

# CSS

Linguaggio utilizzato per  
formattare le pagine web!

**style.css**

[www.insegnarecoding.com](http://www.insegnarecoding.com)

[www.codingcreativo.it](http://www.codingcreativo.it)

Non è un linguaggio di programmazione,  
ma di **formattazione!**



## Cascading Style Sheets

La funzionalità è quella di separare il contenuto  
delle pagine HTML dalla sua formattazione.

Utilizza delle regole per stabilire la  
formattazione di testo, sfondi, paragrafi, ecc.

# index.html

# style.css

Come si collegano? 

Nella pagina **html** si inserisce

```
<head>  
  <title> ... </title>  
  <link rel="stylesheet" href="css/style.css">  
</head>
```

# Proprietà dei CSS

**color**: definisce il colore del testo.

**background-color**: definisce il colore di sfondo di un elemento.

**font-family**: definisce il tipo di carattere del testo.

**font-size**: definisce la dimensione del carattere del testo.

**font-weight**: definisce lo spessore del carattere del testo.

**text-align**: definisce l'allineamento del testo.

# Esempi

```
selettore {  
  proprietà: valore;  
  ...  
}
```



```
body {  
  background-color: #00ff00;  
  font-size: 18px;  
}  
  
h1 {  
  color: grey;  
  font-size: 22px;  
}
```

# Esempi

```
selettore {  
  proprietà: valore;  
  ...  
}
```

*selettore*: classe



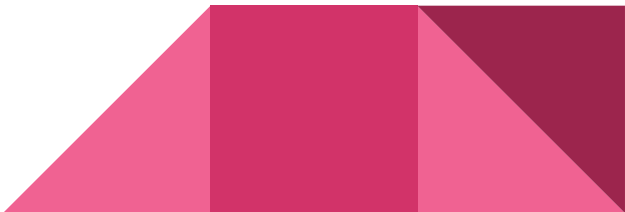
```
.testo-blue {  
  color: #0000ff;  
  font-size: 18px;  
}
```

# Esempi

```
selettore {  
  proprietà: valore;  
  ...  
}
```



```
#section1 {  
  font-size: 18px;  
}
```



# collegamenti ipertestuali

Alcuni possibili stati di **a**:

- **":link"** per definire lo stile dei collegamenti ipertestuali non ancora visitati.
- **":visited"** per definire lo stile dei collegamenti ipertestuali già visitati.
- **":hover"** per definire lo stile dei collegamenti ipertestuali quando l'utente ci passa sopra con il cursore.
- **":active"** per definire lo stile dei collegamenti ipertestuali quando l'utente ci clicca sopra.



# collegamenti ipertestuali

Per formattare un collegamento ipertestuale si può usare il tag **a**, oppure una classe o un id.

Alcune regole:

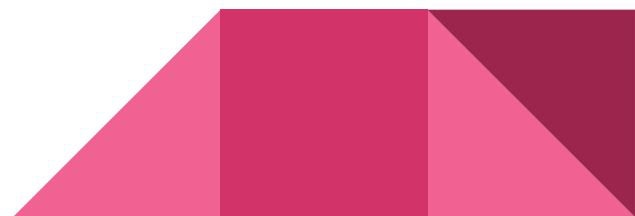
- **text-decoration** per rimuovere l'effetto sottolineato predefinito dei collegamenti ipertestuali.
- **color**: per definire il colore del testo
- **font-size**: per definire la dimensione del testo
- ....

# width ed height

**width:** definisce la larghezza di un elemento block

**height:** definisce l'altezza di un elemento block

**Provate a dare width ed height ad un elemento inline**



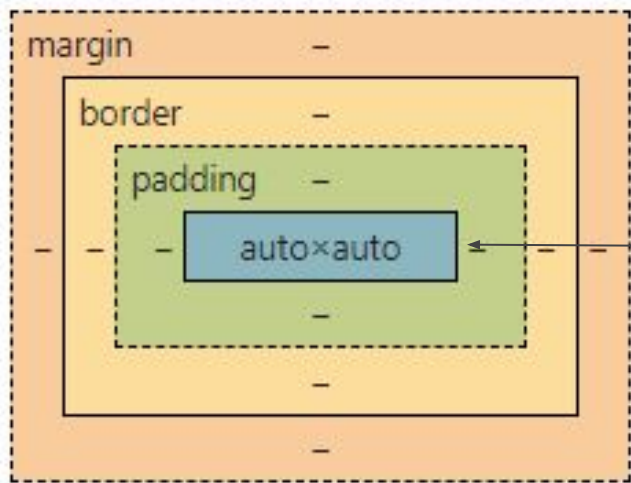
# Altre proprietà CSS

**padding**: definisce lo spazio interno dell'elemento, tra il bordo e il contenuto.

**margin**: definisce lo spazio esterno dell'elemento, tra il bordo e gli elementi adiacenti.

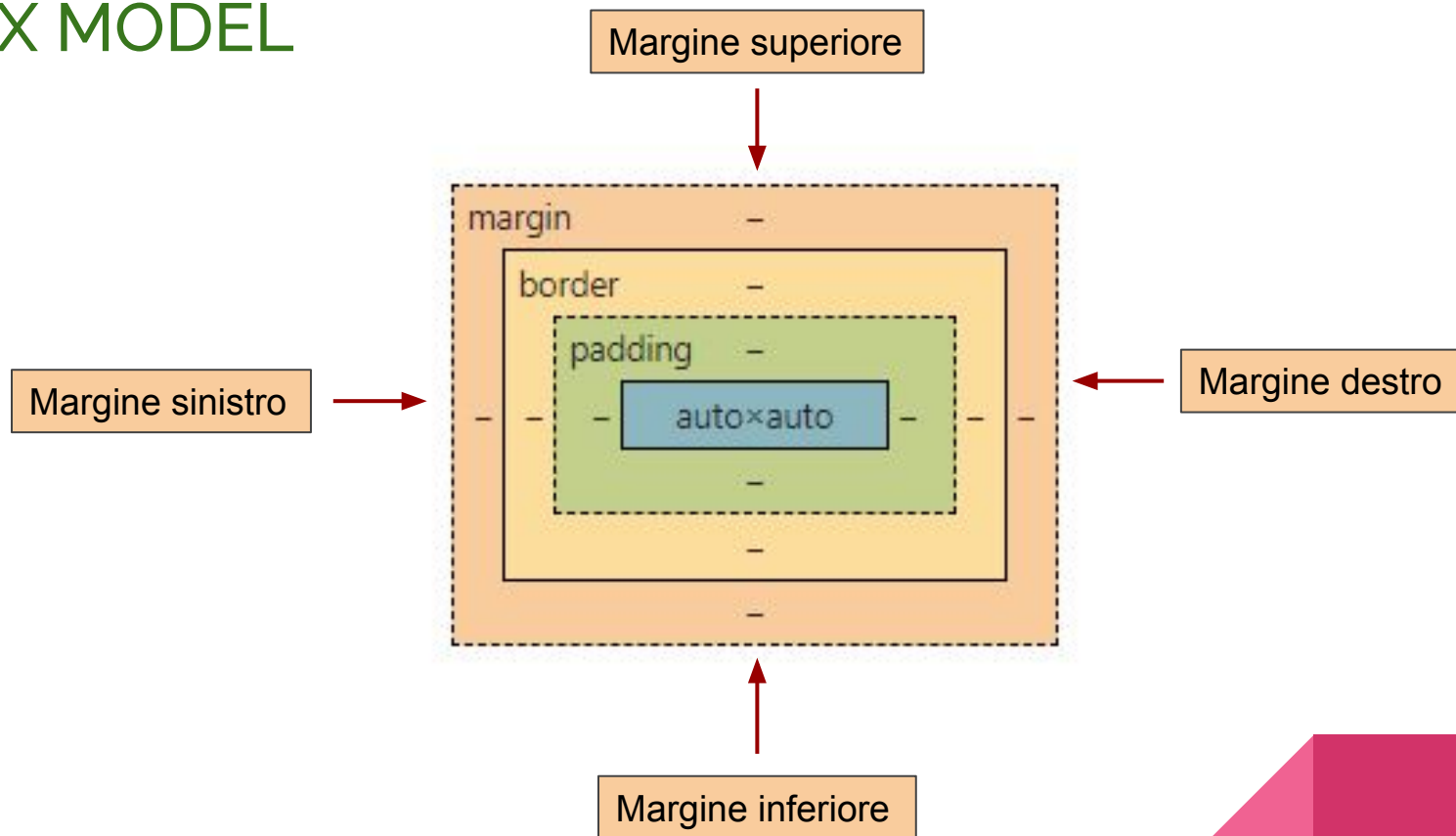
**border**: definisce lo stile, la larghezza e il colore del bordo dell'elemento.

# BOX MODEL

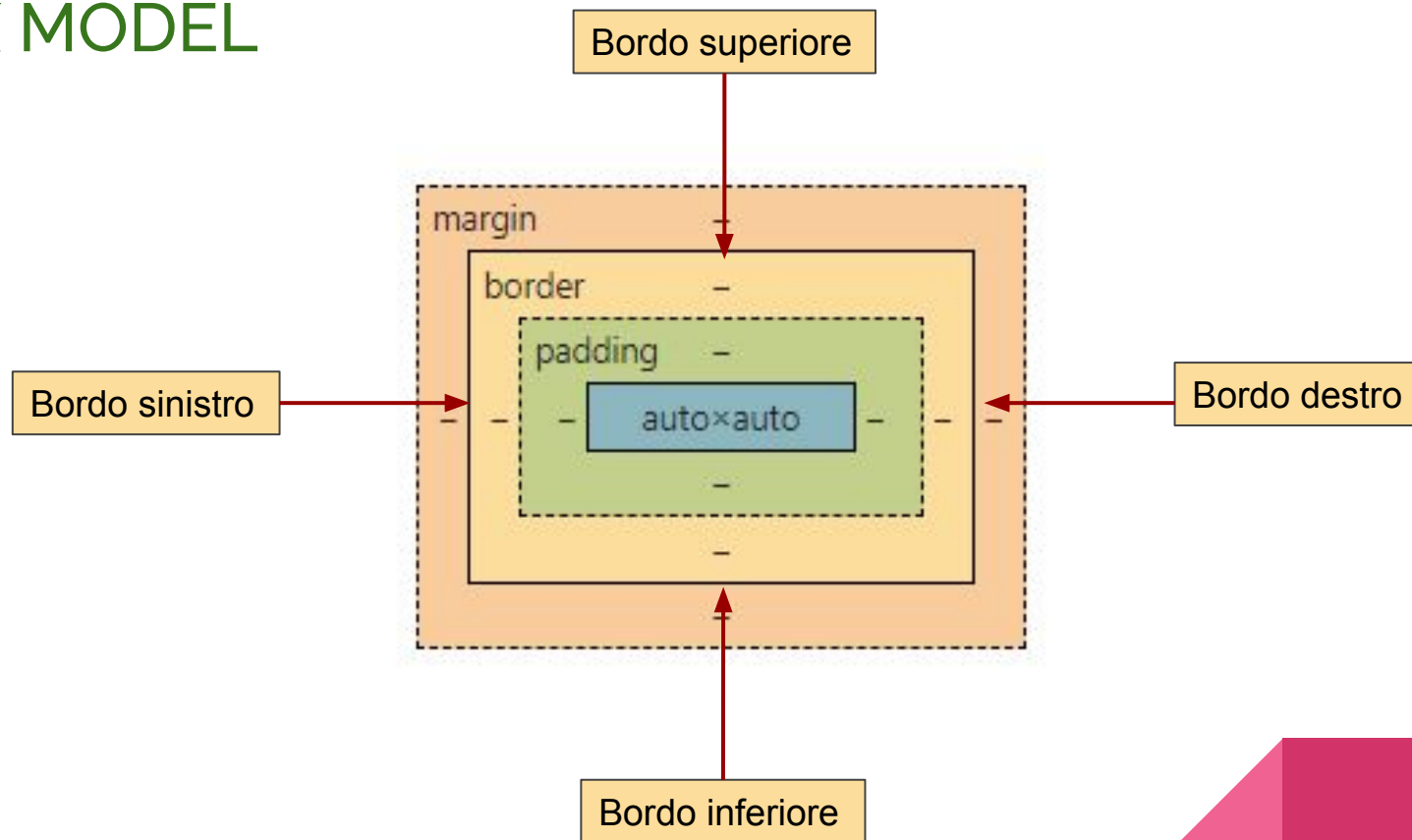


Rappresenta il contenuto vero e proprio

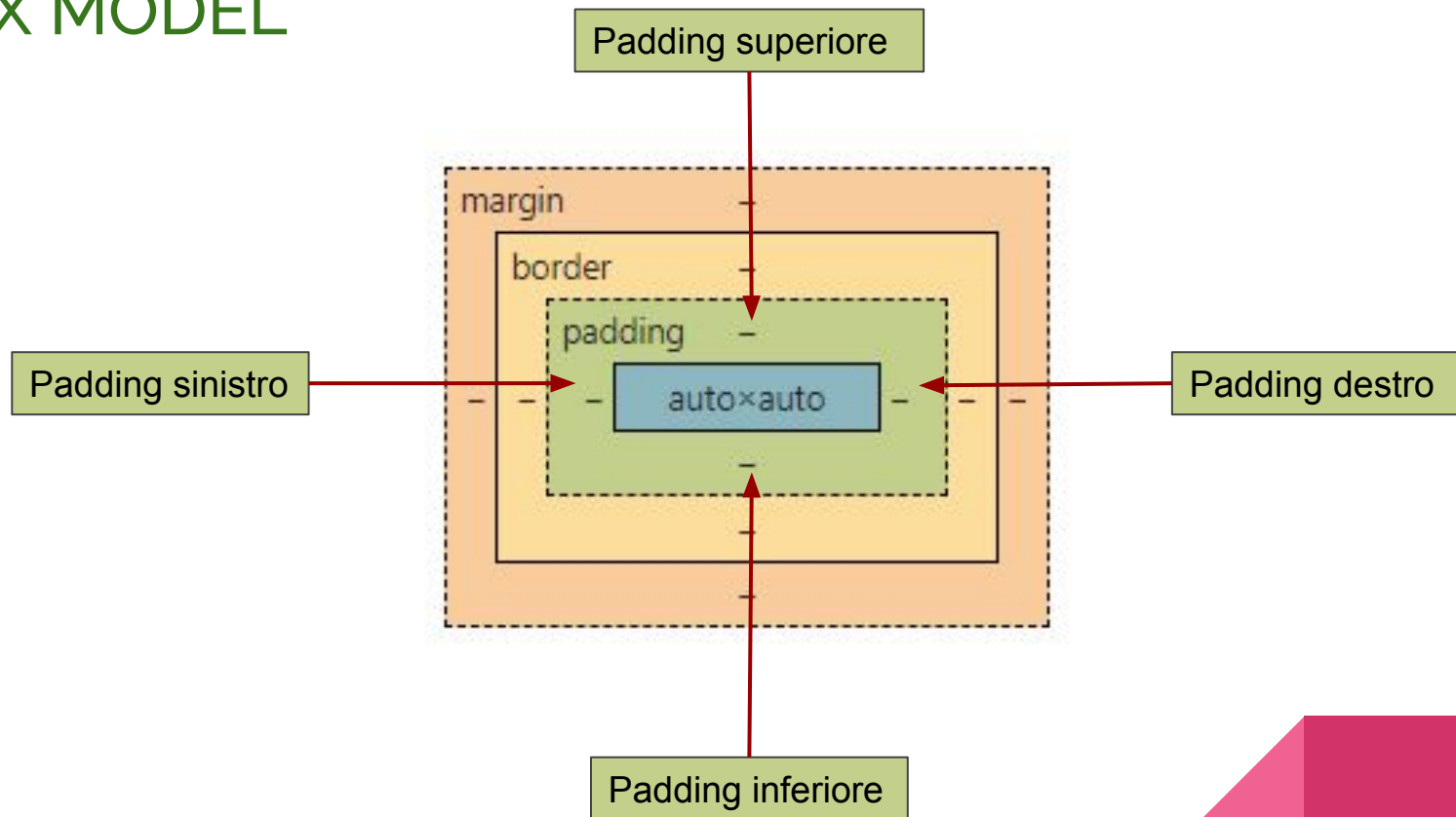
# BOX MODEL



# BOX MODEL



# BOX MODEL



# opacity

```
img {  
  opacity: 0.5;  
}
```

L'opacità specifica la trasparenza di un elemento!

```
img:hover {  
  opacity: 1.0;  
}
```





Esempio d'uso degli **id**:

[www.insegnarecoding.com](http://www.insegnarecoding.com)  
[www.codingcreativo.it](http://www.codingcreativo.it)

```
#box1 {  
  width: 200px;  
  height: 200px;  
  background-color: beige;  
  padding: 20px;  
  margin: 20px;  
}
```

```
#box2 {  
  width: 200px;  
  height: 200px;  
  background-color: aqua;  
  padding: 20px;  
  margin: 20px;  
}
```

Esempio d'uso delle **classi**:

[www.insegnarecoding.com](http://www.insegnarecoding.com)  
[www.codingcreativo.it](http://www.codingcreativo.it)

```
.box1 {  
  width: 200px;  
  height: 200px;  
  background-color: beige;  
  padding: 20px;  
  margin: 20px;  
}
```

```
.box2 {  
  width: 200px;  
  height: 200px;  
  background-color: aqua;  
  padding: 20px;  
  margin: 20px;  
}
```

# box-sizing

**Le dimensioni del padding e del bordo si sommano** alle dimensioni di width e height del blocco. Quindi il blocco occupa una quantità di spazio maggiore nella pagina web con la conseguenza che il layout non sarà come ci aspettiamo.

Tuttavia, è possibile specificare il **box-sizing** come "**border-box**" per far sì che le dimensioni del padding (e del bordo) siano incluse nelle dimensioni totali del blocco, evitando così l'aumento delle dimensioni. Questo è particolarmente utile quando si vuole mantenere un layout preciso e coerente, senza dover continuamente calcolare le dimensioni aggiuntive del padding.

# regole di reset

```
*{  
  
margin: 0;  
  
padding: 0;  
  
box-sizing: border-box;  
  
}
```

# display

La proprietà CSS display definisce il modo in cui un elemento HTML deve essere visualizzato sullo schermo. Esistono diversi valori per questa proprietà, tra cui:

- **block**: l'elemento viene visualizzato come un blocco e occupa tutta la larghezza disponibile. Puoi impostare l'altezza e la larghezza di questi elementi e utilizzare il margine e il padding.
- **inline**: l'elemento viene visualizzato "in linea" con il testo che lo circonda. La larghezza e l'altezza non sono modificabili e il margine e il padding non funzionano allo stesso modo come per gli elementi a blocco.

# display

- **inline-block**: L'elemento viene visualizzato come inline, ma è possibile impostare l'altezza e la larghezza dell'elemento, così come la distanza da altri elementi tramite margin e padding.
- **none**: l'elemento non viene visualizzato sulla pagina. Questo può essere utile per nascondere temporaneamente un elemento o una sezione della pagina.

Ci sono anche altri valori che possono essere utilizzati insieme alla proprietà display, come table, flex e grid. Questi valori ti consentono di creare layout più complessi e di posizionare gli elementi sulla pagina in modo più preciso.

# ereditarietà

Per ereditarietà nei CSS si intende un modo in cui un elemento figlio eredita le proprietà dai suoi elementi genitori.

Questo significa che se un elemento padre ha una proprietà come il colore del testo impostato su verde, l'elemento figlio erediterà automaticamente il colore del testo verde a meno che non venga specificato un colore diverso per l'elemento figlio.

# ereditarietà

L'ereditarietà può essere applicata a molte proprietà CSS, tra cui:

- Sfondo
- Colore del testo
- Famiglia di caratteri, stile e dimensione del testo
- Margini e padding
- width e height
- Opacità e trasparenza





# specificità

Una regola definita in un foglio di **stile inline** ha la precedenza sulla regola definita in un foglio di stile esterno.

Una regola CSS che seleziona un elemento tramite un **ID** ha una maggiore specificità rispetto a una regola che seleziona lo stesso elemento tramite una **classe** o un **tag**.

In caso di parità di specificità, la regola che appare dopo nel foglio di stile ha la precedenza, cioè sovrascrive la precedente.

# position

La proprietà **position** nei CSS può assumere i seguenti possibili valori:

**static** (valore di default): l'elemento viene posizionato nella normale "flow" del documento, seguendo l'ordine in cui è scritto nel codice HTML;

**relative**: l'elemento viene posizionato rispetto alla sua posizione normale nel "flow" del documento, utilizzando le proprietà top, right, bottom e left;

**absolute**: l'elemento viene posizionato in un punto preciso della pagina, utilizzando le proprietà top, right, bottom e left, e viene estratto dal "flow" del documento;

# position

La proprietà **position** nei CSS può assumere i seguenti possibili valori:

**fixed**: l'elemento viene posizionato in un punto preciso della pagina, utilizzando le proprietà top, right, bottom e left, e rimane in posizione fissa anche durante lo scrolling della pagina;

**sticky**: l'elemento viene posizionato in modo relativo fino a quando non raggiunge una determinata posizione rispetto al viewport, a quel punto diventa posizionato in modo "fixed", ma solo finché la sua area di contenimento è visibile.