

Javascript: funzioni

Codice HTML scritto nel file *animazione1.html*

```
<html><head>
  <title>animazione 1</title>
</style>
#Quad {
  width: 40px;
  height: 40px;
  position: absolute;
  border: 3px solid blue;
}
#container {
  width: 200px;
  height: 200px;
  position: relative;
  left: 20px;
  background-color: aqua;
}
</style>
<script type="text/JavaScript" src="animazione1.js">
</script></head>
<body>
  <h4> Funzioni con passaggio Parametri</h4>
  <form name="input" style="background-color: pink">
    scrivi quanti passi vuoi fare<br>
    poi scegli in che verso <br>
    <input type="text" name="T1" value="1" size="5" >
    <input type="button" name="BD" value="a destra" onclick="vai('D')">
    <input type="button" name="BG" value="in basso" onclick="vai('G')">
  </form>
  <div id="container">
    <div id="Quad"></div>
  </div>
  --><br><div style="background-color: pink"
    passi : <span id="des">0</span> pixel a destra,
            <span id="giu">0</span> in basso
  </div> </body></html>
```

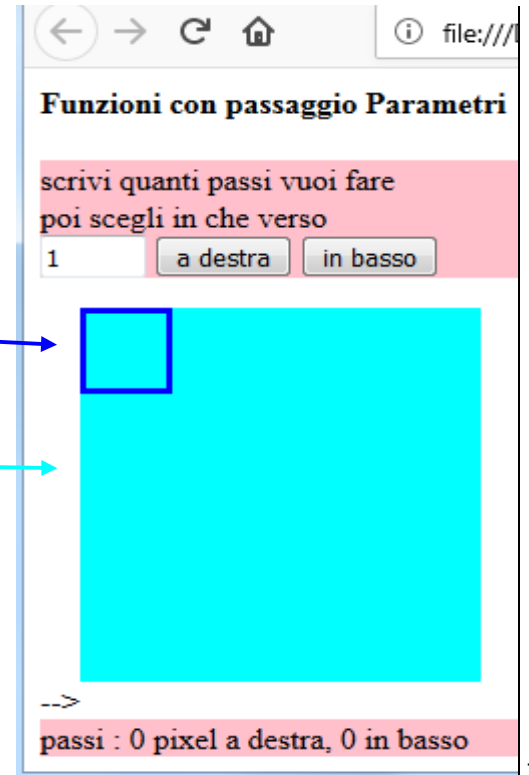


fig.1

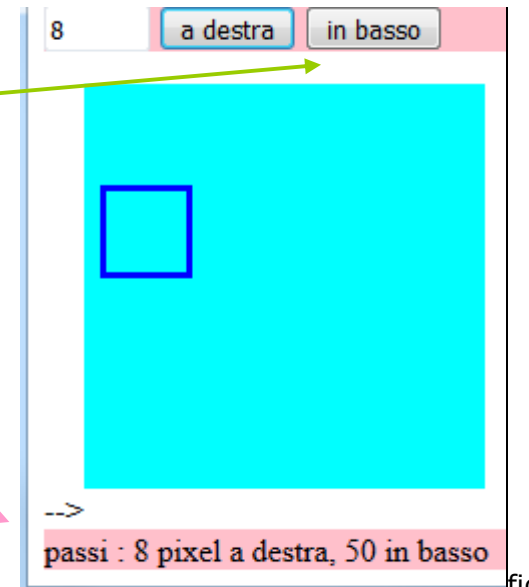


fig.2

Digitando 50 nella cella di input (fig.1), alla pressione del tasto **in basso** la pagina cambia aspetto: il quadrato blu **Quad** si abbassa di 50 pixel rispetto al quadrato color acqua **container**, digitando poi 8 nella cella di input, alla pressione del tasto **a destra** la pagina si presenta come in fig2: il quadrato blu **Quad** si sposta ora a destra di 8 pixel (e vengono aggiornati anche i contatori **passi**)

NOTA: il blocco **container** è posto ad una distanza di 20 pixel dal bordo sinistro della pagina (**left: 20px**); il blocco **Quad** è posto inizialmente a distanza 0px dal blocco che lo contiene (**container**) e i vertici in alto a sinistra si sovrappongono. Per realizzare l'animazione è necessario assegnare a **Quad** la proprietà **position: absolute** in modo che abbiano effetto le modifiche JS delle proprietà **top** e **left** (pag.2)

Javascript: funzioni

Codice Javascript scritto nel file *animazione1.js*

```
var quadrato, quanto;
var pixelDx = 0;
var pixelGiu = 0;

function incrementa (pix, q) {
    pix = pix + q;
    return pix;
}

function vaiGiu() {
    pixelGiu = incrementa (pixelGiu, quanto);

    quadrato.style.top = pixelGiu + 'px';

    document.getElementById("giu").innerHTML = pixelGiu;
}

function Adestra() {
    pixelDx = incrementa (pixelDx, quanto);

    quadrato.style.left = pixelDx + 'px';

    document.getElementById("des").innerHTML = pixelDx;
}

function vai(dove) {
    quadrato = document.getElementById("Quad");
    quanto = parseInt(document.input.T1.value);

    if (dove == "D"){
        Adestra();
    }
    else {
        vaiGiu();
    }
}
```

ATTENZIONE:

sia al click del tasto **in basso** che al click del tasto **a destra** viene richiamata e attivata la stessa funzione (**vai**).

Nel primo caso alla funzione **vai** viene passato il valore **G** (*parametro attuale*) che viene copiato dall'interprete JS nella variabile **dove** (*parametro formale*); nel secondo caso alla funzione **vai** viene passato il valore **D**.

La funzione **vai**, a sua volta, attiva la funzione **Adestra()** se la variabile **dove** coincide con il carattere **D**, altrimenti (**else**) attiva la funzione **vaiGiu()**.

Il questo esempio il codice JS è salvato in un file separato (*animazione1.js*) mentre gli stili sono scritti direttamente nella sezione *head* del file HTML (*animazione1.html*), delimitati dal tag **style** (aperto e chiuso).

Per questa pagina si sono utilizzati vari tipi di **funzioni**; nei linguaggi di programmazione come JS e C++ le **funzioni** sono porzioni di codice cui viene assegnato un nome (esempi: **vai**, **Adestra**, **vaiGiu**, **incrementa**). Le funzioni JS possono essere attivate dai gestori di eventi rilevati sulla pagina oppure da altre funzioni (il gestore **onclick** attiva la funzione **vai** che a sua volta esegue le funzioni **parseInt**, **Adestra** e **vaiGiu**, infine, le ultime due attivano, la funzione **incrementa**.

Le funzioni possono operare: su variabili *locali* definite all'interno del codice della funzione, su variabili *globali* definite all'esterno di ogni funzione (esempi: **quadrato**, **quanto**, **pixelDx**, **pixelGiu** sono variabili *globali* riconosciute da tutte le funzioni di questa pagina) oppure sugli eventuali *parametri* passati alla funzione (esempi: **pix**, **q**, **dove**).

Una funzione, infatti, può:

- **avere parametri in ingresso** indicati tra le parentesi tonde; esempi:

incrementa(pixelGiu, quanto)

in questo esempio i *parametri attuali* sono 2 variabili: **pixelGiu**, **quanto** i cui valori vengono copiati rispettivamente nei *parametri formali* **pix** e **q** invece per la funzione

vai('D') riportata a pag.1

il *parametro attuale* è una costante di tipo carattere: **'D'**

questo valore, il carattere **D**, viene copiato nel *parametro formale* **dove**

- **NON avere parametri in ingresso** (in questo caso la funzione è seguita da parentesi tonde vuote); esempi:

Adestra(), **vaiGiu()**

- **restituire un valore** alla funzione che la richiama (es: la funzione **incrementa** restituisce un valore (mediante l'istruzione **return pix;**) che viene utilizzato nella funzione chiamante a destra del simbolo di **assegnazione** (=) per modificare il valore di una variabile (es: **pixelDx**); anche **parseInt** è una funzione JS che restituisce il numero intero corrispondente alla stringa di caratteri digitati in **T1** (es: 50), numero che viene assegnato alla variabile **quanto**

- **NON restituire valore** (es: **vai**, **Adestra**, **vaiGiu**); in questo caso la funzione NON può essere assegnata ad una variabile.

La funzione **Adestra** modifica lo stile della *div* **Quad** impostando la distanza dal bordo sinistro (proprietà **left** dell'attributo **style** dell'oggetto **Quad** salvato nella variabile **quadrato**) della *div* **container** al numero di pixel digitati in input (variabile **quanto**), pixel da aggiungere a quelli già totalizzati e salvati nella variabile **pixelDx**. La funzione **vaiGiu** opera in modo analogo sulla proprietà **top** (distanza dal bordo superiore di **container**) e su **pixelGiu**.

La funzione **incrementa** effettua la somma e restituisce il risultato.