

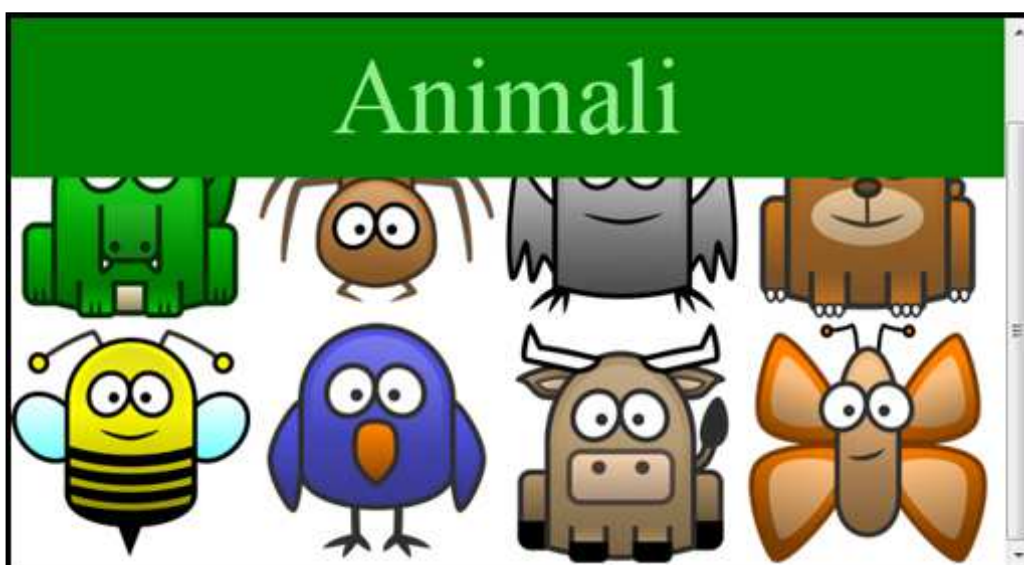
## 1) Cominciamo dalle cose facili

Per prima cosa ti sfido a realizzare il sito che hai visto nelle slide.

Ti ricordo che l'obiettivo è quello di realizzare un sito in cui la banda in alto - che rappresenta il titolo - resti sempre al suo posto.



Il contenuto della pagina deve scorrere sotto questa banda verde.



Tutto il codice lo puoi leggere dalle slide, le immagini invece le trovi qui: [www.bbuio.it/varie/31-animali.zip](http://www.bbuio.it/varie/31-animali.zip)

Sono sicuro che ce la farai!

## 2) Mettiamoci a cucinare

Crea la cartella **30-ricette** con le sottocartelle **css** e **img**.

L'obiettivo è di realizzare un sito come questo rappresentato in figura.



Cominciamo con l'operare i task più semplici.

- recupera il file **reset.css** e mettilo nella cartella **css**
- crea un file **stili.css** vuoto e mettilo nella stessa cartella
- scarica il file **www.bbuio.it/varie/ricette.jpg** e mettilo nella cartella **img**
- crea il file **index.html** e riempi la sezione **head** così come abbiamo imparato

Vediamo in quali sezioni si divide la pagina.



- La parte in alto (**header**)
- La parte centrale (**nav**)
- La parte finale (**article**)

index.html

```

09. <body>
10. <header>
11.     <!-- qui ci vanno la scritta "RICETTE" e l'immagine della bimba -->
12. </header>
13. <nav>
14.     <!-- qui ci vanno le voci di menu -->
15. <nav>
16. <article>
17.     <!-- qui ci vanno i contenuti -->
18. </article>
19. </body>
  
```

Vediamo adesso le sezioni in cui si divide l'**header**.



- La parte di sinistra (un **div** di classe "sinistra") per la scritta
- La parte di destra (un **div** di classe "destra") per l'immagine

```

10. <header>
11.     <div class="sinistra">RICETTE</div>
12.     <div class="destra"></div>
13. </div>
14. </header>
  
```

Cominciamo a ragionare sul css.

Per prima cosa osserviamo che il **body** è sempre centrato orizzontalmente ed è leggermente distanziato dal margine superiore della finestra.

Inoltre ricordiamoci che è buona prassi stabilire nel **body** la dimensione di base dei testi, nonché il tipo di font.

## stili.css

```
01. body {
02.     width:800px;
03.     margin: 15px auto;
04.     font-family: monospace;
05.     font-size: 10px;
06. }
```

Poiché dentro l'**header** ci sono due contenitori (**div**) dovranno essere disposti secondo qualche strategia di posizionamento particolare.

Stabiliamo di realizzare due blocchi flottanti: il primo a sinistra (grande il 75%) e l'altro a destra (grande il 25%).

Da ciò se ne deduce che il loro contenitore (**header**) deve avere la dichiarazione **overflow:auto**.

Parliamo adesso di ciò che sembra ovvio.

- L'**header** è grande il 100% (così come il **nav** e l'**article**)
- Lo sfondo dell'**article** (e del **nav**) viene preso (con **colorpix**) dall'immagine
- L'intestazione è alta 200 pixel
- Il testo è centrato verticalmente e orizzontalmente nel **div** di sinistra
- L'immagine deve essere centrata verticalmente e orizzontalmente nel **div** di destra

Quest'ultima osservazione ci rimanda a quanto visto nello slideshow che ci suggerisce - per centrare un'immagine in un contenitore - dobbiamo usare un doppio contenitore con le proprietà **display** prima a **table** e poi a **table-cell**.

## index.html

```
10. <header>
11.     <div class="sinistra">RICETTE</div>
12.     <div class="destra">
13.         <div class="contenitore_img">
14.             
15.         </div>
16.     </div>
17. </header>
```

## stili.css

```

07. header {
08.     background-color: #451805;
09.     width: 100%;
10.     overflow: auto;
11. }
12. div.sinistra {
13.     float: left;
14.     height: 200px;
15.     width: 75%;
16.
17.     line-height: 200px;
18.     color: white;
19.     font-size: 10em;
20.     text-decoration: underline;
21.     font-weight: bold;
22.     letter-spacing: 10pt;
23.     text-align: center;
24. }
25. div.destra {
26.     float: right;
27.     height: 200px;
28.     width: 25%;
29.     display: table;
30. }
31. div.contenitore_img{
32.     display:table-cell;
33.     vertical-align:middle;
34.     text-align:center;
35. }

```

Prendiamo in esame l'elemento **nav**.

## index.html

```

19. <nav>
20.     <a href="antipasti.html">Antipasti</a>
21.     <a href="primi.html">Primi</a>
22.     <a href="secondi.html">Secondi</a>
23.     <a href="contorni.html">Contorni</a>
24.     <a href="desert.html">Dessert</a>
25. </nav>

```

Il **nav** occupa il 100% dello spazio a disposizione e contiene tanti elementi **a**. Questi elementi sono centrati al suo interno.



Tutti questi elementi **a** devono apparire come dei pulsanti tutti della stessa dimensione.

Inoltre vogliamo impostare a questi pulsanti anche un valore per le proprietà **padding** e la proprietà **margin**.

Gli elementi **a** non sono elementi contenitori e per poter validare proprietà come queste devono essere mostrati come elementi **inline-block**.

#### stili.css

```
36. nav {
37.     width: 100%;
38.     background-color: #451805;
39.     text-align: center;
40. }
41. a {
42.     color: white;
43.     width: 100px;
44.     display: inline-block;
45.     padding: 10px;
46.     font-weight: bold;
47.     text-decoration: none;
48.     margin-left: 10px;
49.     font-size: 1.3em;
50. }
```

Infine, quando l'utente si sposta con il mouse sopra uno di questi pulsanti questi cambiano colore.

```
51. a:hover {
52.     background-color: white;
53.     color: #502310;
54. }
```

Per finire, occupiamoci dell'ultimo elemento della pagina: l'**article** preposto a presentare i contenuti della pagina.

Ai fini di questo esercizio ci basta indicare un contenuto alto 500 pixel con un bordo colorato dello stesso colore dello sfondo degli elementi precedenti.

```
55. article{
56.     height: 500px;
57.     background: white;
58.     border: solid 10px #451805;
59.     border-top: none;
60. }
```

### 3) L'ultimo esercizio

Quest'ultimo esercizio è la scusa per mettere insieme un po' di cose e vedere qualche piccola novità.

Il nostro obiettivo è realizzare una tipica intestazione di un sito.



Mentre la banda grigia si estende a tutta la larghezza della finestra il sito si sviluppa verticalmente in quella sezione indicata dal menu interno.

Pertanto avviciniamoci alla parte principale e cominciamo a realizzare quanto segue.



L'inizio è sempre lo stesso:

- crea la cartella **31-intestazione**
- crea in essa il file **index.html** e riempi con la struttura di un file html
- riempi come al solito la sezione **head**
- crea la sottocartella **css** con i file
- copia in essa il file **reset.css**
- crea un file (vuoto) **stili.css**

A questo punto riempi il body come indicato qui di seguito

```

index.html
10.     <header>
11.         <h1>TECNOLOGIE WEB</h1>
12.         <h2>Una raccolta di siti</h2>
13.         <section>
14.             <nav id="interno">
15.                 Tutti
16.                 Alcuni
17.                 Altri
18.             </nav>
19.         </section>
20.     </header>

```

L'unica osservazione che dobbiamo fare al codice precedente è il fatto che abbiamo voluto dare un **id** al **nav**.

Questo avviene perchè (sebbene al momento non ce ne occupiamo) abbiamo un altro menu in alto a destra ed è quindi doveroso distinguere i due menu.

Ovviamente, con il foglio di stile vuoto e il reset impostato, il browser non fa altro che stampare a video i testi indicati nel documento html tutti con lo stesso stile.

Cominciamo dunque ad editare il file **stili.css** e descriviamo la banda grigia senza limiti di ampiezza ma con un'altezza pari a 200 pixel.

#### stili.css

```
01. header {
02.     background-color: lightgrey;
03.     height: 200px;
04. }
```

Successivamente, andiamo a definire lo stile di **h1**.

Supponiamo il sito predisposto per un monitor con una risoluzione orizzontale di 800 pixel.

```
05. h1 {
06.     width: 800px;
07.     margin: auto;
08.     font-size: 50px;
09. }
```

Ci troviamo con una pagina web come quella che segue.



Decidiamo allora di distanziare l'elemento **h1** dal bordo superiore del suo contenitore con la proprietà **padding-top**

Modifichiamo come segue la regole per **h1**.

```
05. h1 {
06.     width: 800px;
07.     margin: auto;
08.     font-size: 50px;
09.     padding-top: 15px;
10. }
```



Similarmente facciamo con il sottotitolo.

```
11. h2 {
12.     width: 800px;
13.     margin: auto;
14.     font-size: 25px;
15. }
```



La sezione che contiene il **nav** interno sarà evidentemente anch'essa grande 800 pixel e centrata.

```
16. section {
17.     width: 800px;
18.     margin: auto;
19. }
```

Questa regola ovviamente non è che apporti molto alla nostra pagina web.

Vediamo dunque la regola per il menu interno che invece cambia radicalmente le cose.

```
20. nav#interno {
21.     width: 100%;
22.     height: 75px;
23.     border-radius: 10px 10px 0px 0px;
24.     background: linear-gradient(to bottom, black, white);
25.
26.     color: white;
27.     font-size: 25px;
28.     text-align: center;
29.     line-height: 75px;
30. }
```

Le prime quattro proprietà di questa regola si occupano del contenitore **nav** e le ultime quattro dei testi in esso contenuti.

Per prima cosa ci siamo presi cura dello spazio occupato: tutto (orizzontalmente) e 75 pixel (verticalmente).

Poi abbiamo definito l'aspetto degli angoli (il primo - in alto a sinistra - e il secondo - in alto a destra - hanno una curvatura di 10 pixel; gli altri sono ad angolo retto).

Infine ci siamo occupati dello sfondo (con gradiente verticale).

L'unica proprietà interessante relativamente al testo è **line-height**: quando il suo valore è idfentico al valore dell'altezza del contenitore (come in questo caso) il testo risulta centrato verticalmente.

# TECNOLOGIE WEB

Una raccolta di siti

Tutti Alcuni Altri

Per prima cosa notiamo che il sottotitolo è troppo addossato al menu: decidiamo di intervenire mediante la proprietà **padding-bottom** nella regola per **h2**, che viene modificata così:

```
11. h2 {
12.     width: 800px;
13.     margin: auto;
14.     font-size: 25px;
15.     padding-bottom: 15px;
16. }
```

Adesso occupiamoci dello spazio sotto il menu che è rimasto grigio.

Decidiamo di coprirlo con un **div** creato appositamente.

index.html

```
19. <div id="vuoto"></div>
```

Mi raccomando! Questo **div** va collocato dentro la **section** dopo il **nav**.

Per stabilire l'altezza di questo **div** dobbiamo fare due conti:

- l'**header** è alto 200 pixel
- l'**h1** è alto 50 pixel ed ha un **padding** superiore di 15 pixel
- l'**h2** è alto 25 pixel ed ha un **padding** superiore di 15 pixel
- il **nav** è alto 75 pixel

Pertanto:  $200-50-15-25-15-75=20$ . Venti pixel.

Ecco quindi la regola che ci serve.

stili.css

```
32. div#vuoto {
33.     width: 100%;
34.     height: 20px;
35.     background: white;
36. }
```

E siamo giunti a questo punto.



Come distribuire i tre testi equamente nello spazio a disposizione?

Inseriamo ciascun testo in un **div** apposito e creiamo una regola affinché questi tre nuovi contenitori si dividano lo spazio a disposizione.

#### index.html

```
15.         <div class="figlioDiNavInterno">Tutti</div>
16.         <div class="figlioDiNavInterno">Alcuni</div>
17.         <div class="figlioDiNavInterno">Altri</div>
```

#### stili.css

```
37. div.figlioDiNavInterno {
38.     width: 30%;
39.     display:inline-block;
40. }
```

Chiaramente è indispensabile la dichiarazione che modifica la proprietà **display** perchè altrimenti i tre contenitori sarebbero apparsi uno in testa all'altro.



Abbiamo quasi finito questa parte.

Poiché l'oggetto su cui abbiamo lavorato fino ad ora è un **nav** mè verosimile che al suo interno vi siano dei **link**.

Noi non sappiamo dove conducono questi link (e nemmeno ci interessa ai fini dell'esercizio) ma ci piacerebbe che reagissero alla presenza del mouse.

#### index.html

```
15.         <div class="figlioDiNavInterno"><a href="">Tutti</a></div>
16.         <div class="figlioDiNavInterno"><a href="">Alcuni</a></div>
17.         <div class="figlioDiNavInterno"><a href="">Altri</a></div>
```

#### stili.css

```
41. a {
42.     color:white;
43.     text-decoration: none;
44. }
45. a:hover {
46.     color:red;
47. }
```

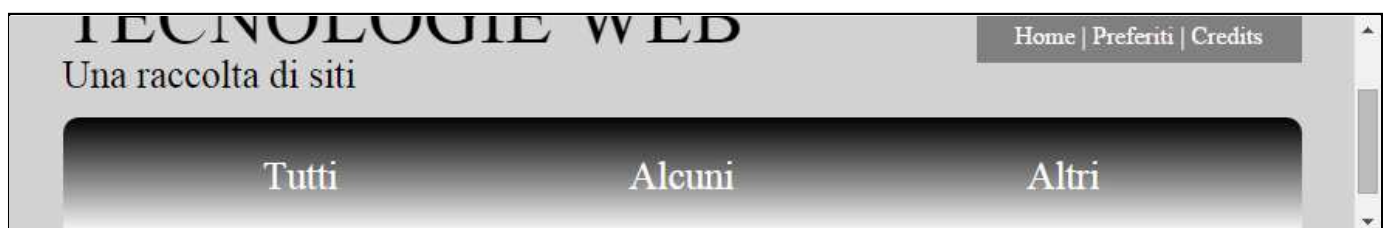
Con le ultime modifiche abbiamo ottenuto questo.



Non resta che piazzare il menu in alto a destra.



Questo menu - anche se dall'immagine non si vede - è uno di quei menu che vogliamo resti in primo piano anche scorrendo la pagina web.



Andiamo quindi a modificare il nostro documento **index.html** in modo da inserire una nuova **section** (con il **nav** principale) prima di ogni altra cosa.

index.html

```
10.     <section id="fissa">
11.         <nav id="principale">
12.             Home | Preferiti | Credits
13.         </nav>
14.     </section>
```

Abbiamo corredato entrambi i nuovi elementi con un id che ci permetterà di distinguerli dagli altri.

Per prima cosa cominciamo a rendere fissa la **section** ed agganciarla al bordo superiore.

stili.css

```
48. section#fissa {
49.     position: fixed;
50.     top: 0px;
51. }
52. nav#principale {
53.     width: 200px;
54.     height: 20px;
55.     background: grey;
56.     color: white;
57.     text-align: center;
58.     padding: 5px;
59. }
```

Anche se non si trova ancora dove vogliamo il nostro menu adesso resta fisso quando si scrolla la pagina.

Ma come fare per spostarlo dall'altra parte?

Dobbiamo ragionare in due volte.

Per prima cosa dobbiamo dare alla **section** delle caratteristiche che hanno tutti gli altri contenitori principali.

```
51.     width: 800px;
52.     right: 0px;
53.     left: 0px;
54.     margin: auto;
```

Aggiungendo queste dichiarazioni alla regola per l'elemento **section** con id **fissa** stiamo impostando una larghezza di 800 pixel (con la prima dichiarazione) e centrando la stessa in mezzo alla finestra (con le altre tre dichiarazioni).

Nota che - diversamente dagli elementi non fissi - qui abbiamo le proprietà **left** e **right** da porre a zero.

Adesso possiamo dire che l'elemento **nav** con **id** pari a **principale** (figlio di questa **section**) si deve collocare nel primo posto a destra disponibile.

```
63.     float: right;
```

Per chiudere il tutto non resta che far diventare link le tre voci presenti nel **nav** e verificare che diventino rossi al passaggio del mouse.

Ti lascio questa attività per esercizio, tanto ormai, dovresti aver capito come fare, no?

Finito?

Bene.

Ora ti seguirò in quest'ultima parte.

Il nostro obiettivo è dare a questo elemento l'aspetto di una tendina in sovrapposizione.

Ricordiamo che questo oggetto resta sempre lì, anche quando scorriamo la pagina.

Cominciamo col distanziare la banda grigia dal bordo superiore della finestra.

Non tanto però: quel poco che basta a far capire che il nostro piccolo menu è in sovrapposizione.

Aggiungiamo dunque un piccolo **div** di **25 pixel** subito prima della banda grigia.

index.html

```
16.     <div id="sostitutoNav"></div>
```

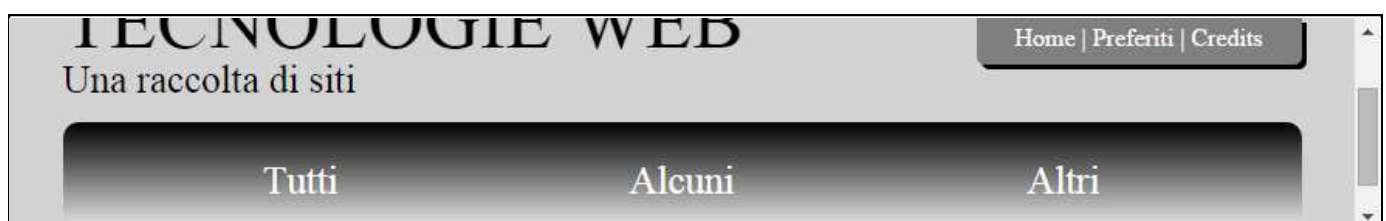
stili.css

```
65. div#sostitutoNav {
66.     height: 25px;
67. }
```



Sempre con l'obiettivo di rendere questo oggetto un aspetto particolare andiamo a modificarne i bordi ed aggiungere un'ombra.

```
68. nav#principale {
69.     border-radius: 0px 0px 5px 5px;
70.     box-shadow: 3px 3px 0 black;
71. }
```



E ancora: rendiamo questa tendina meno visibile con la proprietà **opacity** posta a **0.5** per ripristinarla al valore **1** quando l'utente la raggiunge con il mouse.

```
72. nav#principale {
73.     opacity: 0.5;
74. }
75. nav#principale:hover {
76.     opacity: 1;
77. }
```

Infine con la proprietà **transform** facciamo in modo che questo cambio di **opacity** avvenga in un secondo.

Concludendo presentiamo l'intera regola per il **nav** di classe **principale**.

```
56. nav#principale {
57.     width: 200px;
58.     height: 20px;
59.     background: grey;
60.     color: white;
61.     text-align: center;
62.     padding: 5px;
63.     float: right;
64.     border-radius: 0px 0px 5px 5px;
65.     box-shadow: 3px 3px 0 black;
66.     opacity: 0.5;
67.     transform: opacity 1s;
68. }
```