

Centrare un box nella pagina

I layout più diffusi sono quelli a 2 o 3 colonne, centrati o fluidi. Definire un layout come centrato sta ad indicare che, quando la finestra del browser viene ridimensionata, i contenuti del sito mantengono una larghezza fissa e restano al centro della finestra. Definire un layout fluido, invece, indica la caratteristica del sito di allargarsi e restringersi quando viene ridimensionata la finestra del browser.

Il sito di Repubblica è un tipico layout centrato.

<http://www.repubblica.it/>

Il sito di alberta è un classico esempio di layout fluido.

<http://test.alberta.org/>

Prima di arrivare a disegnare un layout complesso è però necessario imparare a gestire un singolo box. La prima cosa da imparare è centrare un box nella pagina web.

Abbiamo appena imparato che, in base alla tipologia di layout, possiamo avere un **box a larghezza fissa (layout centrato)** oppure a **larghezza mobile (layout fluido)**. Dobbiamo imparare a centrare il box in entrambi i casi. Il codice HTML sarà sempre lo stesso:

```
<body>  
  <div id="centrato">BOX CENTRATO</div>  
</body>
```

Creiamo una cartella nella quale produrremo il nostro codice. Diamo a questa cartella nome "mioSito". Dentro la cartella "mioSito" creiamo una cartella e diamole nome "style" (in questa cartella inseriremo il nostro codice css).

Il nostro file appena creato avrà il seguente codice:

```
<!DOCTYPEhtml>
<html>

  <head>
    <title>Centrare un box</title>
    <metahttp-equiv="Content-Type"content="text/html;charset=utf->
    <link rel="stylesheet" type="text/css" href="style/style_00.css">
  </head>

  <body>
    <div id="centrato">BOX CENTRATO</div>
  </body>

</html>
```

Centrare un box a larghezza fissa

Per fare in modo che un box con id="centrato" abbia una larghezza fissa (ad esempio 350 pixel) e si posizioni sempre al centro della nostra pagina web, dobbiamo usare il seguente codice CSS:

```
#centrato {  
    width:350px;  
    background-color:green;  
    margin-left:auto;  
    margin-right:auto;  
}
```

Il risultato è il seguente:



Abbiamo dunque imparato che dando una larghezza fissa ad un box (width:350px) e impostandone al valore "auto" i margini destro e sinistro (margin-left:auto; margin-right:auto;), il box mantiene la propria larghezza e si posiziona al centro dell'elemento che lo contiene, nel nostro caso il box è contenuto nell'elemento <body>.

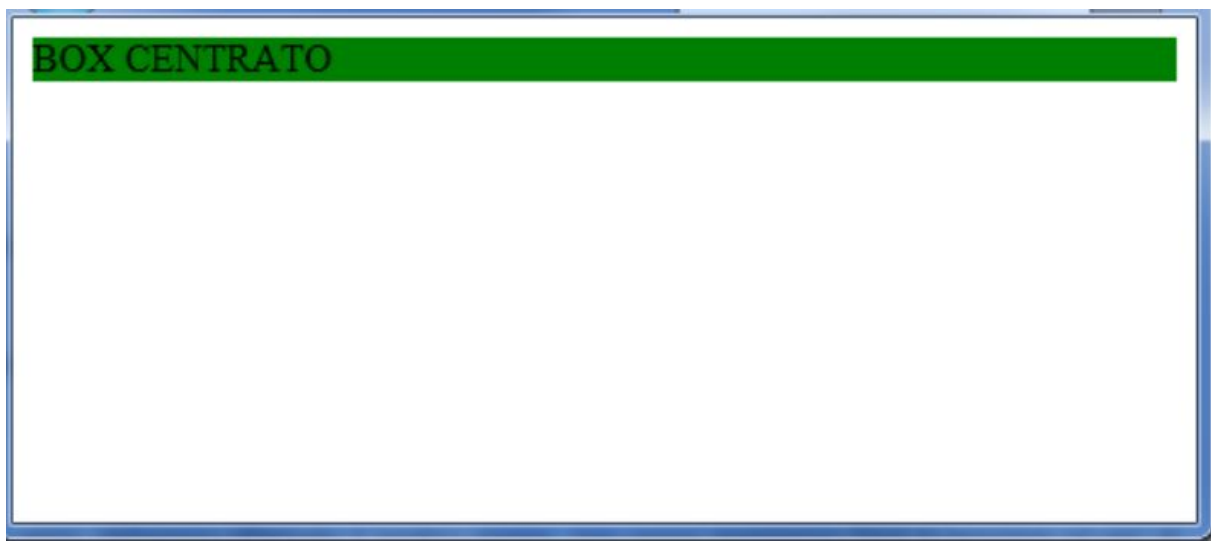
Centrare un box fluido

Ora proviamo a centrare un box fluido. Sappiamo che impostando i margini destro e sinistro ad “auto” teniamo il box al centro, quindi semplicemente eliminando la larghezza fissa, dovremmo già ottenere il nostro scopo.

Proviamo.

```
#centrato {  
    background-color:green;  
    margin-left:auto;  
    margin-right:auto;  
}
```

Il risultato ora è questo.



Sì, è vero, abbiamo ottenuto il nostro scopo, cioè, adesso il box si allarga in base alla larghezza della finestra del browser, ma come possiamo fare a lasciare un po' di spazio tra il nostro box e i margini della finestra del browser?

L'importante è non dare al box una larghezza fissa. Quindi ci basta dargli una larghezza in percentuale.

Ad esempio:

```
#centrato {  
    width:80%;  
    background-color:green;  
    margin-left:auto;  
    margin-right:auto;  
}
```

Che produce infatti il seguente risultato.



A differenza del box centrato a larghezza fissa, in quest'ultimo caso il box rimarrà al centro, ma non avrà una larghezza fissa, bensì essa aumenterà e diminuirà a seconda della dimensione della finestra.

Ora che siamo in grado di gestire un singolo box, possiamo pensare a gestirne due o più.

Layout a 2 colonne

Creare un layout a 2 colonne, significa creare almeno 2 elementi `<div>`.

Sappiamo però che il `<div>` è un elemento di tipo "block" e sappiamo che un elemento di questo tipo non ammette altri elementi né alla sua destra né alla sua sinistra perché occupa tutto lo spazio in cui si trova e costringe ad andare a capo. Come possiamo fare dunque a posizionare due elementi `<div>` sulla stessa "riga"?

La prima idea che viene in mente è quella di usare la proprietà "display" per trasformare i 2 `<div>` da "block" a "inline" e quindi consentirne un posizionamento affiancato. E in effetti l'idea funziona, ma... è l'idea sbagliata. Vediamo perché.

Creiamo i nostri 2 `<div>`:

```
<body>
```

```
  <div id="primaColonna">Questo è il div 1a colonna.</div>
```

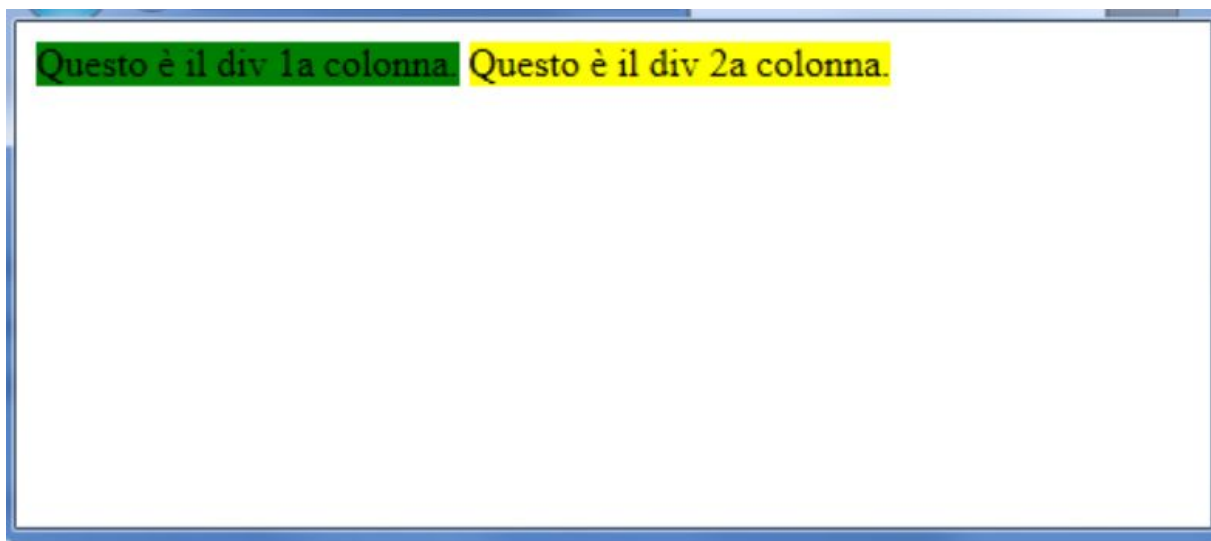
```
  <div id="secondaColonna">Questo è il div 2a colonna.</div>
```

```
</body>
```

e il nostro codice CSS:

```
#primaColonna{  
    display:inline;  
    background-color:green;  
}  
  
#secondaColonna {  
    display:inline;  
    background-color:yellow;  
}
```

Il risultato è il seguente.

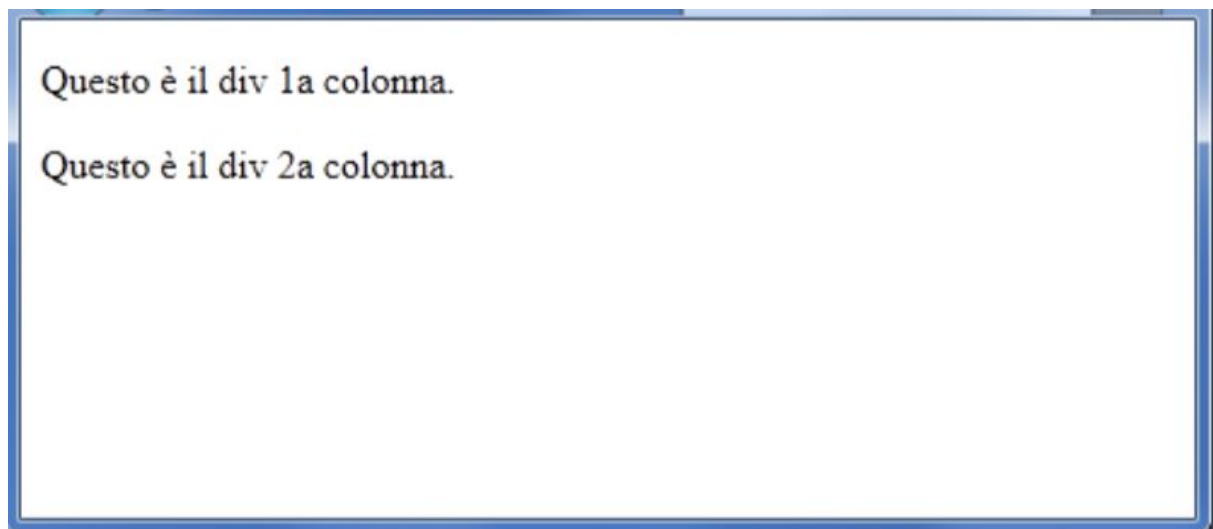


Cioè esattamente quello che volevamo. Quale è il problema?

Il problema è che quasi mai i 2 <div> avranno all'interno del semplice testo! La realtà dice che conterranno paragrafi, immagini, altri <div> è così via. Cosa succede se aggiungiamo elementi di tipo "block" all'interno dei nostri <div> trasformati in tipo "inline"? Basta provare per scoprirlo! Inseriamo ad esempio i 2 testi in due paragrafi.

```
<body>
  <div id="primaColonna">
    <p>Questo è il div 1a colonna.</p>
  </div>
  <div id="secondaColonna">
    <p>Questo è il div 2a colonna.</p>
  </div>
</body>
```

Ed ecco qua cosa mostra il browser.



No, l'idea di trasformare in "inline" i 2 <div> non può funzionare.

E allora?

Il segreto si chiama "float".

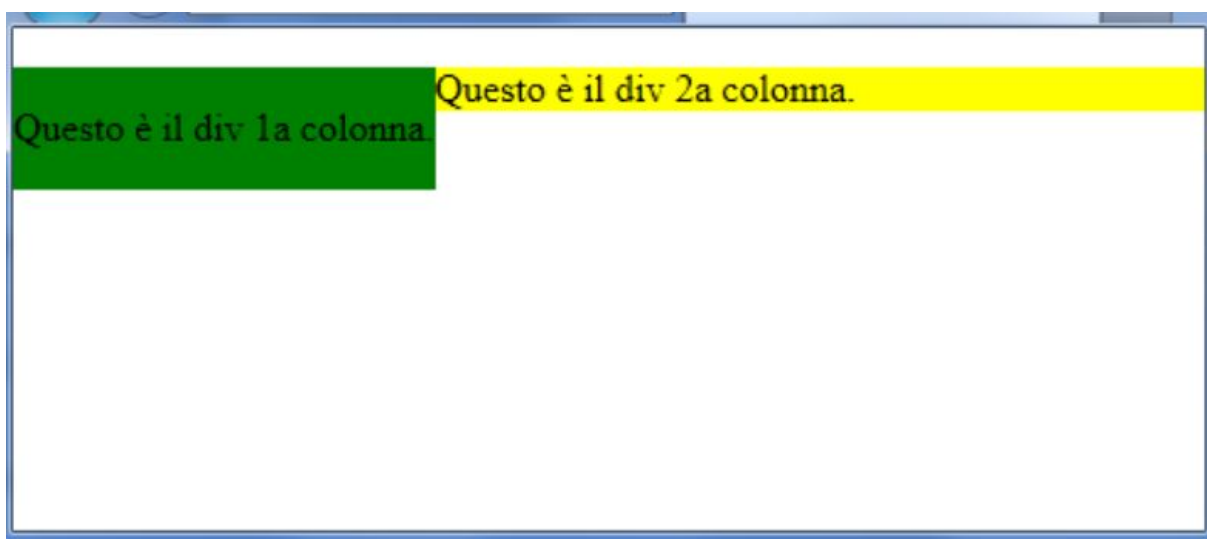
Abbiamo detto e qui ribadiamo che con questa proprietà è possibile rimuovere un elemento dal normale flusso del documento e spostarlo su uno dei lati (destra o sinistra) del suo elemento contenitore. Il contenuto che circonda l'elemento scorrerà intorno ad esso sul lato opposto rispetto a quello indicato come valore di float.

Quindi, se noi dichiariamo float:left il <div> #primaColonna, automaticamente il <div #secondaColonna si posizionerà alla sua destra! Sarà vero? Mano al codice. Lasciamo inalterato il codice HTML e modifichiamo come segue il codice CSS:

```
#primaColonna{  
    float:left;  
    background-color:green;  
}
```

```
#secondaColonna {  
    background-color:yellow;  
}
```

Ottenendo come risultato il seguente.

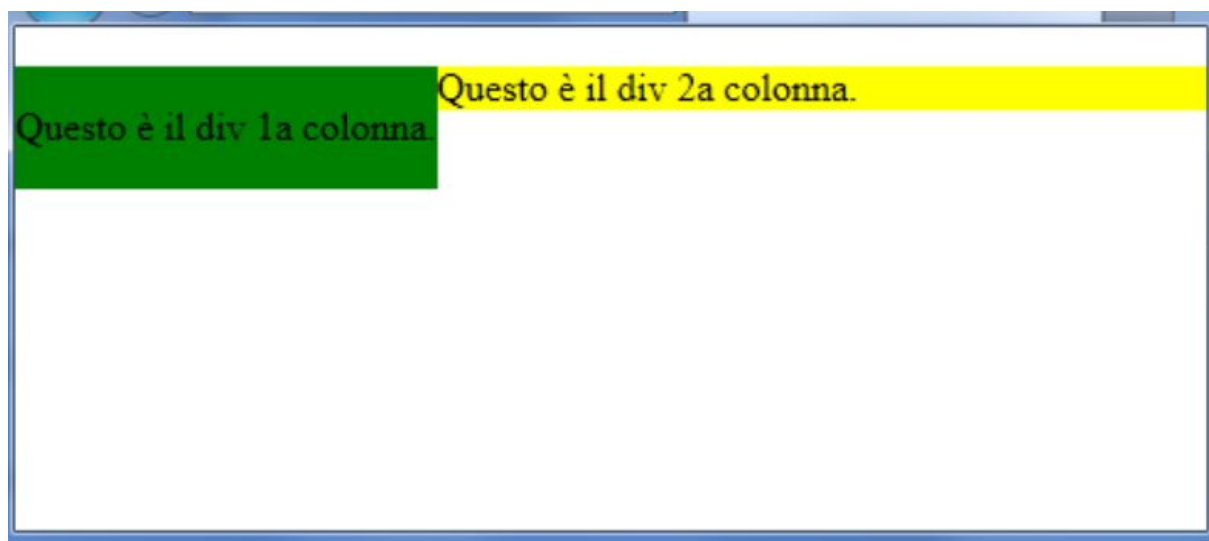


Sì! E' proprio quello che volevamo! Ma... perché i testi sono sfalsati?

La risposta è sempre nella proprietà "float". Questa proprietà infatti ha posizionato il `<div> #primaColonna` in alto a sinistra, costringendo il contenuto del secondo `<div>` a disporsi alla sua destra. Il contenuto! Non il secondo `<div>`. Il contenuto del secondo `<div>`.

E quale è il contenuto del `<div> #secondaColonna`? E' un elemento `<p>`. L'elemento `<p>` porta con se un margine superiore e inferiore ed è questo margine che crea il problema.

Osserviamo la seguente figura.



La linea blu mostra dove inizia il `<div> #secondaColonna`.

Osserviamo con attenzione. Ci saremmo aspettati che il `<div> #secondaColonna` iniziasse alla fine del `<div> #primaColonna`. Invece no! Inizia tutto a sinistra! E non è un errore!

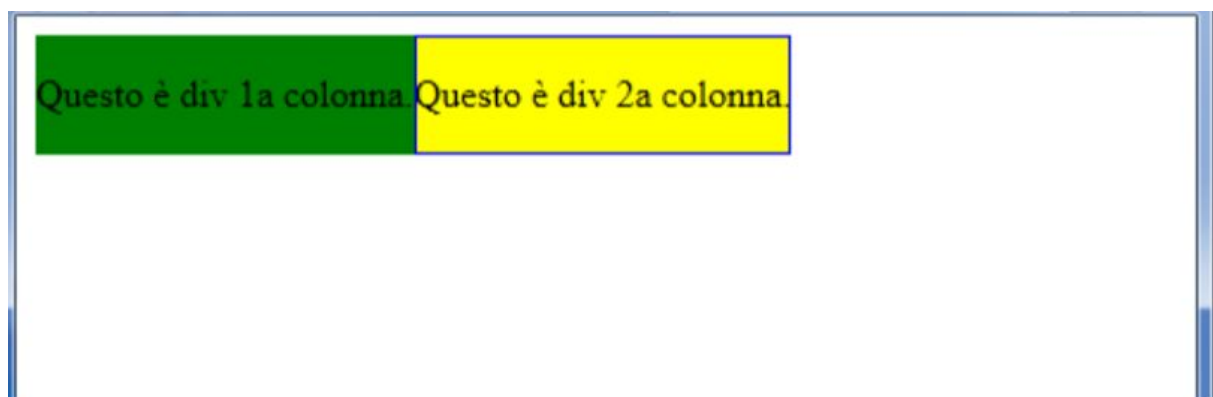
La proprietà "float" impostata sulla prima colonna infatti ha escluso la `#primaColonna` dal normale flusso della costruzione della

pagina. Quindi il `<div> #secondaColonna` la ignora e si comporta come se non esistesse. Questo non vale però per il contenuto della `#secondaColonna` il quale deve invece sottostare alle regole imposte dalla proprietà `float` impostata sulla `#primaColonna`. Tali regole impongono al contenuto della `#secondaColonna` di disporsi tutto intorno alla `#primaColonna`.

Ecco perché il codice che abbiamo scritto non funziona bene. Ma allora come facciamo ad allineare i testi?

Dobbiamo assegnare la proprietà `"float:left"` anche al `<div> #secondaColonna`! In questo modo la `#secondaColonna` sarà spostata anch'essa in alto a sinistra del suo box contenitore, ma troverà lo spazio già occupato dal `<div> #primaColonna` e dovrà quindi posizionarsi alla destra di quest'ultimo.

Osserviamo adesso la seguente figura e la linea blu intorno al box giallo. Come possiamo notare essa non inizia più dove iniziava prima, ma inizia alla fine del box verde. Questa cosa fa la differenza.

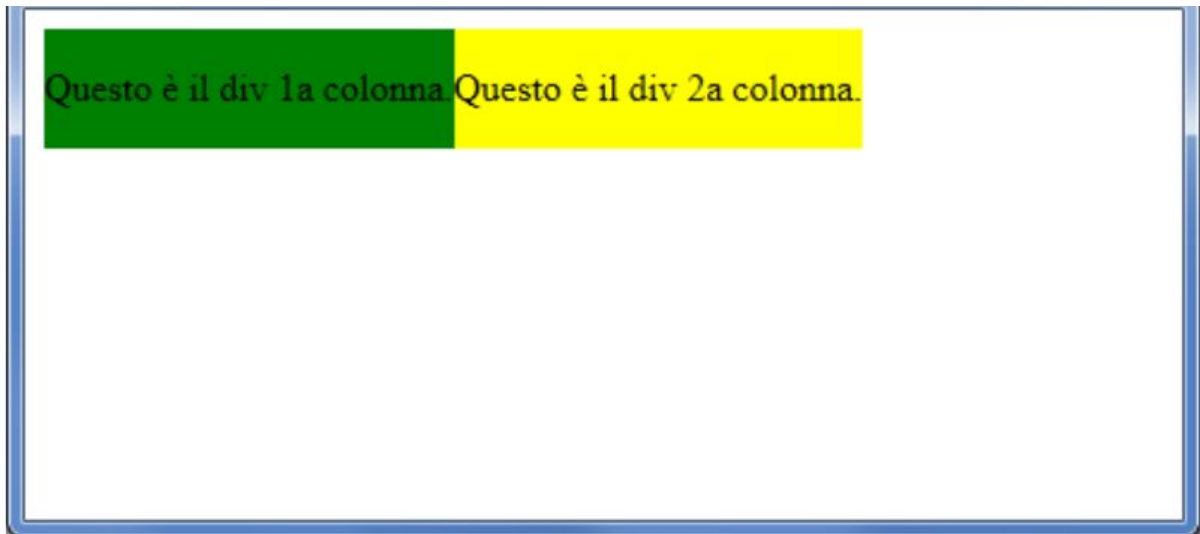


Alla fine il nostro codice CSS sarà quindi:

```
#primaColonna{
```

```
        float:left;  
        background-color:green;  
    }  
  
#secondaColonna {  
    float:left;  
    background-color:yellow;  
}
```

Con questo codice il risultato sarà alla fine.



E questa volta si! E' quello che vogliamo!

Ma non abbiamo ancora finito! Anzi, veramente dobbiamo ancora iniziare! Il nostro obiettivo infatti è realizzare un layout centrato nella finestra del browser con 2 colonne di larghezza fissa, oppure due colonne di larghezza variabile! Fino ad ora abbiamo solo messo 2 colonne una di fianco all'altra!

Due colonne a larghezza fissa centrate

Ora l'intuito dovrebbe iniziare a venirci incontro...

Sappiamo come centrare un box e sappiamo come mettere 2 box uno di fianco all'altro.

Unendo le due informazioni, dovremmo riuscire ad ottenere il nostro layout con 2 colonne a larghezza fissa centrate nella finestra del browser!

Includiamo quindi le nostre 2 colonne in un box che centreremo nella pagina del browser.

Daremo a questo box una larghezza fissa e lo stesso faremo con i due box `#primaColonna` e `#secondaColonna`.

```
<body>
```

```
    <div id="centrato">
        <div id="primaColonna">
            <p>Questo è div 1a colonna.</p>
        </div>
        <div id="secondaColonna">
            <p>Questo è div 2a colonna.</p>
        </div>
    </div>
```

```
<body>
```

Il codice CSS sarà:

```
#centrato {
```

```
width:900px;
margin-left:auto;
margin-right:auto;
}

#primaColonna{
width:250px;
float:left;
background-color:green;
}

#secondaColonna {
width:650px;
float:left;
background-color:yellow;
}
```

Il risultato è proprio lui!



Osserviamo come la larghezza del `<div> #centrato`, 900px, è (deve essere) uguale alla somma delle 2 larghezze dei 2 `<div> #primaColonna` (250px) e `#secondaColonna` (650px).

Due colonne a larghezza fluida centrate

Anche in questo caso l'intuito dovrebbe ormai indicarci la strada: sostituire i px con le percentuali! Quindi trasformare il nostro codice CSS, nel seguente:

```
#centrato {  
    width:80%;  
    margin-left:auto;  
    margin-right:auto;  
}
```

```
#primaColonna{  
    width:25%;  
    float:left;  
    background-color:green;  
}
```

```
#secondaColonna {  
    width:75%;  
    float:left;  
    background-color:yellow;  
}
```

Facciamo attenzione alle percentuali!

Il box #centrato sarà l'80% del suo contenitore che è l'elemento <body>.

Le 2 colonne sono invece contenute nel box #centrato e quindi saranno rispettivamente il 25% e il 75% del box #centrato.

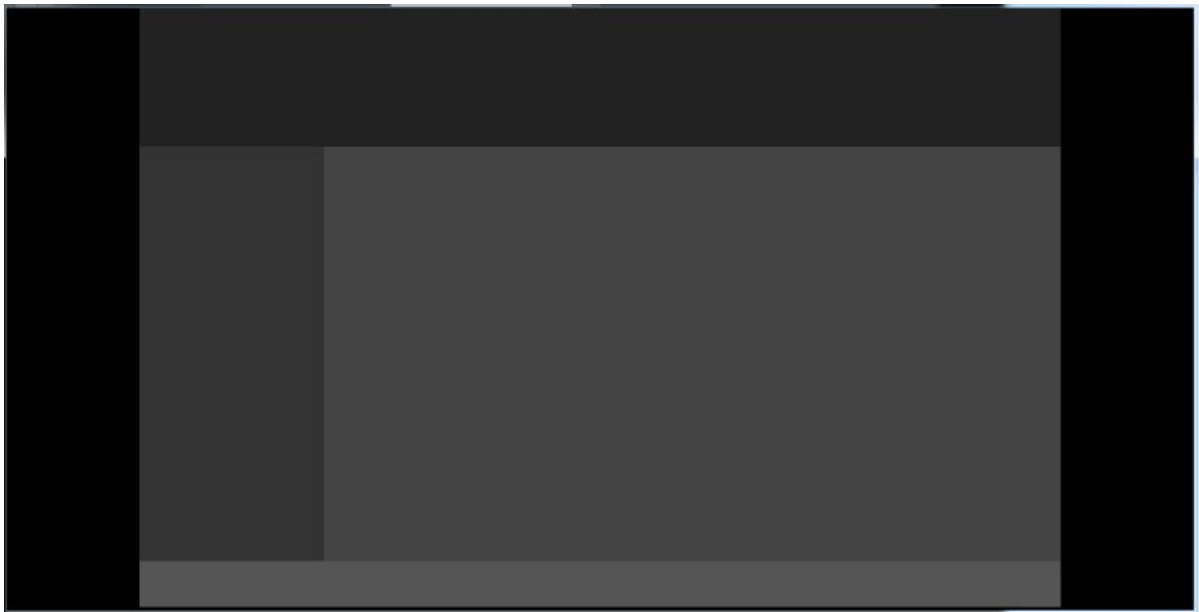
Il prossimo passo sarà quello di aggiungere al nostro layout un header e un footer.

Layout completo a 2 colonne fisse centrate

Ormai ci siamo! Per avere un layout a 2 colonne completo ci manca giusto la testata (header) e il piè di pagina (footer). Qui realizzeremo solo un layout centrato a larghezza fissa. Verrà lasciato al lettore l'implementazione del layout fluido (ossia la sostituzione dei pixel con le percentuali).

Alla fine del paragrafo saremo in grado di produrre il layout mostrato nella seguente

figura.



Vediamo subito il codice HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
      type="text/css"href="style/style_07.css" />
    <title>Layout a 2 colonne completo</title>
  </head>
  <body>
    <div id ="centrato">
      <div id="header"></div>
      <div id="primaColonna"></div>
      <div id="secondaColonna"></div>
      <div id="footer"></div>
    </div>
  </body>
</html>
```