



.ConStile è una guida dedicata all'uso dei CSS per lo sviluppo di siti web leggeri, usabili, conformi agli standard W3C.

[Home](#) > [Tutorial](#) > [Come dimensionare i caratteri](#) > Dimensionare i caratteri con i CSS

#### Articoli correlati:

- [Il box model di IE5/Win \[/tutorial/IE5\\_box\\_model/\]](#)
- [Progettare con i CSS \[/tutorial/progettare\\_con\\_stile/\]](#)
- [Pagine pronte per la stampa con i CSS \[/tutorial/stampare\\_con\\_i\\_css/\]](#)

#### Essenziali

- [99.9% of Websites are Obsolete \[http://www.digital-web.com/features/feature\\_2002-09.shtml\]](#)
- [Progettare con i CSS \[/tutorial/progettare\\_con\\_stile/\]](#)
- [CSS e tabelle a confronto \[/tutorial/css\\_vs\\_table/\]](#)
- [Introduzione ai CSS \[/tutorial/introduzione\\_ai\\_css/\]](#)
- [Il box model e l'errata interpretazione di IE5/Win \[/tutorial/IE5\\_box\\_model/\]](#)

#### Strumenti

- [Commenti sull'articolo \[#commenti\]](#)
- [Stampa l'articolo \[javascript:self.print\(\)\]](#)
- [Aggiungi questo articolo al tuo Blocco Note \[/adminpreferiti.php?a=add&url=/tutorial/come\\_dimensionare\\_i\\_caratteri/&t=Dimensionare\\_i\\_caratteri\\_con\\_i\\_CSS\] \[ 1 \]](#)
- [Aggiungi questo articolo al tuo Blocco Note \[/adminpreferiti.php?a=add&url=/tutorial/come\\_dimensionare\\_i\\_caratteri/&t=Dimensionare\\_i\\_caratteri\\_con\\_i\\_CSS\] \[ 1 \]](#)

## Dimensionare i caratteri con i CSS

I CSS vengono spesso utilizzati per dimensionare i caratteri, ma se male utilizzati possono peggiorare l'accessibilità di un sito. Con questo articolo si cercherà di fornire una piccola guida su come dimensionare i caratteri rispettando l'accessibilità del sito.

### Come definire la grandezza del font

Il codice CSS per definire la grandezza dei font è il seguente:

```
selettore {  
  font-size: valore;  
}
```

In base a *valore* è possibile definire la dimensione dei font del *selettore*.

### I valori che è possibile assegnare alla proprietà font-size

Alla proprietà font-size è possibile assegnare i seguenti valori:

- **dimensioni assolute:** i possibili valori sono: xx-small | x-small | small | medium | large | x-large | xx-large
- **dimensioni relative:** i possibili valori sono: larger | smaller
- **unità di lunghezza:** le unità di lunghezza si suddividono in due classi:
  - **unità di lunghezza assolute**, di cui quelle adatte a definire le dimensioni dei caratteri sono:
    - pt
    - pc
  - **unità di lunghezza relative**, di cui quelle adatte a definire le dimensioni dei caratteri sono:
    - em
    - px
- **percentuali**

#### Dimensioni assolute

```
.class { font-size: small; }  
p { font-size: medium; }  
h1 { font-size: xx-large; }
```

Il dimensionamento assoluto si basa su 7 parole chiave (xx-small | x-small | small | medium | large | x-large | xx-large)

che, come chiaramente indicato dal nome, impostano la dimensione dei caratteri specificando se il carattere debba essere estremamente piccolo, molto piccolo, piccolo, medio, grande, molto grande, estremamente grande.

Ovviamente le parole chiave piccolo e grande, come pure tutte le altre, non avrebbero significato se non ci fosse un riferimento: piccolo rispetto a cosa? Il riferimento è la dimensione (del font) standard dello User Agent (UA) dell'utente.

L'uso di queste parole chiave comporta un doppio vantaggio: per il webmaster e per l'utente. Il webmaster ha a disposizione sette dimensioni, dall'estremamente piccolo all'estremamente grande, che risultano semplici da usare e garantiscono coerenza alle pagine web (spesso, purtroppo, capita di vedere pagine web in cui i caratteri assumono le più svariate dimensioni). L'utente ha il vantaggio di poter variare la dimensione del carattere di default, in questo modo può aumentare la dimensione di caratteri troppo piccoli o ridurre quella di caratteri troppo grandi.

Purtroppo queste parole chiave non vengono interpretate correttamente da molti browser, così da renderle utili solo in casi limitati (ad esempio proiezioni in conferenze). In futuro saranno certamente sfruttate al meglio.

## Dimensioni relative

```
.nota { font-size: smaller; }  
.titolo { font-size: xx-larger; }
```

Il dimensionamento relativo si basa su due sole parole chiave: larger (più grande di) smaller (più piccolo di). Il riferimento è la dimensione dell'elemento parent. L'elemento parent è quello che contiene l'elemento preso in considerazione. Ad esempio, nel seguente codice:

```
<div id="elementoA">  
  ...  
  <div id="elementoB">  
    ...  
    <div id="elementoC">  
      ...  
    </div>  
  </div>  
</div>
```

l'elemento B è il parent dell'elemento C, l'elemento A è il parent dell'elemento B.

I vantaggi offerti da queste due parole chiave sono soprattutto per l'utente. Il webmaster ha a disposizione un elemento poco elastico, ma che può tornare utile in molti casi, mentre l'utente ha gli stessi vantaggi offerti dal dimensionamento assoluto.

Come per il dimensionamento assoluto, molti browser non supportano sufficientemente queste parole chiave.

## Unità di lunghezza

Le unità di lunghezza si dividono in unità di lunghezza assolute e unità di lunghezza relative.

### Unità di lunghezza assolute

```
.class { font-size: 10pt; }  
p { font-size: 12pt; }  
h1 { font-size: 2pc; }
```

Le unità di lunghezza assolute adatte a definire le dimensioni dei caratteri sono:

- pt: punti;
- pc: picas (1pc=12pt).

Il risultato è fortemente dipendente dal dispositivo con cui si visiona la pagina e non dovrebbero mai essere utilizzate per il media 'screen' ([maggiori dettagli \[/tips/css\\_e\\_media\\_type/\]](#)). Questo tipo di unità di misura è molto più utile con media quali la stampante (vedi: "[Pagine pronte per la stampa con i CSS \[/tutorial/stampare\\_con\\_i\\_css/\]](#)").

### Unità di lunghezza relative

Le unità di lunghezza relative adatte a definire le dimensioni dei caratteri sono:

- em: se utilizzata per definire la dimensione dei font, l'unità 'em' è la dimensione del carattere dell'elemento parent (ad esempio 0.80em indicano l'80% della dimensione del carattere dell'elemento parent);

- px: il pixel, la dimensione di un pixel dipende dalla risoluzione del dispositivo che si utilizza per visualizzare le pagine.

#### L'unità 'em'

```
.class { font-size: .80em; }  
p { font-size: 1.00em; }  
h1 { font-size: 1.80em; }
```

L'unità 'em' offre vantaggi e svantaggi al webmaster, e vantaggi all'utente.

Il vantaggio per l'utente sta nel fatto che, come per le lunghezze assolute (xx-smal, medium, large, ...) e le lunghezze relative (larger, smaller), è possibile aumentare o ridurre la dimensione dei font, migliorando la leggibilità, agendo sul browser.

La comodità per il webmaster consiste nella coerenza che è possibile dare alle pagine. Si possono infatti scegliere alcuni valori percentuali (ad esempio  $80\%=0.80em$  e  $125\%=1.25em$  -- si noti come  $0.80*1.25=1.00$ ) con i quali incrementare e ridurre i font su schermo e mantenere omogenee le dimensioni dei font. Questa comodità può però essere uno svantaggio. Annidando troppi elementi, si può raggiungere in breve tempo una dimensione troppo piccola ovvero troppo grande per il testo. Si consideri ad esempio il codice di seguito riportato:

CSS

```
[...]  
.smaller { font-size: 0.80em }  
.bigger { font-size: 1.25em }  
[...]
```

(X)HTML

```
<div>Questo testo è come il corpo dell'articolo  
<div class="smaller">  
  Riuscite a leggere questo testo?  
  <div class="smaller">  
    Riuscite a leggere questo testo?  
    <div class="smaller">  
      Riuscite a leggere questo testo?  
      <div class="smaller">  
        Riuscite a leggere questo testo?  
      </div>  
    </div>  
  </div>  
</div>  
</div>
```

Il risultato è riportato qui di seguito:

Questo testo è come il corpo dell'articolo  
Riuscite a leggere questo testo?  
Riuscite a leggere questo testo?  
Riuscite a leggere questo testo?  
Riuscite a leggere questo testo?

Probabilmente dalla terza riga in poi, il testo risulterà illeggibile.

Il problema può essere superato con un minimo di accortezza in fase di creazione dei fogli di stile. Probabilmente, soprattutto se i selettori sono specificati con una certa intelligenza e per uno scopo specifico (la classe smaller qui utilizzata è solo didattica e non ha uno scopo specifico), non sarà necessario specificare a molte classi la dimensione dei font, prestando la dovuta attenzione, non si avranno questi tipi di problemi.

Bisogna però specificare che NN4 e IE3 non supportano bene l'unità 'em'.

#### L'unità 'px'

```
.class { font-size: 10px; }  
p { font-size: 12px; }  
h1 { font-size: 20px; }
```

L'unità px offre al webmaster il grosso vantaggio di poter specificare la dimensione esatta in pixel (anche se alcuni browser, come NN4 possono restituire testo più piccolo di quello specificato). Il webmaster dovrà però stare attento a

non creare di fogli di stile che generino confusione, preoccupandosi di salvaguardare la coerenza del testo.

Bisogna però prestare molta attenzione alle dimensioni scelte. Se il font fosse troppo piccolo, utenti dotati di una vista non ottima, potrebbero avere difficoltà di lettura e abbandonare il sito. Infatti se browser come Opera o Mozilla permettono di zoomare sul testo indipendentemente dalle impostazioni dei CSS, il browser più diffuso, IE/Win, non riesce a ingrandire il testo attraverso il menu Visualizza > Carattere. In questo caso, potrebbe essere compromessa l'usabilità e l'accessibilità del sito. Inoltre lo script precedentemente illustrato non potrebbe funzionare (vedi la [pagina di esempio \[come dimensionare i caratteri.html\]](#)).

## Percentuali

```
.class { font-size: 80%; }  
p { font-size: 100%; }  
h1 { font-size: 180%; }
```

Le unità percentuali funzionano come le unità di misura 'em' e sono dunque caratterizzate dai medesimi pregi e difetti.

## Quale via percorrere?

Se non fosse per l'incapacità di Internet Explorer di ridimensionare i caratteri specificati in pixel, il metodo più semplice e funzionale per stabilire la grandezza dei font sarebbe specificare le dimensioni in pixel. Purtroppo IE è il browser in assoluto più diffuso e il problema dell'accessibilità non può essere trascurato.

La migliore scelta rimane dunque quella di adottare le unità percentuali e/o l'unità 'em', prestando attenzione alle proprietà degli elementi parent.

Utilizzando le percentuali è nella definizione della dimensione del testo del BODY e per gli altri elementi del documento, per modificare la dimensione del testo di tutto il documento è sufficiente modificare la dimensione nel body:

```
body { font-size: 70% }  
  
oppure  
  
body { font-size: 80% }  
  
oppure  
  
body { font-size: 100% }
```

La proprietà font-size del BODY può addirittura essere modificata "al volo" attraverso l'uso del codice Javascript:

```
/* Codice Javascript */  
function setFontSize(value) {  
  theStyle = null;  
  if (value == "small") theStyle = "70%";  
  if (value == "medium") theStyle = "80%";  
  if (value == "large") theStyle = "100%";  
  if (theStyle)  
    document.getElementsByTagName("body")[0].style.fontSize = theStyle;  
}  
/* funziona solo con un browser  
che supporti bene il DOM come IE5+, NN6, Moz. 1.  
Non funziona con Opera 6*/
```

Un'esempio del funzionamento del codice è disponibile nella [pagina di esempio \[come dimensionare i caratteri.html\]](#).

G.T. -- ultima revisione: 26.09.2002

