

Realizzare il layout di un sito web senza utilizzare frame e tabelle

Questa guida descrive in maniera semplice e essenziale i passi necessari alla realizzazione di una pagina web strutturata in maniera professionale, evitando l'utilizzo di frame e tabelle.

Negli anni '90 sono fioccati moltissimi siti web divisi in *frame* grazie al fatto che questi oggetti riuscivano a fornire un buon grado di dinamicità alle pagine web. Difatti l'idea che nasceva dall'utilizzo dei frame era essenzialmente quella di mantenere una parte del sito, solitamente il menu, fissa e modificare esclusivamente un secondo frame, quello centrale, per visualizzare le varie pagine del sito.

Nonostante questa tecnica sia ancora oggi in voga, è da ritenersi superata per vari aspetti. Innanzitutto dal punto di vista stilistico non appare sempre gradevole un sito con frame, sono riconosciuti dei problemi di indicizzazione per tali siti e in ultimo luogo non tutti i browser supportano i frame, soprattutto se stiamo parlando di browser testuali.

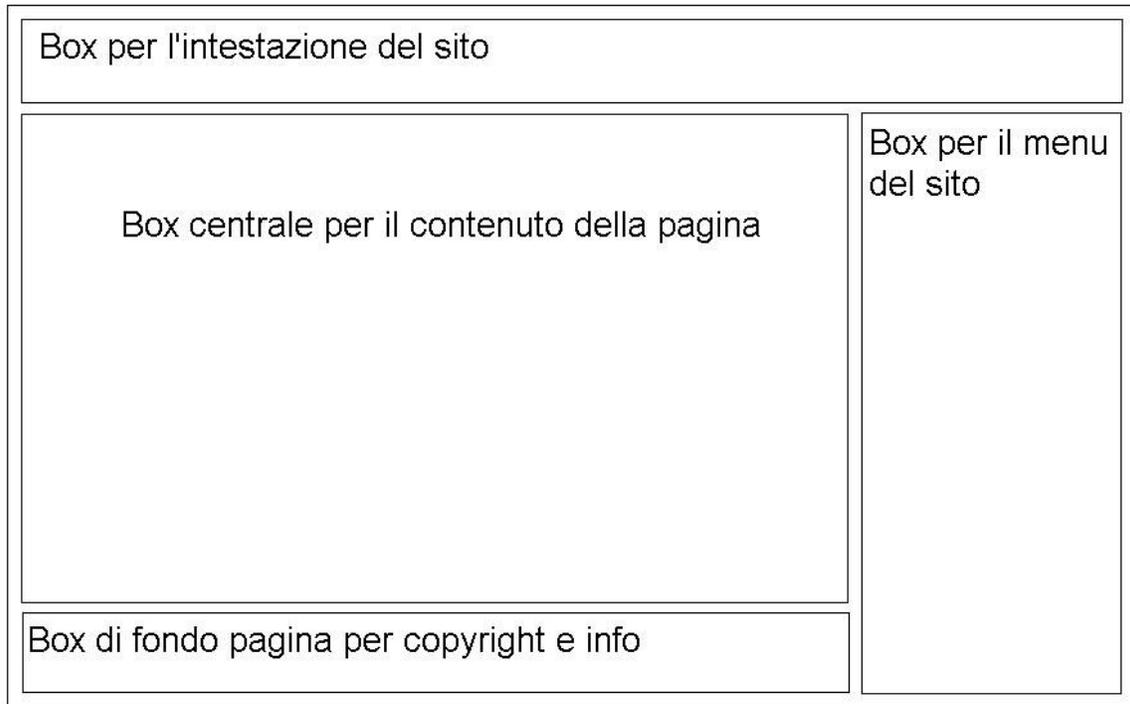
Successivamente i siti web si sono evoluti verso una struttura a *tabella*. Superati i problemi derivati dai frame per molto tempo tale struttura si è ritenuta la più efficiente e semplice per la realizzazione dei layout per siti web.

Ancora oggi sono presenti moltissime pagine sulla rete strutturate in tale modo. La necessità di sostituire una struttura a tabella con una struttura più efficiente nasce esclusivamente da una questione stilistica. In secondo luogo è giusto inoltre sottolineare che una tabella, seppur flessibile e di immediata realizzazione, non nasce come struttura portante di un sito.

Il layout che andremo a realizzare utilizzerà il concetto del box, cioè dei div.

La nostra pagina web sarà in sostanza realizzata tramite la disposizione ben precisa di div.

Ecco lo schema della struttura che implementeremo (fig 1).



(fig. 1)

La pagina html sarà suddivisa in 4 aree, ciascuna di queste sarà un div.

Realizzeremo a tale scopo due file :

- **index.html**
- **stile.css**

stile.css conterrà il foglio di stile necessario a disporre in modo corretto i 4 box.

Prima di vedere il codice sottolineiamo alcuni concetti importanti:

- 1) Verranno utilizzate misure in percentuale così da avere una pagina web a tutto schermo e flessibile in relazione alle diverse risoluzioni desktop.
- 2) Non saranno presenti tabelle o frame, ma verranno utilizzati in maniera massiccia le proprietà di margin e width per posizionare i div.
- 3) Il foglio di stile presentato non tiene conto dei dettagli legati a testo, colore, paragrafi ecc...

Il foglio di stile conterrà innanzitutto la seguente dichiarazione:

```
#intestazione
{
    margin: 0;
    padding: 10px 10px 10px 10px;
    border-bottom: 1px solid #cccccc;
}
```

Il box che conterrà il titolo del sito si estenderà in lunghezza per tutta la pagina senza margini, il contenuto avrà una rientranza di 10px su tutti e 4 i lati e infine il div sarà bordato esclusivamente in basso.

La seconda dichiarazione sarà:

```
#contenuto
{
    float:left;
    width: 75%;
    margin: 0;
    padding: 0 3% 0 10px;
}
```

Il div che conterrà il testo della pagina prenderà spazio per il 75% in larghezza rispetto alla finestra del desktop e sarà posizionato del tutto a sinistra.

La terza dichiarazione sarà:

```
#menu
{
    margin: 0 0 0 79%;
    padding: 0px;
    background-color: #eeeeee;
    border-left: 1px solid #ccc;
    border-bottom: 1px solid #ccc;
}
```

Questo div conterrà il menu del sito e avrà un colore diverso e una bordatura a sinistra e in basso. L'aspetto però più importante è margin che in questo caso denota come questo div sia del 79% spostato verso destra, o se vogliamo di come abbia un margine sinistro pari al 79% della finestra del browser.

E' interessante notare che il valore 79% nasce dal fatto che il box contenuto si estende per il 75% e a questo è aggiunto un padding del 3%, quindi inevitabilmente il menu sarà spostato almeno del 79% verso destra.

Infine sarà dichiarato il box di fondo:

```
#fondo{
    clear: both;
    border: 1px solid #cccccc;
    padding: 10px 10px 10px 10px;
    margin-top: 0px;
}
```

Ecco quindi il foglio di stile completo:

```
#intestazione
{
    margin: 0;
    padding: 10px 10px 10px 10px;
    border-bottom: 1px solid #cccccc;
}

#contenuto
{
    float:left;
    width: 75%;
    margin: 0;
    padding: 0 3% 0 10px;
}

#menu
{
    margin: 0 0 0 79%;
    padding: 0px;
    background-color: #eeeeee;
    border-left: 1px solid #ccc;
    border-bottom: 1px solid #ccc;
}

#fondo{
    clear: both;
    border: 1px solid #cccccc;
    padding: 10px 10px 10px 10px;
    margin-top: 0px;
}
```

Passiamo ora alla costruzione effettiva della pagina index.html del sito.

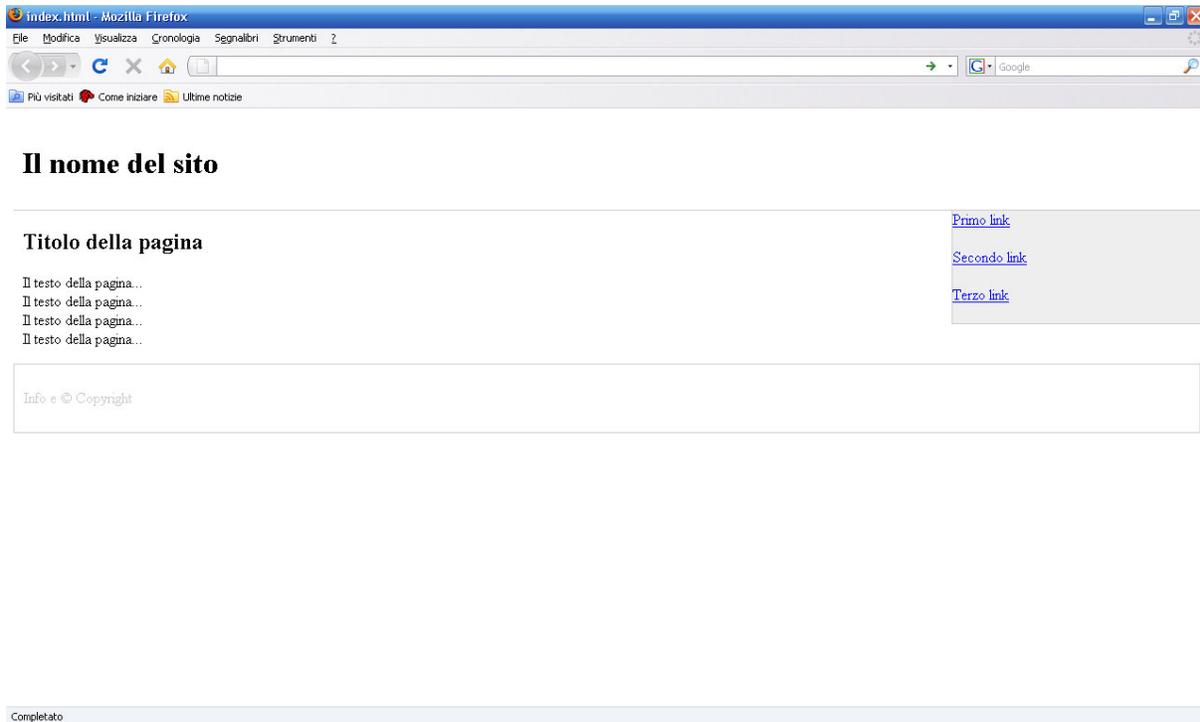
Ecco il codice:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Index.html</title>
<link rel="stylesheet" href="stile.css" type="text/css" />
</head>
<body>
<div id="intestazione">
  <h1>Il nome del sito</h1>
</div>
<div id="contenuto">
  <h2>Titolo della pagina</h2>
  <p>
    Il testo della pagina...<br />Il testo della pagina...<br />Il testo della pagina...<br />
    Il testo della pagina...<br />
  </p>
</div>
<div id="menu">
<p><a href="#">Primo link</a></p>
<p><a href="#">Secondo link</a></p>
<p><a href="#">Terzo link</a></p>
</div>
<div id="fondo">
<p>Info e &copy; Copyright</p>
</div>
<br />
</body>
</html>
```

Logicamente questa pagina sottolinea esclusivamente la struttura tralasciando lo stile legato ai font e ai colori.

Lo scopo che si prefigge è esclusivamente quello di fornire la struttura di partenza sulla quale realizzare un intero sito.

Ecco il risultato nel browser (fig. 2)



(fig. 2)

Concludiamo fornendo un'idea di implementazione dell'intero sito tramite la funzione include di php.

L'obiettivo è realizzare 3 file il cui contenuto non cambia mai in tutte le pagine dell'intero sito:

- **Il titolo** (titolo.html)
- **Il menu** (menu.html)
- **Il fondo** (fondo.html)

D'altronde, considerando il layout appena realizzato, è facile intuire come esclusivamente il box contenuto varierà di pagina in pagina.

Per evitare di copiare il menu, il titolo e il fondo in ciascuna pagina è possibile utilizzare l'include di php dopo aver realizzato i tre file sopraelencati.

Ad esempio il file menu.html potrebbe contenere:

```
<a href="pagina.php">Link a pagina</a><br>
<a href="pagina1.php">Link a pagina 1</a><br>
<a href="pagina2.php">Link a pagina 2</a><br>
<a href="pagina3.php">Link a pagina 3</a><br>
```

Quindi esclusivamente il menu, tralasciando i tags <body>, <html> ecc.

La nuova pagina html sarà innanzitutto con estensione **.php** e diventerà:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Index.php</title>
<link rel="stylesheet" href="stile.css" type="text/css" />
</head>
<body>
<div id="intestazione">
  <?php include("titolo.html") ?>
</div>
<div id="contenuto">
  <h2>Titolo della pagina</h2>
  <p>
    Il testo della pagina...<br />Il testo della pagina...<br />Il testo della pagina...<br />
    Il testo della pagina...<br />
  </p>
</div>
<div id="menu">
  <?php include("menu.html") ?>
</div>
<div id="fondo">
  <?php include("fondo.html") ?>
</div>
<br />
</body>
</html>

```

Così facendo la pagina **index.php** risulterà estremamente snella e flessibile, di facile lettura e soprattutto facilmente modificabile.

Realizzare così, tutte le pagine del nostro sito risulterà veloce e stilisticamente pulito. Basterà difatti cambiare il contenuto del div centrale per ottenere pagine sempre diverse in cui menu, titolo e fondo sono sempre uguali, col vantaggio aggiuntivo di poter modificare un solo file, ad esempio menu.html, per apportare modifiche a tutti i menu delle pagine.

In questa guida si è realizzato il layout per un sito web professionale, nel quale si è utilizzato il concetto del box (div) per la struttura, il css per lo stile e il linguaggio php per realizzare ogni singola pagina in maniera più efficiente.

Concludiamo sottolineando come la tecnica utilizzata sia esclusivamente una delle tante nel campo della progettazione web.

L'obiettivo era innanzitutto evitare l'utilizzo di strutture superate, come frame e tabelle e descrivere quindi una strada semplice ed estremamente efficiente per realizzare un sito che rispetti gli standard qualitativi del momento.