

Box model e Javascript /1

07

1) Rivediamo il progetto di Amnesty International

Comincia col fare una copia del progetto **13-amnesty** e chiamalo **27-amnesty** (cavolo! ne è passato di tempo!).

In effetti, se guardi il codice del file **index.html** vedrai che ne abbiamo imparate di cose!

index.html

```

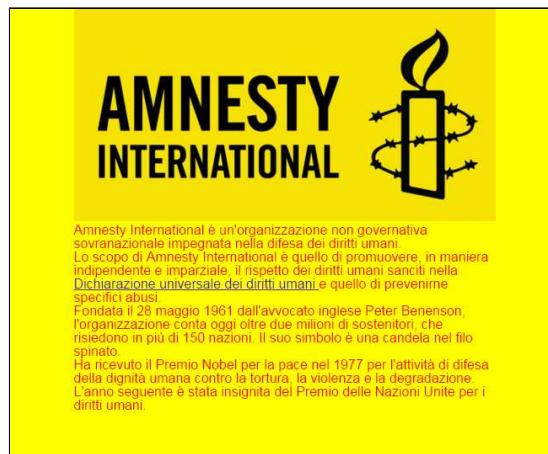
01. <!doctype html>
02. <html>
03.   <head>
04.     <title>Amnesty International</title>
05.     <meta charset="UTF-8" />
06.     <style>
07.       body {
08.         font-family:sans-serif;
09.         color:red;
10.         width:500px;
11.         background-color: yellow;
12.         margin:auto;
13.       }
14.       img {
15.         width:500px;
16.       }
17.     </style>
18.   </head>
19.   <body>
20.     <br/>
21.     
22.     <p>
23.       ...

```

Per prima cosa togliamo l'elemento **style** da **head** e spostiamo quelle due regole css in un file esterno che chiameremo **stili.css**; linkiamo quindi il file **reset.css** e questo nuovo file.

Come al solito entrambi i file saranno nella cartella **css**.

L'ultima modifica che facciamo all'html è l'eliminazione dell'elemento **
** all'inizio del documento: adesso sappiamo come distanziare gli oggetti tra loro.



Box model e Javascript /2

07

Come si può vedere c'è un po' di lavoro da fare.

Cominciamo con una cosa facile facile: il colore.

Nella lezione 6 ti è stato mostrato lo strumento **colorpix**: è il suo momento. Scaricalo, lancialo e scopri la codifica esadecimale del giallo che funge da sfondo all'immagine.
Imposta questo colore come sfondo alla finestra.



Come abbiamo visto in passato: è una buona regola posizionare dentro un elemento contenitore tutto il contenuto del **body** per poter gestire separatamente il **body** e il suo contenuto.

Aggiungi quindi un elemento **article** che contiene tutto quanto ora è annidato in body e separa la regola del body come segue.

stili.css

```
01. body {
02.     font-family:sans-serif;
03.     color:red;
04.     background-color: #F6E303;
05. }
06. article{
07.     width:500px;
08.     margin:auto;
09. }
```

Adesso occupiamoci degli spazi che abbiamo perso a causa del reset.

Cominciamo con il distanziare l'immagine sia dal testo che dal margine superiore della finestra.

stili.css

```
10. img {
11.     width:500px;
12.     margin-top: 10px;
13.     margin-bottom: 10px;
14. }
```



Box model e Javascript /3

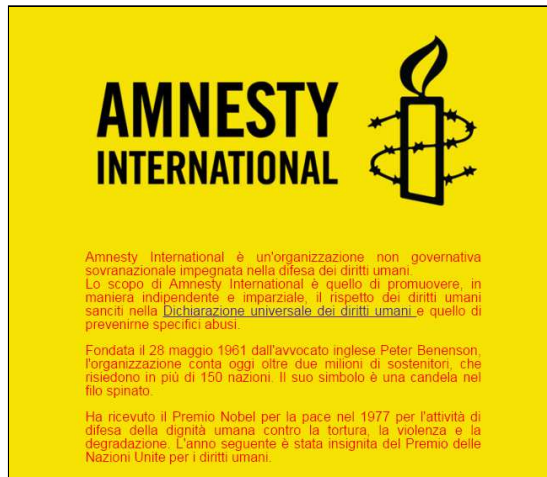
07

Occupiamoci infine del **paragrafo**.
Distanziamo i paragrafi tra loro e li mostriamo giustificati.

stili.css

```
15. p {
16.     margin: 15px;
17.     text-align: justify;
18. }
```

Dovresti avere ottenuto qualcosa di simile a questo.



Occupiamoci adesso della parte dinamica.

Per ottenere quel che hai visto nella lezione numero sette bisognerà aggiungere un piccolo script all'inizio della nostra pagina html.

index.html

```
09.     <body>
10.         <script>
11.             var nome = prompt("Come ti chiami?");
12.         </script>
```

In questo modo ci assicuriamo che la prima cosa che si fa quando si renderizza questa pagina web è eseguire questo piccolo script che mette a video un prompt con la domanda Come ti chiami? e che conserva la risposta in una variabile chiamata **nome**.

Subito dopo andiamo a definire un elemento **footer** come segue.

index.html

```
13.         <footer>
14.             Sessione di <script> document.write(nome); </script>
15.         </footer>
```

Il contenuto di questo **footer** è il testo "sessione di" seguito dal contenuto della variabile **nome**.

Hai dato un'occhiata alla pagina web prodotta?

L'elemento **footer** se non viene chiaramente "agganciato" al fondo della pagina resta un elemento come tutti gli altri.



Box model e Javascript /4

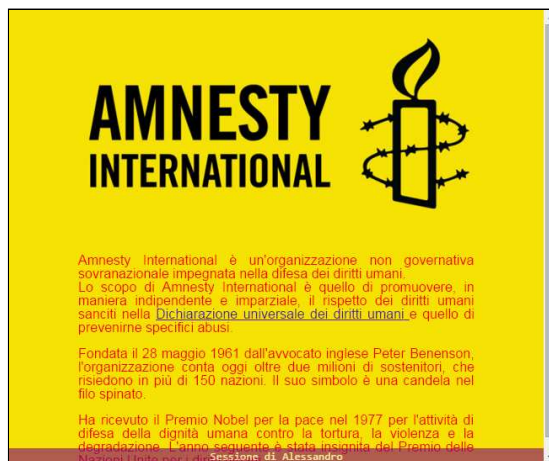
07

Guarda la differenza tra prima e dopo l'aver aggiunto questa regola css.

stili.css

```
19. footer{
20.     position: fixed;
21.     bottom: 0px;
22. }
```

Ovviamente siamo ancora lontani dall'aspetto che abbiamo visto nelle slide.



Cominciamo a dare una dimensione al box.

stili.css

```
23. width: 100%;
24. height: 20px;
25. line-height: 20px;
```

Vogliamo occupare il 100% dello spazio a disposizione (del **body** e non dell'**article**); vogliamo un'altezza di 20 pixel e vogliamo il testo centrato verticalmente (questo si ottiene impostando la proprietà **line-height** allo stesso valore di **height**).

Adesso occupiamoci dei colori

stili.css

```
28. background: purple;
29. color: white;
30. opacity: 0.6;
```

Anche stavolta c'è una novità: la proprietà **opacity** imposta la trasparenza dell'oggetto.

Infine, andiamo banalmente ad impostare dimensione, tipo e allineamento del testo.

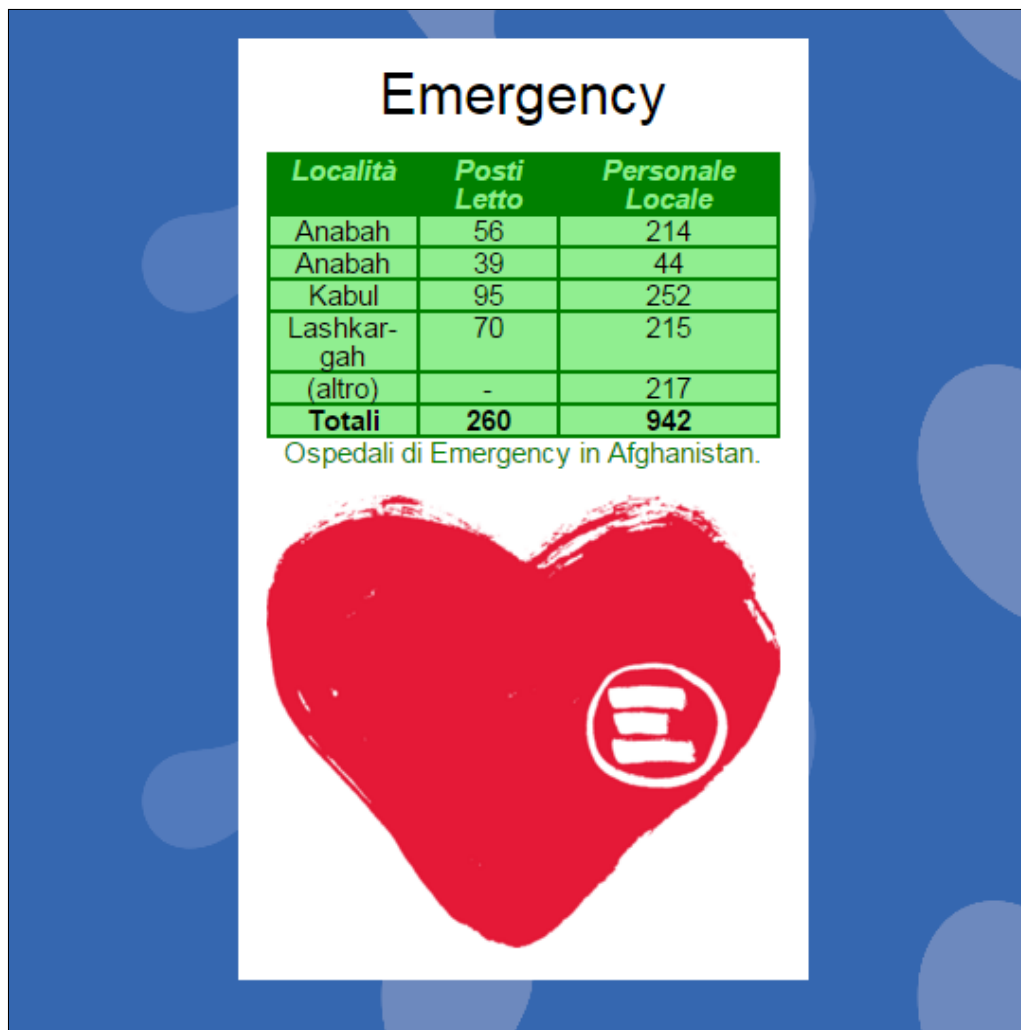
stili.css

```
32. font-size: 1em;
33. font-family: monospace;
34. text-align: center;
```

Modifica come credi il file **dichiarazione.html**



2) Rivedi il progetto di Emergency



Fatta la copia della cartella **22-emergency** in **28-emergency** edita il file **index.html** ed effettua le seguenti modifiche.

- fai diventare l'elemento **div** un elemento **article** (sia nell'html che nel css)
- togli tutti i **
** dall'html messi lì per fare spazio.

Successivamente edita il file **stili.css** ed apporta le seguenti modifiche.

- distanzia l'**article** dal lato superiore della finestra impostando la sua proprietà **margin-top** a 15px
- fai in modo che il suo contenuto non sia a ridosso del suo contorno impostando un **padding** di 15px
- reimposta opportunamente la proprietà **margin** della tabella:
 - gli elementi verticali devono essere distanziati di 15px
 - orizzontalmente deve restare la tabella deve restare al centro

Infine fai in modo che appaia un popup con il seguente messaggio: "Fai una donazione di due euro al 45678!"

Box model e Javascript /6

07

3) Nuovo progetto

Vogliamo realizzare un piccolo giochino.

Appena il progetto parte compare un popup con la scritta "Giochiamo!".

Successivamente viene renderizzato il sito che si presenta come in figura.



Quando l'utente passa su una delle immagini lo sfondo di questa immagine diventa rosso.

Quando l'utente clicca su una delle immagini viene caricata un'altra pagina web con l'esito della sua scelta.

Se l'utente clicca sull'immagine raffigurante l'ebrea ha vinto e gli viene presentata una pagina simile a quella mostrata qui di seguito; se clicca sulle altre due, perde e gli viene mostrata una delle pagine presentate successivamente.



Resta evidente che il progetto consta di quattro pagine web, di tre immagini e dei soliti due file css.

Realizza la cartella **29-suora** e al suo interno crea le cartelle **img** e **css**.

Preleva le immagini necessarie all'indirizzo www.bbuio.it/varie/29-suora.zip e copiale nella cartella **img**.

Inserisci nella cartella **css** il classico **reset.css** ed il nostro file **stili.css** (per il momento vuoto).

Procedi con la creazione del file **index.html** e realizza la sezione **head** come al solito.



Box model e Javascript /7

07

Realizza quindi il corpo della pagina come di seguito mostrato.

index.html

```
11. <h1>Indovinello.</h1>
12. <h2>Chi studia il Pentateuco?</h2>
13. <a href="musulmana.html">
14.     
15. </a>
16. <a href="ebrea.html">
17.     
18. </a>
19. <a href="suora.html">
20.     
21. </a>
```

Per ottenere il riquadro come quello visto in apertura sarà necessario porre tutto il contenuto del **body** in un **article** e cominciare ad impostare qualche regola css.

stili.css

```
01. body {
02.     background: lightgrey;
03.     font-size: 10px;
04. }
05. article {
06.     width: 400px;
07.     margin: 20px auto;
08.     padding: 20px;
09.     background: white;
10. }
```

Siamo a questo punto.



Box model e Javascript /8

07

Occupiamoci adesso dei testi.

Modifichiamo come segue la regola per **article** e aggiungiamo le regole per **h1** e **h2**.

stili.css

```

05. article {
06.     width: 400px;
07.     margin: 20px auto;
08.     padding: 20px;
09.     background: white;
10.     text-align: center;
11.     color: gray;
12.     font-family: Verdana;
13. }
14. h1 {
15.     font-size: 5em;
16. }
17. h2 {
18.     font-size: 3em;
19. }

```

Osserviamo che abbiamo sistemato i testi ed occupiamoci delle immagini.



Le immagini vanno rese un po' più piccole e soprattutto vanno rese dinamiche: vogliamo che abbiano lo sfondo in rosso quando si trovano nello stato **hover**.

Aggiungiamo le seguenti regole a **stili.css**.

stili.css

```

20. img{
21.     width: 100px;
22. }
23. img:hover{
24.     background:red;
25. }

```



Box model e Javascript /9

07

Otteniamo finalmente qualcosa di carino.



Notiamo infine quei minuscoli trattini tra un'immagine e l'altra e - soprattutto - notiamo che nel passaggio del mouse tra una immagine e l'altra esiste un piccolo spazio vuoto.

Questo spazio vuoto coincide con quel trattino ed altro non è che uno spazio (di testo) che ci viene messo in automatico.

Per aggirare il problema, infiliamo le tre immagini in una **section** e poniamo una regola che imposti una dimensione del testo pari a zero in questa **section**.

index.html

```
13. <section>
14.   <a href="musulmana.html">
15.     
16.   </a>
17.   <a href="ebrea.html">
18.     
19.   </a>
20.   <a href="suora.html">
21.     
22.   </a>
23. </section>
```

stili.css

```
26. section {
27.   font-size: 0px;
28.   margin-top: 15px;
29. }
```

Hai già dato un'occhiata al risultato?

Perfetto!



Box model e Javascript /10

07

Passiamo alla realizzazione della pagina successiva: **musulmana.html**



In realtà la parte superiore di questa pagina appare uguale a quella appena realizzata. Mentre la parte seguente riporta l'immagine della donna musulmana molto più grande e con uno sfondo grigio.

Ovviamente gli elementi **img** saranno due ma l'immagine sarà la stessa: dobbiamo distinguere quindi questi due tipi di immagine: la musulmana piccola e la musulmana grande.

Inoltre, in questa pagina, le immagini piccole non devono essere cliccabili e non devono reagire al passaggio del mouse.

Quindi le immagini piccole del file **index** devono essere di un tipo e le immagini piccole degli altri file devono essere di un altro tipo.

Poichè il file **stili.css** vogliamo sia lo stesso per tutte le pagine abbiamo la necessità di distinguere tutti i tipi di immagine di tutto il sito.

- le immagini piccole, cliccabili e sensibili al passaggio del mouse saranno di tipo **opzione**
- le immagini piccole, non cliccabili e non sensibili al passaggio del mouse saranno di tipo **noOpzione**
- le immagini grandi, con sfondo grigio saranno di tipo **sceltaSbagliata**
- le immagini grandi, con sfondo rosso saranno di tipo **sceltaCorretta**



Box model e Javascript /11

07

Eliminiamo quindi queste due regole dal file **stili.css**.

stili.css

```
20. img{
21.     width: 100px;
22. }
23. img:hover{
24.     background:red;
25. }
```

Modifichiamo il file **index.html** in modo tale che tutte le sue immagini diventino di **classe** opzione (si mostra - come esempio - la prima immagine del file).

index.html

```
14. <a href="musulmana.html">
15.     
16. </a>
```

E si aggiungono le seguenti regole al file css.

stili.css

```
24. img.opzione, img.noOpzione {
25.     width: 100px;
26. }
27. img.opzione:hover{
28.     background:red;
29. }
30. img.sceltaSbagliata {
31.     background:gray;
32.     width: 300px;
33. }
34. img.sceltaCorretta {
35.     background:red;
36.     width: 300px;
37. }
```

A questo punto andiamo a realizzare il file **musulmana.html**.



Box model e Javascript /12

07

Il **body** di questo file comincia con un codice molto simile all'**article** di **index.html**; le uniche differenze sono l'assenza dei tag **a** e la **classe** delle immagini che non sarà più **opzione** ma sarà **noOpzione**.

musulmana.html

```

10. <article>
11.     <h1>Indovinello.</h1>
12.     <h2>Chi studia il Pentateuco?</h2>
13.     <section>
14.         
15.         
16.         
17.     </section>
18.     
19.     <p>Sbagliato!</p>
20. </article>

```

Ovviamente prima di chiudere l'**article** abbiamo aggiunto l'immagine della musulmana assegnandole la classe **sceltaSbagliata** e un testo con la scritta "Sbagliato".



Vediamo dall'immagine che abbiamo ulteriori due modifiche da fare.

La prima per distanziare l'immagine grande dalle piccole e la seconda per dimensionare in modo opportuno il paragrafo in fondo.

In entrambi i casi si tratta di apportare due piccole modifiche al file **stili.css**.



Box model e Javascript /13

07

stili.css

```
17. h2,p {  
18.     font-size: 3em;  
19. }
```

Qui abbiamo modificato il **selettore** aggiungendo l'elemento **p** all'elemento **h2**.

stili.css

```
20. section {  
21.     font-size: 0px;  
22.     margin: 15px 0px;  
23. }
```

E qui abbiamo rinunciato ad impostare solo la proprietà **margin-top** per stabilire con precisione il margine sui 4 lati che sarà di 15 pixel per il sopra e il sotto e nullo per i lati.

Ottenendo (finalmente) quanto segue.



Box model e Javascript /14

07

Giacchè il file **suora.html** segue le stesse osservazioni appena fatte per **musulmana.html**, te lo lascio fare in autonomia.



Finito **suora.html** possiamo cominciare a pensare a **ebrea.html**.

In realtà anche questo dovrebbe essere ampiamente alla tua portata: tutto ciò che devi fare è assegnare un'altra classe all'ultima immagine, ottenendo quindi questo.



Ma se osservi bene questa pagina web con quella mostrata in apertura di esercizio ti accorgi che il colore della scritta "Giusto" è diverso.



Box model e Javascript /15

07

Realizza una nuova classe **giusto** ed assegna il paragrafo di questo file a questa classe. L'unica dichiarazione che ci sarà l'impostazione del colore a rosso.



Sebbene tu sia esausto manca un rigo di **Javascript** al tuo progetto.

Cos'è che doveva apparire appena avviato il progetto?



