

data di ultima modifica

30/01/2013

Prerequisiti per la comprensione

aver usato almeno una volta l'editor notepad++

aver creato almeno una volta una pagina web, usando `<h1>` `<p>`

Una calcolatrice in javascript

È utile ricordare che il linguaggio HTML viene usato per creare il contenuto delle pagine web, il linguaggio CSS per modificarne l'aspetto esteriore, cioè la parte estetica, mentre il linguaggio Javascript è un vero linguaggio di programmazione, per l'elaborazione automatizzata dei dati. Quest'ultimo offre, quindi, la possibilità di aggiungere alla pagina capacità di calcolo e interattività con le attività svolte dall'utente.

Questi tre linguaggi possono essere **usati insieme** per creare le pagine web.

Le pagine web sono dei file di testo, di solito con estensione .htm oppure .html, che:

- vengono creati usando un editor di testo non formattato, come notepad++
- vengono visualizzate usando un browser, come Mozilla Firefox.

Come primo esempio scopriremo il funzionamento di una semplice calcolatrice che fa il prodotto di due numeri. Si tratta di un esempio apparentemente banale, ma utile come primo esempio.

Suggerimenti:

provare almeno una volta ad utilizzare il programma finito, fornito dal docente

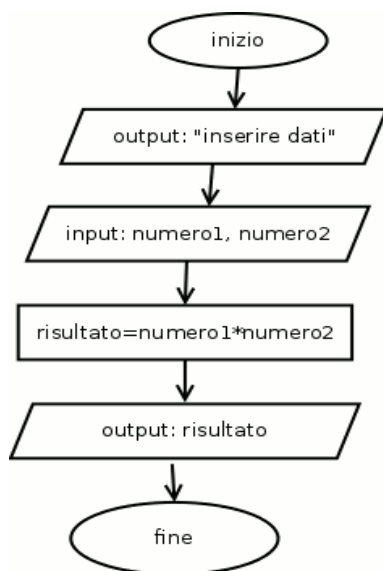
durante la lettura di questo documento, di tenere aperto il codice riportato in fondo a questo documento.



L'algoritmo potrebbe essere descritto con le seguenti parole:

1. chiedere di inserire i dati
2. leggere i dati dell'utente nelle variabili numero1 e numero2
3. memorizzare il risultato dell'operazione nella variabile risultato
4. visualizzare la variabile risultato

Oppure potrebbe essere rappresentato tramite il seguente diagramma di flusso:



Nella prima istruzione si chiede di visualizzare la frase "inserire dati" (scritta per questo motivo tra virgolette).

Nella seconda istruzione si ricevono i dati inseriti: **numero1** e **numero2** sono due variabili dove vengono memorizzati i dati inseriti dall'utente.

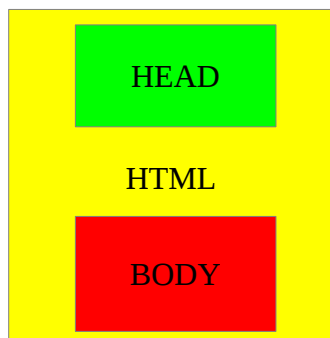
Nella variabile **risultato** viene memorizzato il risultato dell'operazione aritmetica (l'asterisco è il simbolo usato per indicare la moltiplicazione).

Ricordare quali sono i significati delle diverse forme usate in questo diagramma...

Domanda: quali sono i dati di input e quali di output?

La struttura di una pagina web

Sebbene il programma potrebbe essere realizzato, traducendo questo diagramma di flusso usando



esclusivamente il linguaggio Javascript, in questo esempio si userà anche il linguaggio HTML.

Solo le parti di input e di output possono essere create usando il linguaggio HTML, tutto il resto dovrà essere necessariamente in Javascript. Se si vuole rendere più gradevole l'aspetto della pagina si può aggiungere anche qualcosa in linguaggio CSS.

La pagina web ha una struttura formata da elementi contenuti uno all'interno dell'altro, la cui posizione è determinata da delle parole chiave chiamate **tag**.

Ecco come si presenta la struttura di una pagina vuota:

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Ogni parola chiave è chiamata "tag" (che significa etichetta) ed è racchiusa tra un simbolo minore e maggiore (chiamate anche parentesi angolari). Ogni tag di solito deve essere prima "aperto" e poi "chiuso". La chiusura si fa aggiungendo il carattere "slash" dentro il tag. In questo modo ogni coppia di tag (uno di apertura e uno di chiusura) racchiude una parte della pagina, un testo o altri tag. È preferibile scrivere i tag in minuscolo.

- i tag `<html>` racchiudono tutta la pagina
- i tag `<head>` racchiudono l'intestazione (o testata)
- i tag `<body>` racchiudono il corpo della pagina (dove si trova quello che si vede nel browser)

In questo esempio ogni tag si trova su una nuova riga, ma non è importante andare a capo.

Inoltre non è importante nemmeno il numero degli spazi che viene inserito, che è utilizzato solo per facilitare la lettura del codice.

Per sapere quali sono gli altri tag, e come si usano visitare:

http://www.w3schools.com/html/html_reference.asp

Creazione dell'area di input

Per prendere pratica con il linguaggio HTML, provare a creare una pagina che contiene solo il form con i campi di testo e il pulsante.

Per realizzare la frase "Inserire dati" si può usare il tag `<h1>`, mentre, per realizzare le caselle dove inserire i dati e il pulsante da cliccare, si possono usare i tag `<form>` e `<input>`. Questa parte dovrà essere visibile nel browser, quindi sarà a sua volta contenuta nel tag `<body>`.

```
<body>
  <h1>Inserire Dati</h1>
  <form>
    inserire primo numero <input>
    inserire secondo numero <input>
    <input>
  </form>
</body>
```

Domanda: se `<form>` è un tag "contenitore", allora quali sono i tag contenuti al suo interno?

Come si vede dal codice, mentre `<form>` è un tag contenitore (con tag di apertura e di chiusura), `<input>` **non è contenitore**, quindi **non si deve chiudere** (si dice che si auto-chiude).

Provare ad aprire la pagina appena creata usando il browser per verificare che i tag `input` sono visibili come delle caselle di testo, mentre il tag `<form>` non è un elemento visibile.

Rendere visibile un contenitore

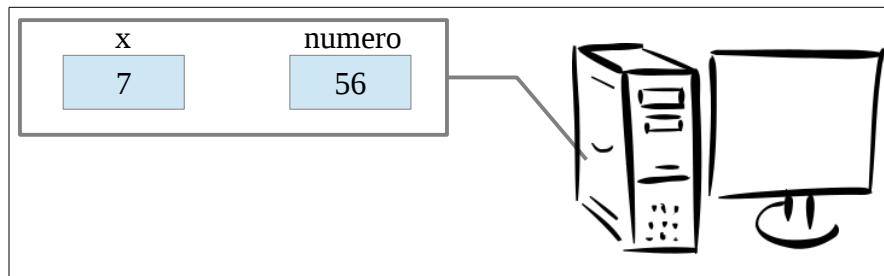
Per aggiungere elementi di decorazione, si può usare il linguaggio CSS. Poiché il CSS non aggiunge elementi visibili nel browser (come `<h1>` o `<input>`), ma solo una decorazione, il codice CSS può essere contenuto nel tag `<head>`. Il tag `<head>` conterrà `<style>`, che a sua volta conterrà le regole in CSS. Provare il seguente codice per vedere un bordo rosso:

```
<head>
  <style>
    form {
      border-style: solid;
      border-color: red;
    }
  </style>
</head>
```

Variabili

Dopo che l'utente ha inserito i valori questi devono essere memorizzati dentro la memoria del computer. Per farlo facilmente si usa il concetto di **variabile**, analogo a quello usato in matematica:

- una variabile è un contenitore per i dati;
- una variabile può contenere solo un valore alla volta, ma il valore può essere modificato;
- una variabile deve avere un nome e il suo nome è usato come riferimento ad un'area della memoria centrale.



Attributi in HTML

Rileggendo il diagramma di flusso visto prima, si vede che ci sono due variabili da inserire e che queste dovranno contenere due numeri. Se in HTML si usa `<input>` per inserire un elemento di input (un campo di testo), e se chiamiamo questi contenitori **n1** ed **n2**, allora il codice dovrà essere modificato, aggiungendo l'*attributo* name (in verde) dentro il tag `<input>`:

```
<body>
  <h1>Inserire Dati</h1>
  <form>
    inserire primo numero <input name="n1">
    inserire secondo numero <input name="n2">
    <input type="button">
  </form>
</body>
```

In questa immagine l'attributo è di colore verde, mentre il valore assegnato è di colore viola.

Da questo momento in poi, si aggiungeranno spesso anche altri attributi nei tag.

Nell'ultima riga dell'ultima figura, si può notare che dentro il tag `<input>` è possibile usare anche l'attributo type per specificare la forma di un elemento di input (un pulsante).

Aprire il file appena creato usando il browser e verificare la presenza di un nuovo pulsante.

Nella seguente tabella sono riassunti i principali attributi che si possono usare con `<input>`:

name	definisce il nome della variabile corrispondente a quell'elemento
type	definisce il tipo di elemento
value	definisce il valore contenuto nell'elemento

Gli attributi possono essere usati anche contemporaneamente, separandoli con uno spazio, quindi si possono aggiungere anche type e value: nel seguente codice, "text" significa che la casella è destinata a contenere testo.

Anche sul tag `<style>` si deve aggiungere un attributo `<style type="text/css">`, ma questo type ha un significato leggermente diverso.

```
<body>
  <h1>Inserire Dati</h1>
  <form name="due_numeri">
    inserire primo numero <input type="text" name="n1">
    inserire secondo numero <input type="text" name="n2">
    <input type="button" value="calcola">
  </form>
</body>
```

Fino a questo punto, si è utilizzato solo il linguaggio HTML e la pagina permette solo all'utente di inserire i numeri. Per poter elaborare i dati si dovrà utilizzare un altro linguaggio, il Javascript. La prima operazione da fare è quella di conoscere quali siano i valori contenuti negli elementi di input. In questo momento nel codice HTML ci sono due elementi: n1 e n2

variabile	HTML	
input	n1	
input	n2	

La pagina può contenere sia elementi del linguaggio HTML che elementi del linguaggio Javascript. È possibile copiare il valore contenuto negli elementi di input (HTML) in due nuove variabili in Javascript. Questo è necessario per poter elaborare i dati inseriti dall'utente.

variabile	HTML	Javascript
input	n1	numero1
input	n2	numero2

Domanda: dove si possono inserire le istruzioni in Javascript? Il tag <head> conterrà <script>, che a sua volta conterrà le istruzioni in Javascript.

Codice in Javascript

La prima istruzione, **function**, serve solo a dare un nome ad un gruppo di istruzioni per poi poterle utilizzare tutte insieme, ad esempio, quando si fa click con il mouse. Il nome della funzione **non deve contenere spazi o lettere accentate** e deve terminare con le **parentesi tonde**. Il gruppo di istruzioni dovrà essere racchiuso tra parentesi graffe.

La copia dei valori contenuti in n1 e n2, viene realizzata con un'istruzione di *assegnazione* (si chiamano così tutte le istruzioni che contengono il simbolo dell'uguale):

```
<head>
  <style type="text/css">
    form {
      border-style: solid;
      border-color: red;
    }
  </style>

  <script type="text/javascript">
    function funzioneProdotto()
    {

    }
  </script>
</head>
```

```
function funzioneProdotto()
{
    var numero1=document.forms["due_numeri"]["n1"].value;
```

- **var** definisce una nuova variabile in Javascript
- **numero1** è il nome della nuova variabile

A seconda dei gusti, l'istruzione di assegnazione si potrebbe leggere da sinistra verso sinistra:

- **value** significa "prendere" il valore di...
- **forms[...]["n1"]** specifica il campo input "n1" contenuto nel form di nome "due_numeri"
- **document** è un'oggetto che rappresenta il codice html della pagina attualmente aperta nella finestra del browser (che invece è un'oggetto chiamato **window**)

Perché queste parole sono sempre separate dal punto? Forse è troppo presto per spiegarlo...

Un programma che contenesse solo questa istruzione di assegnazione non farebbe nulla di utile per l'utente. Le due variabili vanno moltiplicate e poi il risultato va visualizzato:

```
function funzioneProdotto()
{
    var numero1=document.forms["due_numeri"]["n1"].value;
    var numero2=document.forms["due_numeri"]["n2"].value;

    risultato=numero1*numero2;
    window.alert("Risultato: " + risultato );
}
```

Scopriamo che non è obbligatorio scrivere var davanti ad una nuova variabile :)

Le variabili numero1 e numero2 contengono un testo composto da cifre, quindi, dal punto di vista formale, queste variabili andrebbero convertite in formato numerico, in questo modo, usando **parseFloat()** è una funzione che trasforma un testo in un numero in virgola mobile prima di effettuare l'operazione del prodotto.:

```
risultato=parseFloat(numero1)*parseFloat(numero2);
```

Infine va dedicata una spiegazione particolare all'ultima istruzione.

window è l'oggetto che rappresenta la finestra del browser

alert è una funzione (si riconosce dal fatto che il suo nome termina con le parentesi tonde)

La funzione **alert()** visualizza un messaggio di **avviso** all'utente (in questo caso serve solo a produrre l'output del risultato della moltiplicazione). È usata sull'oggetto window, perché blocca la finestra del browser, costringendo l'utente a cliccare su ok.

Dentro le parentesi tonde è specificato cosa deve essere visualizzato nell'avviso: la frase tra virgolette viene mostrata sempre esattamente come è scritta dentro le virgolette, seguita poi dal valore numerico **contenuto** nella variabile "risultato" che è senza le virgolette, e che ogni volta può contenere un risultato diverso. Il simbolo del più non effettua una somma tra numeri, ma è necessario per unire le due parti da visualizzare. Poiché unisce due parole per creare una sola frase, si chiama operatore concatenamento.

La frase tra virgolette si chiama anche string.

A questo punto, ci sono tutte le istruzioni necessarie, ma provando ad eseguire il programma, il click sul pulsante calcola non funziona...

Javascript è un linguaggio che utilizza gli oggetti "event".

Servono a rilevare le azioni svolte dall'utente, come i click del mouse, e a permettere di "reagire", e di eseguire eventuali istruzioni.

È possibile attivare gli eventi sugli elementi HTML usando specifici attributi.

Per attivare il click su un pulsante basta aggiungere l'attributo onclick seguito dalle istruzioni.

```
<input type="button" value="calcola!" onclick="funzioneProdotto()" >
```

A questo punto si può perfezionare l'istruzione

Segue il codice dell'intero programma (senza la parte CSS)

```
<html>
  <head>
    <script type="text/javascript">
      function funzioneProdotto()
      {
        var numero1=document.forms["due_numeri"]["n1"].value;
        var numero2=document.forms["due_numeri"]["n2"].value;

        risultato=parseFloat(numero1)*parseFloat(numero2);
        window.alert("Risultato: " + risultato );
      }
    </script>
  </head>

  <body>
    <h1>Inserire Dati</h1>
    <form name="due_numeri" >
      primo numero <input name="n1" type="text" >
      secondo numero <input name="n2" type="text" >
      <input type="button" value="calcola!"
onclick="funzioneProdotto()" >

    </form>
  </body>
</html>
```