare con Javascript

Validare dei form (moduli html – Partiamo da qui);

Gestire degli eventi all'interno di una pagina;

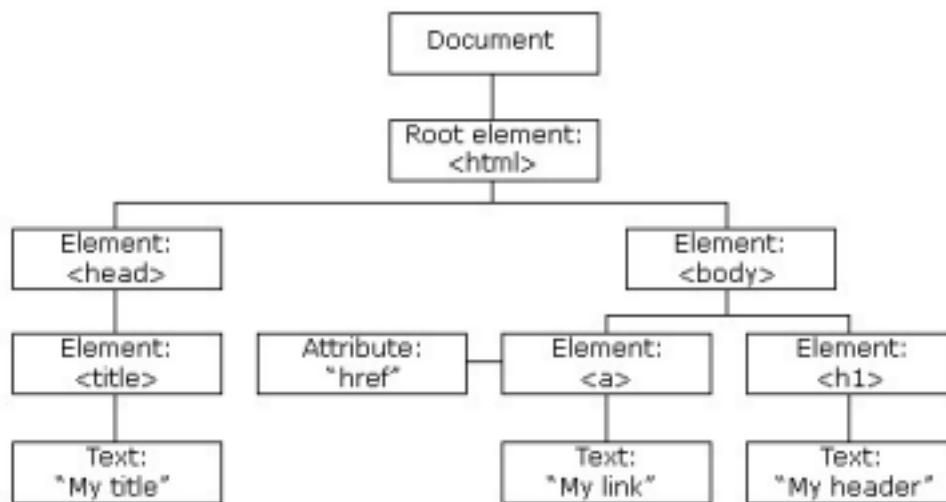
Aggiungere degli effetti alle pagine (tipo slideshow, photogallery);

Aprire finestre aggiuntive, soprattutto a scopo pubblicitario;

Realizzare un menù responsive per i dispositivi mobili;

Fornire messaggi di risposta in base a delle scelte effettuate

DOM



Per comprendere JS occorre capire come è fatto l'HTML e la sua struttura: la figura rappresenta come gli elementi sono in relazione tra di loro. Si dice infatti che l'HTML è un meta-linguaggio perché non è strutturato come un linguaggio vero e proprio ma ha, al suo interno, una logica gerarchica che permette al layout grafico che rappresenta di essere contenuto, ed a sua volta di contenere.

Come si può notare dal disegno, l'elemento principale è il "document" che è il formato HTML nel suo complesso: infatti, quando costruiamo un documento HTML iniziamo col definire il tag `<html>`.

Successivamente, creiamo due sezioni `<head>` e `<body>`. All'interno del `body`, poi, inseriamo gli elementi che possono contenere altri elementi: `<div>``<p>``<h1>`.

Ogni elemento può avere degli attributi ed un contenuto testuale. Uno di questi attributi è l'ID

In questo modo, quando ci riferiamo ad un elemento, possiamo attribuire ad esso un comportamento dedicato indicando un'azione.

Il comando è: `document.getElementById("demo")` dove "demo" è l'ID del tag. Es.

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Ciao!";
</script>

</body>
</html>
```

In questo caso la stringa "Ciao!" comparirà all'interno del tag `<p>`

Provate!

Caselle di testo

Per scrivere all'interno di una pagina HTML è possibile utilizzare un controllo "Input" che riceve una stringa di qualsiasi tipo, numerica o alfanumerica. Il tag è `<input type="text" id="demo">`

Per scatenare un evento è possibile utilizzare un bottone: `<input type="button" value="Cliccami" onclick="stampa()">Cliccami</button>`

Es.

```
<!DOCTYPE html>
<html>
<head>

<script>
function stampa() {
var print=document.getElementById("demo").value;

document.getElementById("dest").innerHTML=print;
}
</script>

</head>
<body>

<input type="text" id="demo">
<input type="button" value="Cliccami" onclick="stampa()">
<p id="dest"></p>
</body>

</html>
```

Questo esempio preleva il valore contenuto nell'input type text con id `demo` e lo stampa nel paragrafo di id `dest`

Analizzando lo script, il documento è diviso in due parti: la prima è in HTML e descrive la sezione di input. Due tipi di Input type permettono di scrivere il valore e di attivare lo script tramite l'evento onclick. Inoltre, predispone il paragrafo "p" a ricevere il risultato dello script.

La seconda parte, di tipo javascript, descrive la funzione "stampa". Come prima cosa si inizializza una variabile "print" con il contenuto dell'input text di id "demo" tramite la proprietà "value".

Infine, si inserisce all'interno del tag con id "dest" il valore della variabile "Print" tramite la proprietà "innerHTML".

La funzione ha necessariamente le parentesi (), per un eventuale argomento, e le parentesi graffe {} per racchiudere tutta la funzione. In questo caso non c'è un ritorno di valore perché si svolge all'interno della pagina html.

Anche in questo caso provate a svolgere un piccolo esercizio con le proprietà value e innerHTML