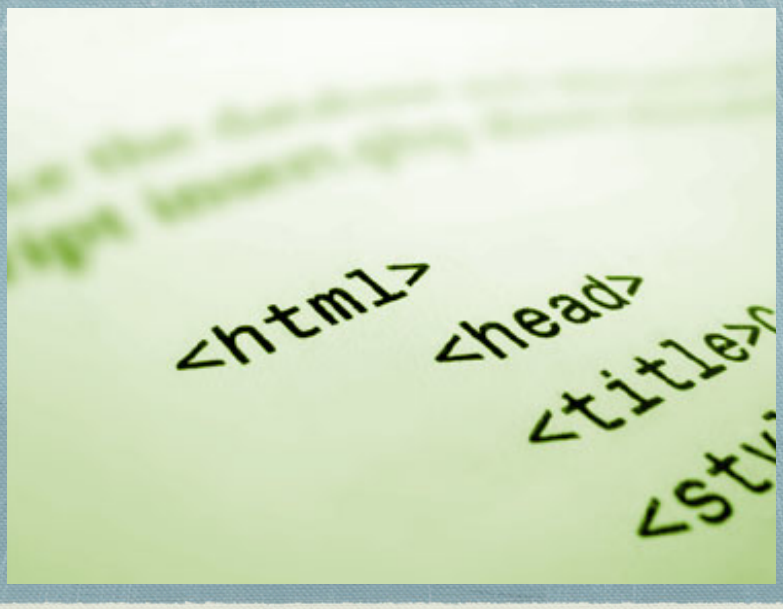
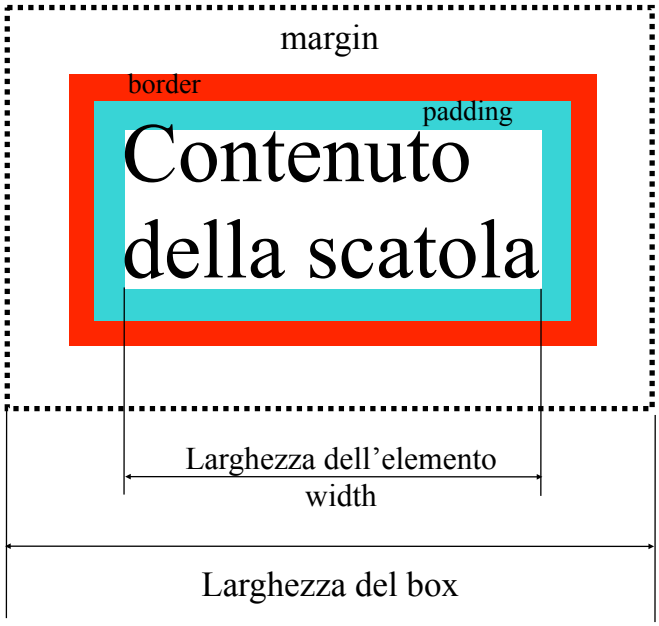


CSS - parte 2

Corso di Web Design
Fabio Pittarello, Università Ca' Foscari Venezia - DAIS pitt@unive.it

Proprietà dei contenitori (box)



Proprietà dei contenitori (box)

- I fogli di stile considerano ogni elemento della pagina come se fosse contenuto all'interno di una scatola.
- Il contenuto vero e proprio (zona bianca interna con testo, vedi figura) ha una determinata larghezza (**width**), è circondato da un margine interno (**padding**, segnalato in celeste nella figura), da un bordo (**border**, segnalato in rosso nella figura) e da un margine esterno (**margin**, trasparente).
- Qualsiasi sfondo applicato ad un elemento si estende nello spazio del padding e sotto il bordo.
- CSS consente di controllare le caratteristiche del box, incluse la grandezza del padding e dei margini, l'aspetto dei bordi, il colore e le immagini di sfondo.

Proprietà CSS dei contenitori (box)

- Margin-top, margin- right, margin-bottom, margin-left
  - Per specificare la dimensione del margine in maniera individuale nei quattro lati
- Margin
  - E' una forma abbreviata per specificare tutte le dimensioni dei margini in una sola regola
- Padding-top, padding-right, padding-bottom, padding-left
  - Per specificare la dimensione del padding in maniera individuale nei quattro lati
- Padding
  - E' una forma abbreviata per specificare tutte le dimensioni del padding in una sola regola
- Border-top-width, border-right-width, border-bottom-width, border-left-width
  - Per specificare la dimensione del bordo in maniera individuale nei quattro lati
- Border-width
  - E' una forma abbreviata per specificare la dimensione del bordo nei quattro lati

Proprietà CSS dei contenitori (box)

- Border-color
  - Per specificare il colore del bordo anche in maniera individuale nei quattro lati
- Border-style
  - Per specificare lo stile del bordo; i valori possibili sono none|dotted| dashed|solid|double|groove|ridge|inset|outset
- Border-top, border-right, border-bottom, border-left
  - Per specificare la dimensione, il colore e lo stile del bordo in una sola regola
- Border
  - E' una forma abbreviata per specificare la dimensione, il colore e lo stile di del bordo per tutti e quattro i lati dell'elemento in una sola regola

Proprietà CSS dei contenitori (box)

- Width
  - Per stabilire la larghezza dell'elemento; si può applicare ad es. a paragrafi e immagini
- Height
  - Per stabilire l'altezza dell'elemento; si può applicare ad es. a paragrafi e immagini

Come funziona il box model

- Secondo la specifica CSS si assegnano in maniera additiva valori al contenuto, al margine interno, al bordo e al margine esterno
- Es.
  - contenuto 300 px
  - padding (margine interno) 100 px
  - bordo 5 px
  - larghezza totale del box: 5 + 100 + 300 + 100 + 5 = 510 px
- E' possibile mescolare unità di misura diverse per le singole parti di un elemento

Il box model … rotto

- Le prime implementazioni CSS (IE 4 fino a IE 5.5 Windows e IE Macintosh prima della 5.0) interpretavano in modo diverso il box model: la larghezza e l'altezza del contenuto venivano riferite al box intero, margini esterni compresi.

- Implementazioni sbagliate secondo le specifiche CSS, ma coerenti con l'idea che fosse opportuno dare le dimensioni complessive della scatola più che dare le dimensioni di quello che ci sta dentro.

Come rimediare

- Per i box senza margini e bordi
  - Il problema non si pone, perchè le dimensioni del contenuto coincidono con le dimensioni del box
- Per gli altri casi
  - Div nidificati
  - Regola diTantek
  - Escape hack
  - Commenti condizionali

Div nidificati

- L'utilizzo di due tag div nidificati risolve il problema:
  - dichiarando indirettamente la larghezza complessiva del box interno attraverso la dichiarazione della larghezza del contenuto del box esterno (sprovvisto di margini e bordi, in modo che larghezza contenuto= larghezza contenitore e quindi ottenendo una visualizzazione identica sui diversi browser);
  - per il box interno vengono definite solo le caratteristiche del bordo interno e del padding, evitando di dichiarare la larghezza del contenuto.

```
<style type="text/css">
<!--
.box { width:224px;}
.content { border:5px solid #900; padding:7px; }
-->
</style>
[...]
<div class="box">
       <div class="content">
       Lorem ipsum dolor sit amet, consectetuer
       adipiscing elit, sed diam nonummy ...
       </div>
</div>
```

Div nidificati II

Div nidificati III

- Il limite di questa soluzione è che vengono generati degli elementi di markup non strutturali, il cui unico scopo è risolvere bug implementativi.
- L'uso di troppi elementi di questo tipo porta ad una sgradevole *malatttia* .. la divite (cfr. Zeldman)

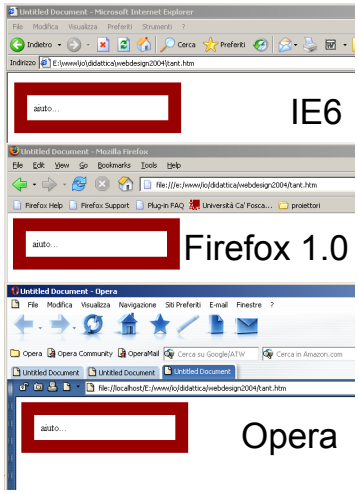
Regola di Tantek

- Utilizza un errore di interpretazione di IE 5.X Windows per fornire una falsa dimensione e sovrascriverla successivamente con la dimensione reale.

Regola di Tantek - 2

- La prima regola serve a ingannare IE 5.X Win, sfruttando un bug di questo browser nell'interpretazione delle regole voice-family; a causa di questo il secondo valore (corretto per i browser conformi, ma problematico per IE) per la larghezza viene ignorato da IE.
- La seconda regola serve a dare il giusto valore di larghezza a Opera 5, che interpreta correttamente i valori dimensionali, ma soffre dello stesso problema di IE per le regole voice-family.
- Es. generazione di un box di dimensioni complessive (margini esterni esclusi) pari a 260 px = 200px contenuto + 20 px padding + 40 px bordo.

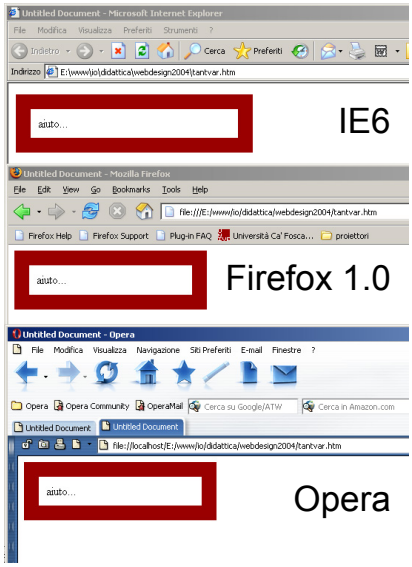
```
…
<style type="text/css">
<!-
.content {width:260px;
          border:20px solid #900;
          padding:10px;
          voice-family: "\"}\"";
          voice-family: inherit;
          width:200px;}
html>body .content { width:200px; }
-->
</style>
<body><div class="content">aiuto … </div></body >
```



Regola di Tantek - 3

- Una semplificazione ulteriore delle regole di stile, come quella proposta nel codice che segue non è applicabile, dal momento che non solo IE 5.X Windows, ma anche IE 6 Windows non è in grado di comprendere html > body e pertanto applicherebbe una larghezza scorretta pari a 260 px.

```
…
<style type="text/css">
<!-
.content {width:260px;
          border:20px solid #900;
          padding:10px;}
html>body .content { width:200px; }
-->
</style>
<body><div class="content">aiuto … </div></body >
```



Escape hack

- Hack alternativo ai precedenti, nel quale si utilizza una diversa interpretazione del carattere di escape (\) da parte di IE 5.X per Windows.
  - Per la maggior parte dei browser la presenza del carattere viene semplicemente ignorata, e questo fa sì che il valore della larghezza possa essere portato alla giusta dimensione (200 nell'esempio), sovrascrivendo il valore precedente (260 nell'esempio) che realizza una presentazione corretta solo per IE 5.X
  - Per IE 5.X la presenza di tale carattere rende non legggibile il carattere seguente (t), modificando di fatto il nome della proprietà width in widh (che non ha alcun significato); in questo modo il valore precendente (260) per IE non viene sovrascritto.

```
…
<style type="text/css">
<!-
.content {width:260px;
          border:20px solid #900;
          padding:10px;
          wid\th:200px;}
-->
</style>
<body><div class="content">aiuto … </div></body >
```

Escape hack per altre proprietà

- L'escape hack può essere usato anche per risolvere problemi legati ad altre proprietà (avendo l'avvertenza di non inserire il carattere di escape direttamente prima di una delle prime 6 lettere dell'alfabeto, che attribuiscono un significato speciale al carattere).
- Ad esempio l'escape hack può essere usato per specificare in modo diverso la grandezza del carattere attraverso le keyword, risolvendo così un'altra manchevolezza di IE 5.X per Windows.

```
…
<style type="text/css">
<!-
.content2 {font-size: x-small; /* per IE 5.X */
          f\ont-size: small; /* regola non interpretata da IE 5.X */}
-->
</style>
<body><p class="content2">Lorem ipsum … </p></body >
```

Commenti condizionali

- I commenti condizionali sono una forma particolare di commenti che vengono visti come commenti HTML veri e propri da parte di tutti i browser (e che quindi ne ignoreranno i contenuti) tranne che da IE, che - **dalla generazione 5 fino alla 9** - è in grado di leggere quello che viene dichiarato all'interno.
- Microsoft non ha previsto di utilizzare ancora i commmenti condizionali nella versione 10 del proprio browser, quando questo gira in modalità standard
  http://blogs.msdn.com/b/ie/archive/2011/07/06/html5-parsing-in-ie10.aspx
- Si possono dunque utilizzare i commenti condizionali per dichiarare sezioni specifiche di HTML leggibili solo da IE; tali sezioni possono essere nidificate sia all'interno del tag <head> che all'interno del tag <body> e comprendere l'uso di qualsiasi tag HTML. Nel caso in cui venga dichiarato all'interno del commento un tag <STYLE> è possibile perciò specificare regole di stile a solo beneficio di IE.
- La lettura dei contenuti può inoltre avvenire in maniera selettiva, specificando attraverso una sintassi specifica quali generazioni di Internet Explorer sono abilitate a leggere i contenuti

Commenti condizionali - esempi

```
<!--[if IE]>
istruzioni speciali per IE
<![endif]-->

<!--[if IE 6]>
istruzioni speciali per IE 6
<![endif]-->

<!--[if gte IE 8]>
istruzioni speciali per IE 8 e superiori
<![endif]-->
<!--[if gt IE 6]>
istruzioni speciali per IE superiore a 6
<![endif]-->

<!--[if lt IE 9]>
istruzioni speciali per IE inferiore a 9
<![endif]-->
<!--[if lte IE 7]>
istruzioni speciali per IE 7 e inferiori
<![endif]-->
```

Commenti condizionali - applicazione

```
<head>
<style type="text/css">
    .content {width:200px;}
</style>
<!--[if lt IE 6]>
        <style type="text/css">
              .content {width:260px;}
        </style>
<![endif]-->
</head>
<body>
    <div class="content">aiuto … </div>
</body >
```

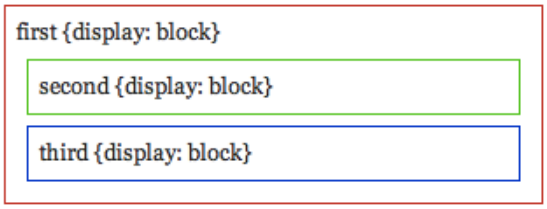
- Un'applicazione dei commenti condizionali alla definizione della larghezza di un tag .
- La dichiarazione di larghezza dell'elemento viene prima dichiarata all'interno di una sezione <style> standard all'interno del tag <head>.
- Successivamente viene dichiarato un commento condizionale all'interno del quale viene inserita un'ulteriore sezione <head> che contiene un nuovo valore per il selettore . content. Tale valore, solo per IE < 6, sovrascrive, per le priorità associate alle regole di stile, il valore precedente.

Il modello di formattazione visuale

- Ogni elemento dell'albero del documento genera zero o più box secondo il modello di box definito da CSS. La disposizione di questi box dipende da vari fattori:
  - dimensioni e tipo del box;
  - schema di posizionamento (normale, float e posizionamento assoluto);
  - relazioni tra gli elementi nell'albero del documento;
  - informazioni esterne (dimensione della finestra, dimensione delle immagini, ecc.).

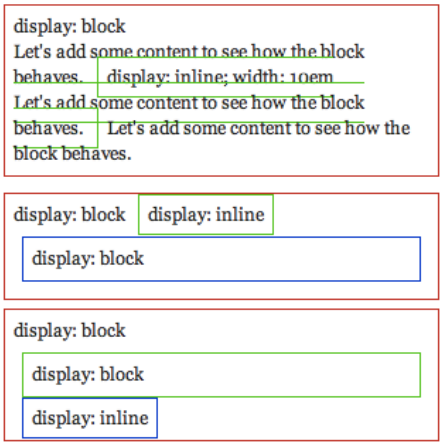
Tipi di box

- Semplificando, ai fini del posizionamento esistono due tipi fondamentali di box:
  - block box: questo tipo viene generato da elementi come p, div o table; corrisponde alla nostra idea intuitiva di box strutturato visivamente come un rettangolo
  - inline box: vengono generati da span, img o da testo non strutturato; ha una struttura visuale meno caratterizzata, può essere renderizzato su più righe
- E' possibile modificare la tipologia di default attraverso la proprietà display
  - display: block farà sì che l'elemento generi un block box, anche se di default era inline (ad es. un'immagine può essere dichiarata come block box);



Tipi di box

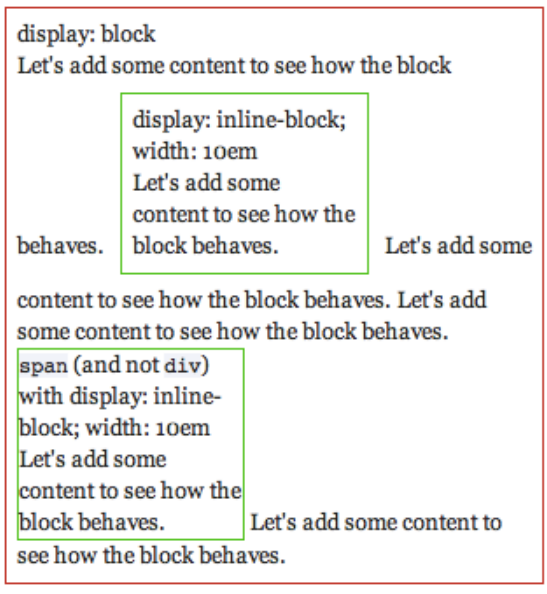
- display: inline; farà sì che l'elemento generi un inline box, anche se di default era block;



- display: none eviterà la generazione di un blocco associato all'elemento; molto utile per eliminare la presentazione di immagini larghe su piccoli schermi oppure di elementi non necessari su alcuni media come la carta stampata (ad es. un menu di navigazione può risultare inutile nel caso della stampa di una pagina web).

Tipi di box

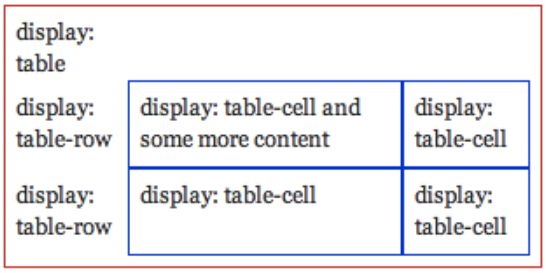
- display: inline block farà sì che l'elemento venga posizionato come un elemento inline, ma il contenuto viene formattato come un block box;



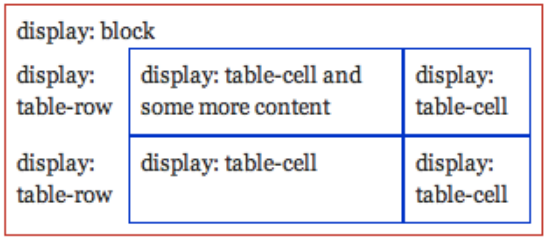
Tipi di box

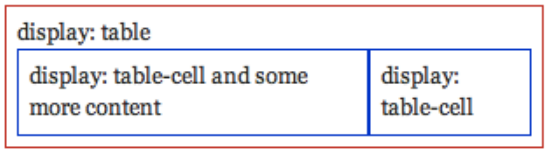
- display: list-item; farà sì che l'elemento generi un block box e un elemento di lista preceduto da un marcatore;

- display: table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, table-caption questi valori fanno sì che, nel rendering della pagina, l'elemento si comporti come una tabella o una sua componente.



Tipi di box





- Nota: tutti gli esempi sono tratti dal sito: http://www.quirksmode.org/css/display.html

Block box (box contenitori) - 2

- Il box contenitore viene utilizzato per determinare la posizione dei box contenuti e (in alcuni casi) anche per determinare la loro dimensione.
- Per elementi posizionati in maniera assoluta, si considerano come confini del blocco contenitore i limiti di contenuto del block box immediatamente superiore nella gerarchia. Se non c'è si considera la finestra del browser come blocco contenitore.

Schemi di posizionamento

- Un box può essere disposto secondo tre schemi di posizionamento fondamentali:
  - secondo il flusso normale (include la formattazione a blocco, inline, il posizionamento relativo e il posizionamento di box run-in);
  - float: un box viene prima disposto secondo il flusso normale dei documenti; poi viene estratto dal flusso e spostato il più possibile a sinistra o a destra;
  - posizionamento assoluto: un box viene rimosso dal flusso normale (senza impatto sugli elementi fratelli che seguono); ad esso viene assegnata una posizione in riferimento ad un blocco contenitore.
- Questi schemi di posizionamento vengono implementati utilizzando le proprietà descritte nelle slides successive

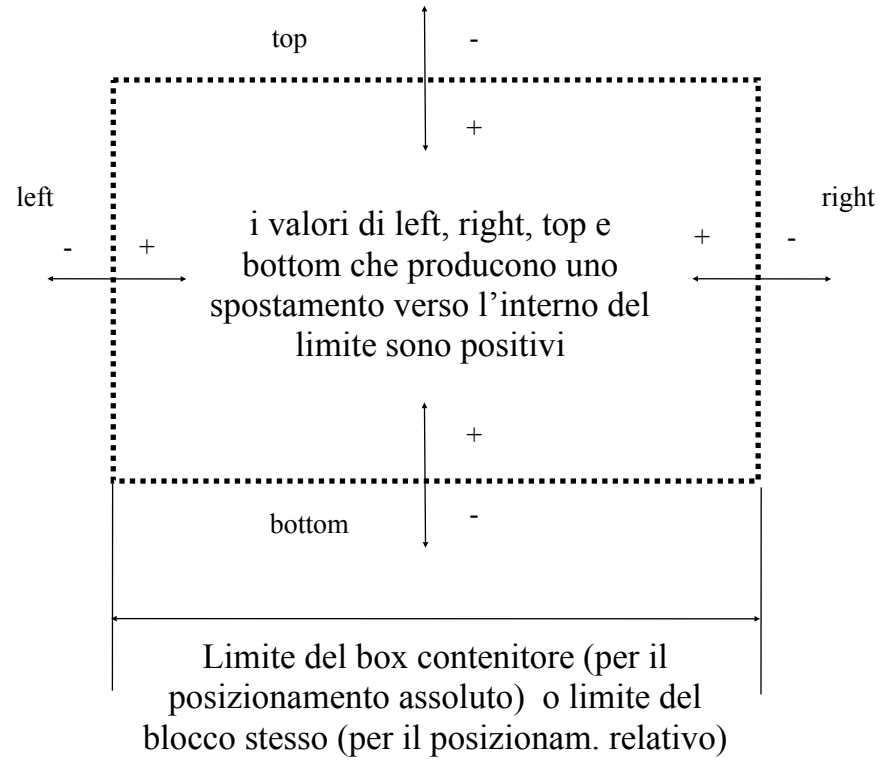
Le proprietà position e float

- Le due proprietà determinano quale algoritmo di CSS deve essere utilizzato per calcolare la posizione di un box
- position static | relative | absolute | fixed | inherit
  - static: il box viene disposto secondo il flusso normale
  - relative: la posizione viene calcolata secondo il flusso normale; poi il box viene spostato; se un box B viene posizionato in maniera relativa, la posizione dei box seguenti viene calcolata come se il box B non fosse stato spostato
  - absolute: la posizione (e anche la dimensione) vengono specificati attraverso le proprietà top, right, bottom e left; i box posizionati in maniera assoluta vengo estratti dal normale flusso; i margini dei box posizionati in maniera assoluta non collassano con altri margini
  - fixed: la posizione viene calcolata come nel modello absolute, ma viene fissata rispetto a qualche riferimento; nel caso di media types handheld, projection, screen, tty e tv il box viene fissato rispetto alla finestra di visualizzazione e non si muove quando l'utente attiva lo scrolling; nel caso del media type print il box viene stampato su ogni pagina; nel caso di altri media types la presentazione non è definita; è possibile attraverso la regola @media specificare valori diversi di posizionamento (ad es. static per print e fixed per screen)
  - inherit: stesso valore dell'elemento genitore
- float right | left | none | inherit

Box offset: le proprietà top, right, bottom e left

- Un elemento viene definito come posizionato se il valore della proprietà position è diverso da static. I box posizionati vengono disposti secondo le quattro proprietà:
  - top <length> | <percentage> | auto | inherit
    - **per positionamenti relativi** l'offset è calcolato rispetto al limite superiore esterno del box stesso nel flusso normale
    - **per positionamenti assoluti** nei quali il blocco contenitore è basato su elemento block-level si considera un offset dal limite superiore esterno del padding dell'elemento contenitore
  - right <length> | <percentage> | auto | inherit
    - come top, ma rispetto al limite destro
  - bottom <length> | <percentage> | auto | inherit
    - come top, ma rispetto al limite inferiore
  - left <length> | <percentage> | auto | inherit
    - come top, ma rispetto al limite sinistro
  - Se viene dichiarato un certo valore positivo per la proprietà top (es. top : 50px), il box posizionato viene spostato verso il basso; per bottom per valori positivi lo spostamento avviene verso l'alto, per right verso sinistra e infine per left verso destra.

Proprietà dei contenitori (box)

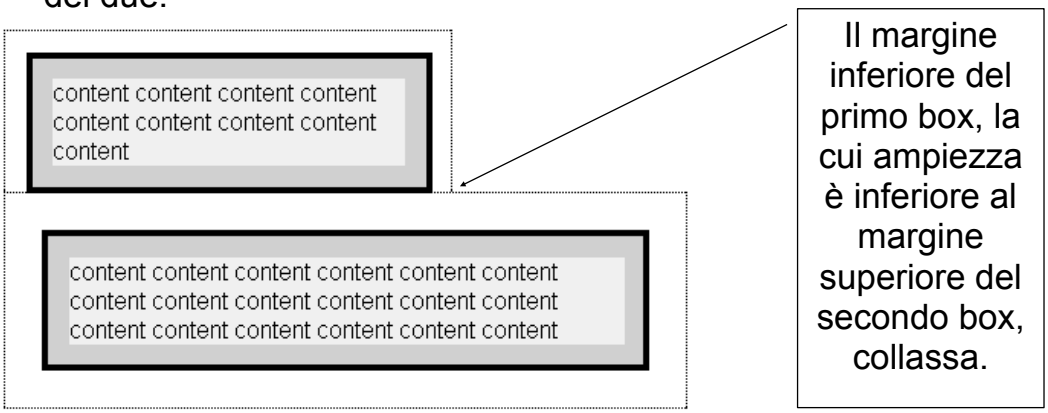


Box offset

- I valori delle quattro proprietà hanno il seguente significato:
  - <length>: distanza fissa dal limite di riferimento; sono permessi valori negativi
  - <percentage>: percentuale della larghezza (per left e right) o dell'altezza (top e bottom) del box contenitore; sono permessi valori negativi
  - auto: dipende (vedi specifica CSS per approfondimenti)
  - inherit: dipende dall'elemento genitore

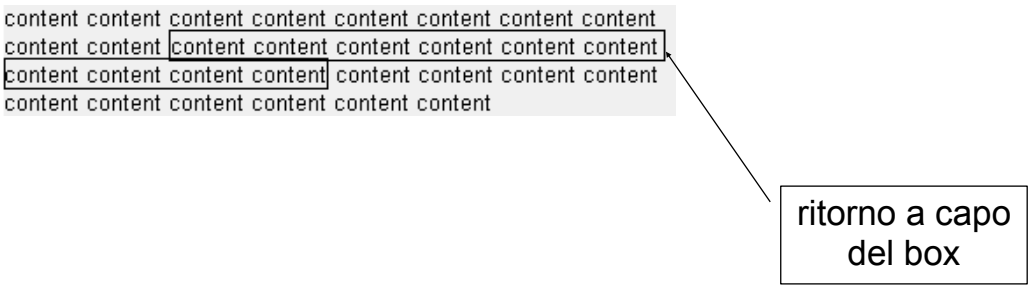
Flusso normale

- Schema di default applicato per il posizionamento. Si applica a qualsiasi elemento non floated per cui sia specificato position: static oppure relative
- I block box scorrono verticalmente a partire dalla sommità (top) del blocco contenitore; ognuno viene disposto sotto al precedente
- I margini verticali di box consecutivi vengono collassati, cioè invece di addizionare il margine inferiore di un box al margine superiore del box precedente, viene considerato solo il maggiore dei due.



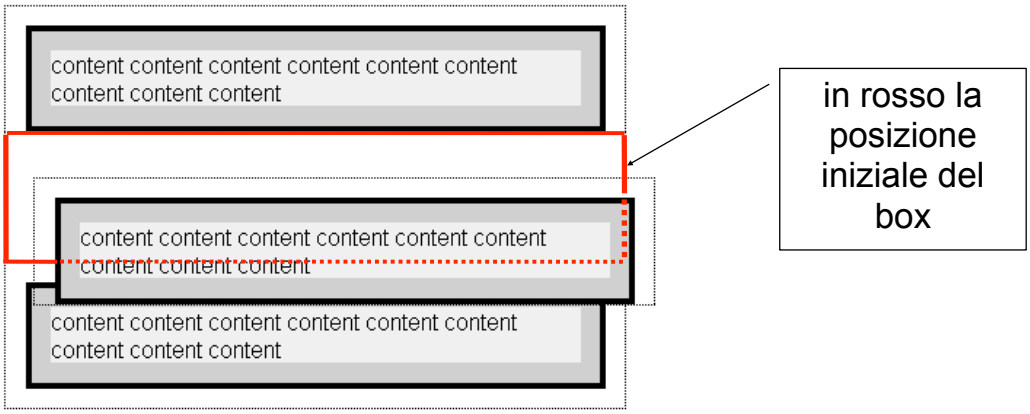
Flusso normale - 2

- Gli inline box scorrono orizzontalmente da sinistra verso destra.
- per gli inline box è consentito il ritorno a capo su una nuova linea quando si eccede la larghezza disponibile (vedi esempio).



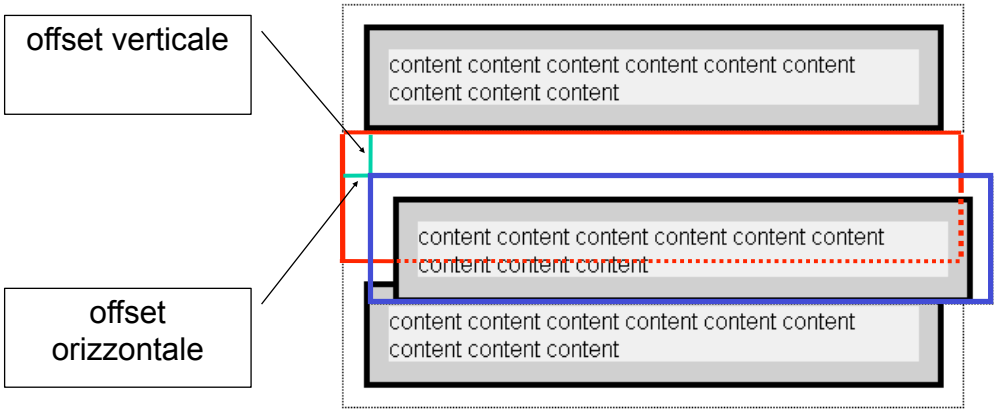
Posizionamento relativo

- La dichiarazione position: relative (a differenza della dichiarazione position: static) consente di spostare il box rispetto alla posizione determinata dal flusso normale.
- Si noti nell'esempio che i box vicini conservano la posizione determinata dal flusso normale (comprendendo anche il collassamento dei margini verticali) e che il box riposizionato può sovrapporsi agli altri box.



Posizionamento relativo - Valori di offset

- Lo spostamento viene determinato attraverso una combinazione di valori delle proprietà top | right | left | bottom.
- Ogni valore viene interpretato come la distanza alla quale il limite esterno del box corrispondente deve essere spostato rispetto alla sua posizione originaria nel flusso.
- Nel caso in cui vengano specificati valori sia per left che per right uno dei due verrà ignorato; analogamente succede per top e bottom.



Posizionamento relativo - 2

- Priorità di visualizzazione dei browser:
  - la specifica non è chiara a proposito della visualizzazione di box che si sovrappongono; generalmente i browser mostrano un box posizionato relativamente davanti agli elementi fratelli che lo precedono nel codice e dietro a quelli che lo seguono.

Posizionamento dei discendenti

- **Se l'elemento A**, **posizionato relativamente**, **è di tipo block**, questo stabilisce un nuovo blocco contenitore e un nuovo sistema di coordinate rispetto al quale si riferiranno i discendenti.
- In particolare, il posizionamento dei discendenti avverrà rispetto alla posizione del blocco contenitore.
- Possiamo dire in altri termini che **viene creata una nuova posizione per l'inizio del flusso normale**, che verrà utilizzata dagli elementi nidificati
- Ad esempio, nel caso di elementi figli caratterizzati da position:relative, questi verranno posizionati nel flusso a partire dalla posizione dell'elemento padre (la posizione del padre dovrà tenere conto di eventuali offset applicati); per determinare la posizione finale dei figli si dovrà poi tener conto di eventuali altri offset applicati ad essi.

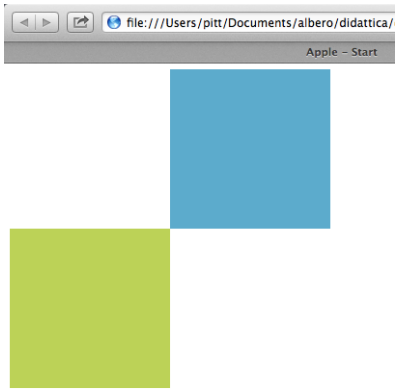
Posizionamento relativo dei discendenti - es. 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
<html lang="en">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example C</title>
    <style>
              .box { position: relative; width: 200px; height: 200px;}
              #box_1 { background: #ee3e64; }
              #box_2 { background: #44accf; left: 200px; }
              #box_3 { background: #b7d84b; }
    </style>
  </head>
  <body>
    <div id="box_1" class="box"></div>
    <div id="box_2" class="box"></div>
    <div id="box_3" class="box"></div>
  </body>
</html>
```

- Tutti e tre i box, gerarchicamente allo stesso livello, sono caratterizzati dalla proprietà position:relative; i box vengono posizionati nel flusso, uno di seguito all'altro; al secondo contenitore viene applicato un offset, calcolato rispetto alla propria posizione iniziale nel flusso

Posizionamento relativo dei discendenti - es. 2

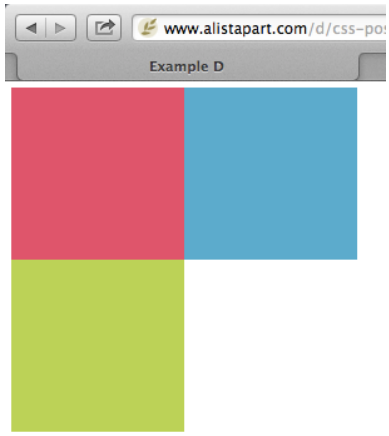
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```



- In questo esempio tutti e tre i box sono caratterizzati dalla proprietà position:relative, ma il secondo contenitore è nidificato dentro il primo.
- **L'elemento box_1 origina un nuovo sistema di coordinate**, rispetto al quale avviene il posizionamento dell'elemento figlio; in particolare, avendo l'elemento box_2 le stesse dimensioni di box_1, si sovrapporrà completamente all'elemento padre, rendendolo non visibile

Posizionamento relativo dei discendenti - es. 3

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #ee3e64;
        }
        #box_2 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #44accf;
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b;
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```



- Una variante dell'esempio precedente in cui è stata aggiunta una regola di offset per box_2; in questo caso dunque la posizione di box_2 verrà calcolata rispetto alla posizione di box_1 e poi verrà spostata a destra di 200px

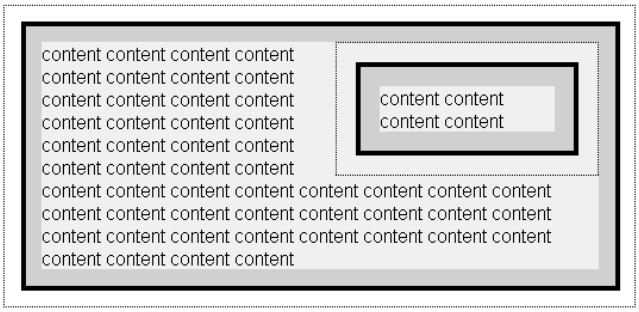
Posizionamento relativo dei discendenti - es. 4

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
        position: relative;
            left: 100px;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_4 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #cccccc; /* grey */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
        <div id="box_4"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```

- In questo esempio vengono descritti 4 elementi div, di cui due sono nidificati dentro il primo.
- Il primo elemento è caratterizzato dalla proprietà position:relative ed è spostato di 100 px a destra
- La posizione dei due elementi box_2 e box_4, nidificati dentro il primo, viene definita nel flusso a partire dalla posizione finale di box_1; successivamente i due elementi vengono spostati a destra, il primo di 100px, il secondo di 200px.

Floats

- Si ottiene settando la proprietà float di un elemento come left o right.
- Il box viene posizionato verticalmente come lo sarebbe stato nel flusso normale, ma orizzontalmente viene spostato quanto più possibile a destra o a sinistra nei limiti di contenuto del box contenitore; i contenuti inline che lo circondano possono fluire sul lato opposto.
-
```
<p>
     <span style="float:right;width:40%;(altri parametri
     omessi …)">
     content...
     </span>
     content content content content content content...
</p>
```



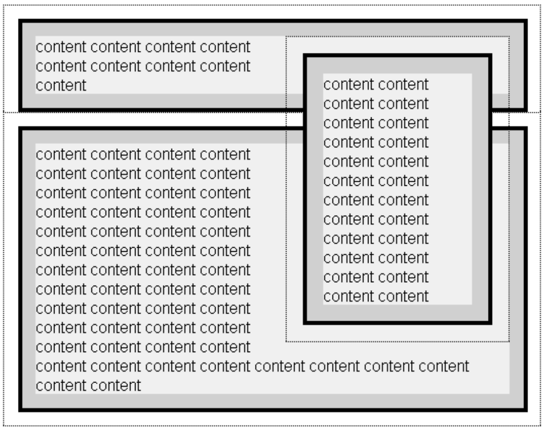
Floats - 2

- Un box floated dovrebbe avere associata una **larghezza**, esplicitamente o implicitamente. In assenza di essa, riempirà orizzontalmente il box contenitore, come gli elementi non-floated; **i box floated vengono sempre considerati come block box anche se sono definiti utilizzando elementi inline**.
- A differenza dei box del flusso normale, **i margini verticali di un box floated non collassano** con i margini di box sopra o sotto di esso.
- **Un box floated può sovrapporsi a block box** adiacenti ad esso **che si trovano nel flusso normale (vedi esempio seguente)**.

Floats - 3

- <p>
   content content content content content content content content content content content content content content content content...
  content content content content...
  </p>
  <p> content content content content content content content content... </p>

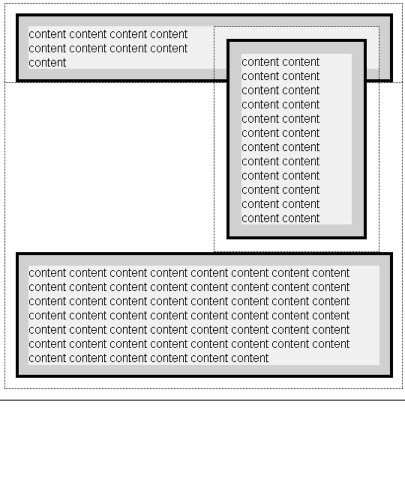
- Il block box floated generato dall'elemento span si dispone sul limite destro e superiore del contenuto dell'elemento p all'interno del quale è nidificato. Il testo contenuto in p fluisce intorno all'elemento span. Inoltre il box floated si sovrappone al secondo elemento p, che viene di seguito nel codice.



Floats - 4

- Si può evitare la sovrapposizione al secondo elemento p utilizzando la proprietà clear, settando clear:right; per il box generato dal secondo tag p; il margine esterno superiore del box associato a p viene incrementato per permettere al suo bordo e al suo contenuto di posizionarsi sotto il margine del floated box.

- <p>
   content content content content content content content content content content ...
  content content content content...
  </p>
  <p style="clear:right;"> content content content content content content content content... </p>



Floats - 5

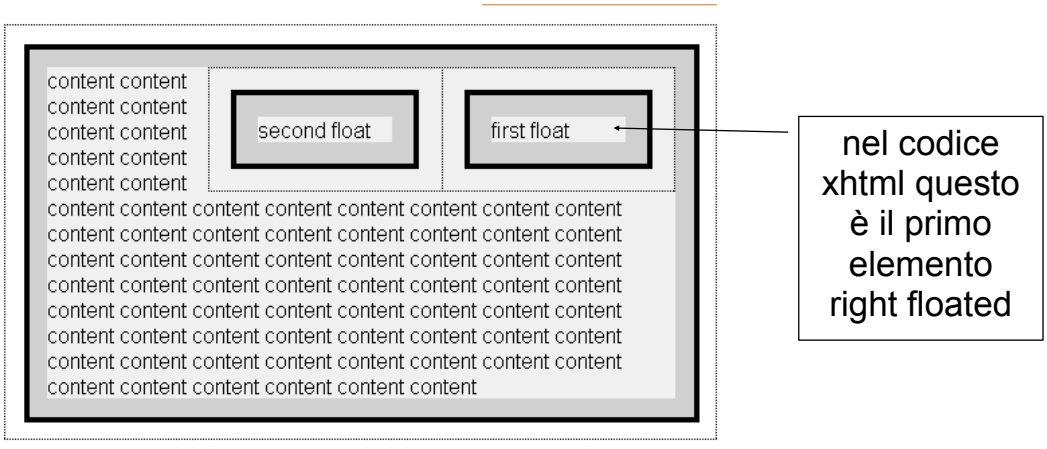
- Si noti che il box floated generato da span ancora si sovrappone al box generato dal primo tag p; si può evitare questo scrivendo il tag span al di fuori del primo paragrafo.

- content content content content content content content content content...
  content content content content...
  content content content content...

- La proprietà clear può essere settata a left right | both |inherit. Ha significato solo per gli elementi block.

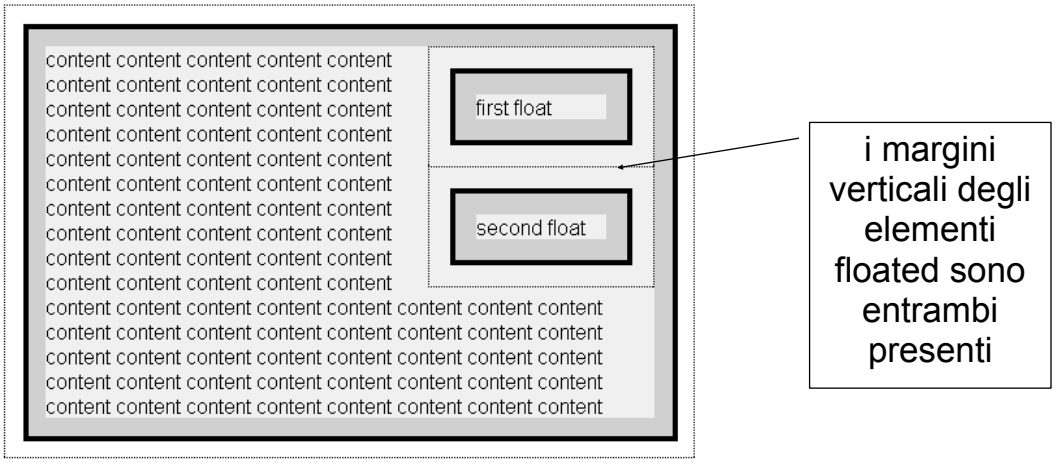
Floats adiacenti

- Quando due o più elementi adiacenti vengono floated, il loro limite superiore viene posizionato sulla stessa linea se c'è spazio orizzontale sufficiente. Se non c'è, gli elementi successivi vengono spostati in basso dove c'è spazio
- L'illustrazione mostra il posizionamento di due elementi right-floated. Si noti che il primo elemento floated appare più vicino al margine destro.



Floats adiacenti 2

- La proprietà clear può essere utilizzata su un elemento floated per forzarlo sotto float adiacenti. L'esempio illustra l'effetto della regola clear:right per il secondo elemento.
- Si noti che i margini verticali dei box floated non collassano.



Posizionamento assoluto

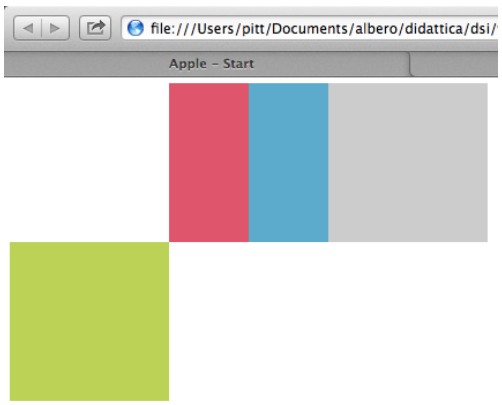
- Lo schema si applica a qualsiasi elemento che abbia la proprietà position settata ad absolute o fixed.
- A differenza di quello che succede per position static o relative, il box viene rimosso dal flusso, e non condiziona o viene condizionato da altri elementi che fanno parte del flusso
- Gli elementi posizionati in maniera assoluta sono sempre trattati come elementi block-level.
- La posizione viene determinata dai valori di offset attribuiti a top | right | bottom | left, che vengono utilizzati con le stesse modalità valide per il posizionamento relativo; ma a differenza di qust'ultimo, dove gli offset vengono misurati dalla posizione dell'elemento nel flusso normale, **gli spostamenti vengono calcolati rispetto al box contenitore**.

Posizionamento assoluto - 2

- **Qual'è il blocco contenitore?**
  - Il blocco contenitore nel caso del posizionamento assoluto è il suo **antenato più prossimo gerarchicamente** che ha proprietà position uguale ad absolute, relative o fixed. Se non c'è, viene usato il blocco contenitore iniziale (generalmente la **finestra del browser**).
- **Se l'elemento contenitore è di tipo block, gli offset vanno misurati dal limite esterno del padding** (e non del contenuto)

Posizionamento assoluto- es. 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
        position: absolute;
            left: 100px;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_4 {
            position: absolute;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #cccccc; /* grey */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
        <div id="box_4"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```



- In questo esempio gli elementi box_2 e box_4, nidificati dentro box_1, vengono posizionati in maniera assoluta; perciò vengono tolti dal flusso ed entrambi i loro offset vengono calcolati a partire dalla posizione di box_1. Si noti la differenza nel posizionamento rispetto all'esempio 4 di queste slides riferito al posizionamento relativo, che differisce solo per i valori della proprietà position per box_2 e box_4.

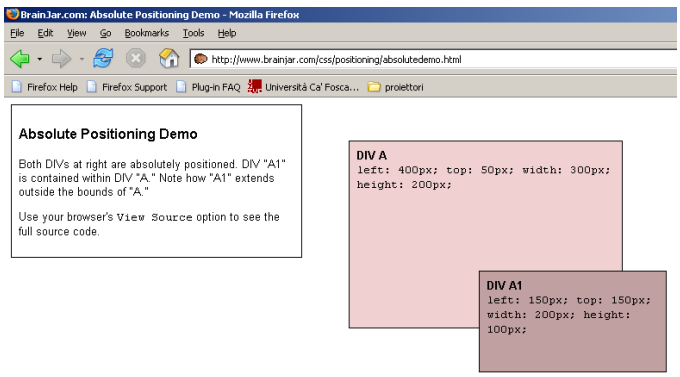
Posizionamento relativo dei discendenti - es. 4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
        position: relative;
            left: 100px;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_4 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #cccccc; /* grey */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
        <div id="box_4"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```

- ATTENZIONE: questa slide ripropone l'esempio 4 sul posizionamento relativo; si noti la posizione del box_3 e lo si confronti con la posizione del box_3 nella slide precedente.

Posizionamento dei discendenti

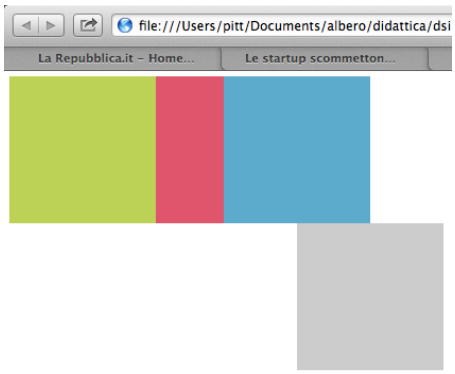
- Un elemento posizionato in maniera assoluta stabilisce un blocco contenitore per gli elementi interni ad esso; gli elementi che seguono obbediscono alle regole di posizionamento nella maniera consueta, eccettuato un offset corrispondente alla posizione dell'elemento contenitore.
- Questi elementi interni possono a loro volta essere posizionati in modalità assoluta; in questo caso vengono rimossi dal normale flusso del blocco contenitore.




```
…
<style type="text/css">
#A {
 padding: 8px;
  position: absolute;
  left: 400px;  top: 50px;
  width: 300px;  height: 200px; }
#A1 {
  margin: 0px;  padding: 8px;
  position: absolute;
  left: 150px;  top: 150px;
  width: 200px;  height: 100px; }
</style>
</head>
<body>
<div id="demoBox"> Absolute … </div>
<div id="A"> …
                <div id="A1"> … </div>
</div>
</body>
```

Pos. assoluto e posizionamento relativo dei discendenti - es. 2

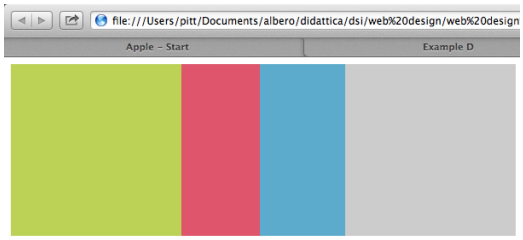
```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: absolute;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
        position: relative;
            left: 100px;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_4 {
            position: relative;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #cccccc; /* grey */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
        <div id="box_4"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```



- L'elemento box_1 viene posizionato in maniera assoluta, ma gli elementi box_2, box_3 e box_4 vengono posizionati in maniera relativa.
- Il posizionamento di box_1 avviene rispetto alla finestra del browser, mentre il posizionamento di box_2 e box_4 avviene come prima rispetto alla posizione di box_1, però a partire dalla loro posizione del flusso
- box_4 si posiziona nel primo spazio utile della finestra, essendo stato box_1 tolto dal flusso.

Pos. assoluto e posizionamento assoluto dei discendenti - es. 3

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Example D</title>
    <style>
        #box_1 {
            position: absolute;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #ee3e64; /* red */
        }
        #box_2 {
        position: absolute;
            left: 100px;
            width: 200px;
            height: 200px;
            background: #44accf; /* blue */
        }
        #box_4 {
            position: absolute;
            left: 200px;
            width: 200px;
            height: 200px;
            background: #cccccc; /* grey */
        }
        #box_3 {
            position: relative;
            width: 200px;
            height: 200px;
            background: #b7d84b; /* green */
        }
    </style>
</head>
<body>
    <div id="box_1">
        <div id="box_2"></div>
        <div id="box_4"></div>
    </div>
    <div id="box_3"></div>
</body>
</html>
```



- Variante dell'esempio precedente di posizionamento assoluto nel quale anche gli elementi box_2 e box_4 vengono posizionati in maniera assoluta; anch'essi vengono perciò tolti dal flusso.
- Il posizionamento di box_1 avviene rispetto alla finestra del browser, mentre il posizionamento di box_2 e box_4 avviene rispetto alla posizione di box_1
- Anche in questo caso si noti la posizione di box_4 (nel primo spazio utile della finestra), derivante dal fatto che box_1 è stato tolto dal flusso.

Fixed positioning

- Il posizionamento fixed è un caso speciale del posizionamento assoluto. Un elemento fixed non si muove quando la pagine web visualizzata sul browser viene navigata con lo scrolling.
- Nella stampa un elemento fixed dovrebbe essere stampato su ogni pagina.
- Il livello di supporto da parte dei browser per questo tipo di posizionamento sta aumentando gradualmente

Stacking order - 2

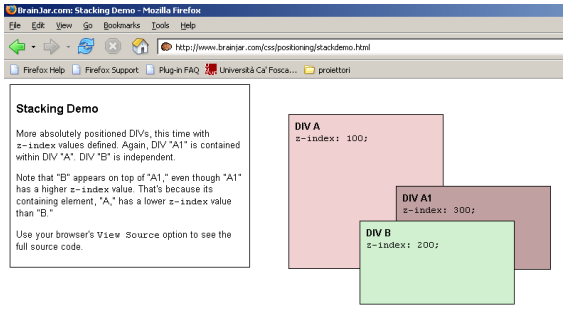
- Gli elementi posizionati in maniera assoluta possono sovrapporsi. Questo determina un problema di precedenza nella visualizzazione. Lo stacking context e la proprietà z-index determinano quale elemento dovrà apparire di fronte ad un altro.
- Un elemento posizionato in maniera assoluta crea uno stacking context locale che si applica agli elementi posizionati in maniera assoluta in esso contenuti; tra questi discendenti valgono regole di precedenza nella visualizzazione:
  - si considera prima il valore della proprietà z-index associata ai diversi elementi del contesto; **valori più elevati di z-index corrispondono ad una precedenza nella visualizzazione**;
  - **quando** due elementi hanno lo **stesso valore** viene considerato l'ordine degli elementi nel **codice sorgente**: **l'elemento definito prima viene visualizzato dietro**;
  - La specifica prevede anche valori negativi per z-index; un elemento con valore negativo viene mostrato dietro un qualsiasi altro con valore positivo o non definito (attenzione, i vecchi browser non gestiscono in maniera appropriata valori negativi di z-index).

Stacking order - 2

- Quando si sovrappongono elementi che appartengono a diverso stacking context diversi, questi vengono mostrati davanti o dietro a seconda del livello di stack del loro elemento contenitore definito dalla proprietà z-index

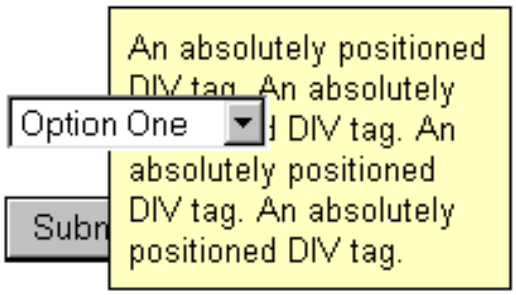
Stacking order - 3

```
…
<style type="text/css">
#A {background-color: #f0d0d0; border: solid 1px #000000; color: #000000;
  padding: 8px; position: absolute; left: 400px; top: 50px; width: 200px; height: 200px;
  z-index: 100;}
#A1 {background-color: #c0a0a0; border: solid 1px #000000; color: #000000;
  padding: 8px; position: absolute; left: 150px; top: 100px; width: 200px; height: 100px;
  z-index: 300;}
#B {background-color: #d0f0d0; border: solid 1px #000000; color: #000000;
  padding: 8px; position: absolute; left: 500px; top: 200px; width: 200px; height: 100px;
  z-index: 200;}
</style>
</head>
<body>
<div id="demoBox">...</div>
<div id="A"> …
          <div id="A1"> … </div>
</div>
<div id="B"> … </div>
</body>
```



Stacking order - 4

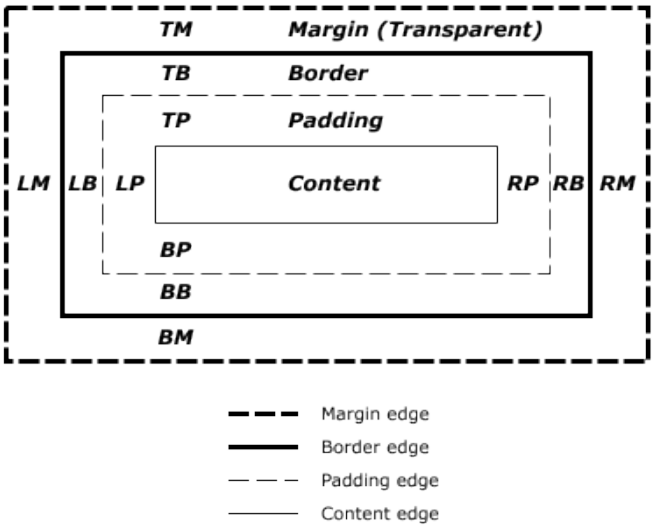
- Alcuni elementi come i controlli per moduli, gli applet e plug-in come Flash possono provocare problemi nel caso di sovrapposizioni; questo accade perchè i browser possono lasciare che altri programmi gestiscano la presentazione di questi elementi; in questi casi anche settare il valore di z-index non aiuta.
- L'unico rimedio possibile in questi casi consiste nell'evitare sovrapposizioni o, quando possibile, nell'utilizzare proprietà come visibility: hidden o display:none per nascondere gli elementi problematici.



Stacking order - 2

- Gli elementi posizionati in maniera assoluta possono sovrapporsi. Questo determina un problema di precedenza nella visualizzazione. Lo stacking context e la proprietà z-index determinano quale elemento dovrà apparire di fronte ad un altro.
- Un elemento posizionato in maniera assoluta crea uno stacking context locale che si applica agli elementi posizionati in maniera assoluta in esso contenuti; tra questi discendenti valgono regole di precedenza nella visualizzazione:
  - si considera prima il valore della proprietà z-index associata ai diversi elementi del contesto; valori più elevati di z-index corrispondono ad una precedenza nella visualizzazione;
  - **quando** due elementi hanno lo **stesso valore** viene considerato l'ordine degli elementi nel **codice sorgente**: **l'elemento definito prima viene visualizzato dietro**;
  - La specifica prevede anche valori negativi per z-index; un elemento con valore negativo viene mostrato dietro un qualsiasi altro con valore positivo o non definito (attenzione, i vecchi browser non gestiscono in maniera appropriata valori negativi di z-index).

The containing box



Si noti che per tutti gli elementi diversi dall'elemento root il riferimento per il loro posizionamento relativo è quello del **content edge** del box contenitore (**containing box)** generato dall'antenato più prossimo.

The containing box

Per quanto riguarda invece gli elementi con proprietà position:absolute il posizionamento nel layout verrà determinato utilizzando come riferimento la posizione dell'antenato più prossimo avente proprietà position uguale a relative, absolute o fixed.
In questo caso, se l'antenato è di tipo block-level, per il posizionamento si farà riferimento al **padding edge** del box contenitore box generato dall'antenato stesso.

Nel caso in cui non ci sia nessun antenato caratterizzato dalla proprietà position con valori value relative, absolute o fixed, si considererà come blocco contenitore l'**initial containing block**, i cui limiti corrispondono per i media continui ai confini della viewport del browser.

The initial containing box: nota

Si osservi che per gli elementi con position: absolute che fanno riferimento all'initial containing box, stabilire valori pari a zero e lasciare indefinite le proprietà di offset non è equivalente:

- nel primo caso si fa riferimento alla distanza dai limiti dell'initial containing block (cioè dai limiti della finestra del browser);
- nel secondo caso si fa riferimento ai limiti della finestra al netto dei margini (che di default non sono pari a zero).

Questo comportamento non si verifica quando si fa riferimento ad un containing block diverso (cioè in questo caso è equivalente stabilire valori pari a zero per l'offset o non dichiararli).

Alcune osservazioni finali sulla nidificazione degli elementi

In riferimento all'utilizzo di strutture di markup per la redazione di griglie di presentazioni, si osservi una marcata differenza nel comportamento delle strutture tabellari e delle strutture basate su div nel caso il cui le dimensioni dei contenuti eccedano quelle dei contenitori.

Posto che in entrambi i casi la condizione generalmente non è desiderabile e va evitata, si noti che:

nel primo caso la nidificazione di un elemento all'interno di una cella tale che le dimensioni dell'elemento eccedano quelle prefissate del contenitore td si risolve in un allargamento della cella e della tabella, per far si che il contenuto nidificato possa essere ospitato;
nel secondo caso invece il contenuto nidificato si espande oltre i limiti dell'elemento contenitore, ma le dimensioni di quest'ultimo rimangono invariate.

Proprietà dello sfondo

- Background-attachment
  - Specifica se l'immagine deve scorrere insieme al documento oppure deve rimanere in una posizione fissa; valori scroll|fixed
    - body {background-image: url(pinco.gif); background-attachment: scroll}
- Background-position
  - Per specificare la posizione dell'immagine di sfondo
    - p.avviso {background-color: lime}
- Background
  - E' una forma abbreviata per specificare tutte le caratteristiche del background in una sola regola

Proprietà dello sfondo

- Alcune **linee guida per l'uso delle immagini di sfondo**
  - Utilizzare un'immagine che non interferisca con la leggibilità del testo
  - Utilizzare un'immagine di dimensioni (in Kbyte) piccole
  - Specificare un colore di sfondo di tonalità simile a quella dell'immagine, che possa essere visualizzato mentre l'immagine si carica
  - Utilizzare lo stesso formato grafico se si vogliono ottenere le stesse tonalità di colore nelle immagini di sfondo e in quelle inserite all'interno della pagina (es. GIF con GIF, JPEG con JPEG)
  - Se si utilizzano colori non web-safe, ci possono essere effetti non previsti per gli utenti che utilizzano schermi a 256 colori, dipendenti dalla particolare implementazione del browser

Block box (box contenitori)

- I block box agiscono anche come contenitori per altri box nidificati all'interno di essi.
- Per semplificare il processo di posizionamento, la specifica CSS stabilisce che **un box di tipo block può contenere box appartenenti alla categoria block oppure (esclusivo) alla categoria inline**. Per soddisfare questa prescrizione in alcuni casi alcuni box di tipo block vengono generati come wrapper di box inline.

- ```

       This is the first sentence.
       This is the second sentence.

  ```
- In questo esempio viene generato un block box associato a div e un block box nidificato associato al paragrafo p. Per quanto detto sopra, viene generato un ulteriore block box per la stringa di testo anonima `This is the first sentence` (che di default darebbe luogo ad un inline box).

Posizionamento relativo - 2

- Compatibilità dei browser:
  - la specifica non è chiara a proposito della visualizzazione di box che si sovrappongono; generalmente i browser mostrano un box posizionato relativamente davanti agli elementi fratelli che lo precedono nel codice e dietro a quelli che lo seguono.
  ```