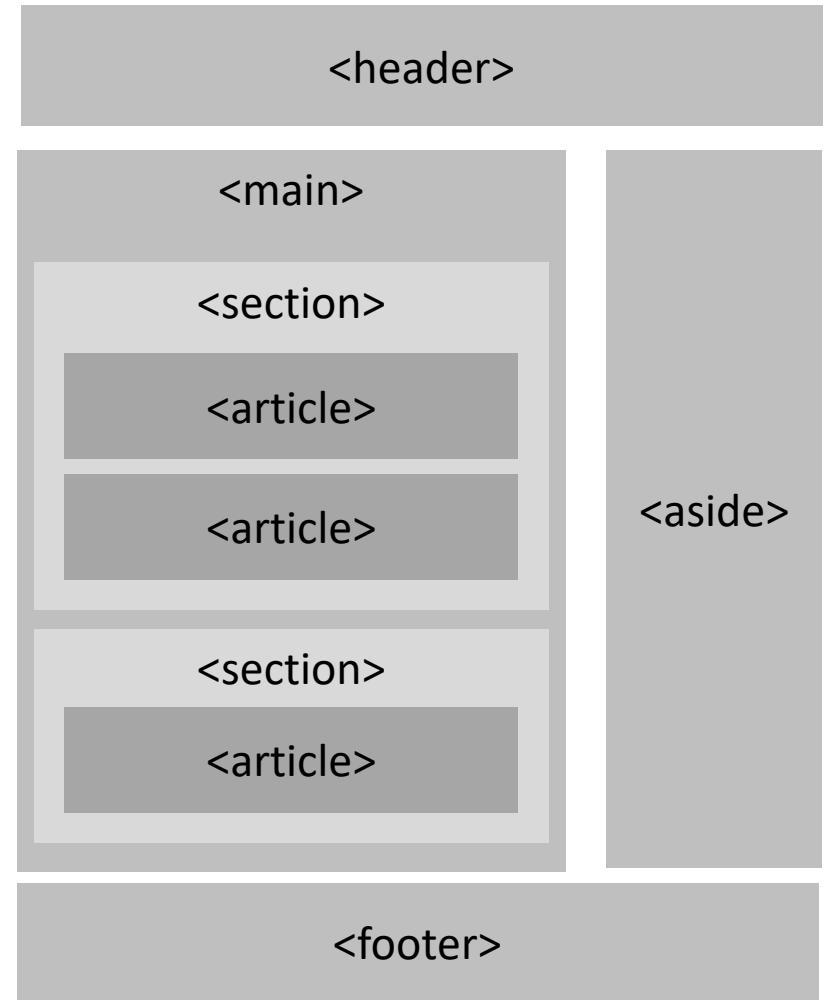


# HTML SEMANTICO

# Organizzare il contenuto

Esigenza: raggruppare logicamente degli elementi

- Prima di HTML5:
  - block division: `<div>`
  - inline division: `<span>`
- HTML5
  - header: `<header>`
  - footer: `<footer>`
  - main: `<main>`
  - sezioni: `<section>`
  - articoli: `<article>`
  - navigazione: `<nav>`
  - contenuto a parte: `<aside>`



- Nati da uno studio di Google:  
[code.google.com/webstats/2005-12/classes.html](http://code.google.com/webstats/2005-12/classes.html)

# main

- Il tag `<main>` specifica il contenuto principale di un documento.
  - Il contenuto all'interno dell'elemento `<main>` dovrebbe essere univoco per il documento.
  - Ci può essere un solo elemento `<main>` in un documento.
  - NON deve essere un discendente di un elemento `<article>`, `<aside>`, `<footer>`, `<header>` o `<nav>`.

[https://www.w3schools.com/tags/tag\\_main.asp](https://www.w3schools.com/tags/tag_main.asp)

# Aside (sidebar)

- `<aside>...</aside>`
- Sono elementi correlati ma "marginali" rispetto al contenuto
  - Non li potevano chiamare sidebar perché "side" si riferisce alla presentazione
- Uso tipico
  - sondaggi
  - citazioni
  - informazioni aggiuntive
  - annunci
  - link
- non ha un aspetto di default

# <nav> navigator

- Sezione che permette la navigazione nel sito
  - tipicamente si mettono delle liste nel <nav>
    - non tutte le liste vanno nel nav!!
- Nei siti responsive spesso viene compressa per i dispositivi piccoli

```
<nav>
  <ul>
    <li><a href="">Serif</a>/li>
    <li><a href="">Sans-serif</a></li>
    <li><a href="">Script</a></li>
    <li><a href="">Display</a></li>
    <li><a href="">Dingbats</a>/li>
  </ul>
</nav>
```



# Header

- `<header> </header>`
- Si usa per materiale introduttivo
- Due usi comuni
  - header della pagina
  - header di una sezione o articolo
- Se usato in un articolo tipicamente contiene titolo, autore, data, etc.

```
<header>
  
  <hgroup>
    <h1>Nuts about Web Fonts</h1>
    <h2>News from the Web Front</h2>
  </hgroup>
  <nav>
    <ul>
      <li><a href="">Home</a></li>
      <li><a href="">Blog</a></li>
      <li><a href="">Shop</a></li>
    </ul>
  </nav>
</header>

... page content ...
```

```
<article>
  <header>
    <h1>More about WOFF</h1>
    <p>by Jennifer Robbins,<time
datetime="11-11-2011" pubdate>November
11, 2011</time></p>
  </header>
  <p>...article content starts here...</p>
</article>
```

# Footer

- Contiene le info che vanno alla fine di sezione logica

- autore, copyright
- documenti correlati
- link

- Usi comuni:

- fondo pagina
- fine article o section

```
<article>
  <header>
    <h1>More about WOFF</h1>
    <p>by Jennifer Robbins, <time datetime="11-11-2011"
pubdate>November 11, 2011</time></p>
  </header>
  <p>...article content starts here...</p>
  <footer>
    <p><small>Copyright &copy;2012 Jennifer Robbins.</small></p>
    <nav>
      <ul>
        <li><a href="">Previous</a></li>
        <li><a href="">Next</a></li>
      </ul>
    </nav>
  </footer>
</article>
```

# <section> ... </section>

- Sezioni tematiche
  - possono contenere heading, paragrafi
  - tutti gli elementi che abbia senso raggruppare
- Può essere usato in diversi contesti
  - da dividere le sezioni della pagina
  - a dividere le sezioni di un articolo

```
<section>
  <h2>Typography Books</h2>
  <ul>
    <li>...</li>
  </ul>
</section>
<section>
  <h2>Online Tutorials</h2>
  <p>These are the best
tutorials on the web.</p>
  <ul>
    <li>...</li>
  </ul>
</section>
```



# <article> ... </article>

- Porzioni di pagina auto-consistenti
  - teoricamente riusabili in contesti differenti
  - e.g. syndacation: porzioni a disposizione di altri siti
- Contesti operativi:
  - blog
  - giornali

# articoli e sezioni

## Sezioni di un articolo:

```
<article>
<h1>Get to Know Helvetica</h1>
<section>
  <h2>History of Helvetica</h2>
  <p>...</p>
</section>
<section>
  <h2>Helvetica Today</h2>
  <p>...</p>
</section>
</article>
```

## Sezione di articoli:

```
<section id="essays">
  <article>
    <h1>A Fresh Look at Futura</h1>
    <p>...</p>
  </article>
  <article>
    <h1>Getting Personal with
    Humanist</h1>
    <p>...</p>
  </article>
</section>
```

# figure

- Un'immagine e una didascalia possono essere raggruppate in un elemento `<figura>`.
  - L'elemento `<img>` definisce l'immagine, l'elemento `<figcaption>` definisce la didascalia.

```
<figure>
  
  <figcaption>Fig1. - Trulli, Puglia,
Italy.</figcaption>
</figure>
```

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_figcaption](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_figcaption)

# head vs header vs h1,h2,..., h6

- `<header>`
  - elemento strutturale che identifica un segmento di pagina
  - posto all'interno del body o di u suo sotto-elemento
- `<head>`
  - direttamente incluso nel tag html
  - non mostrato dal browser
  - contiene i metadati
- `<h1>,<h2>,...,<h6>`
  - identificano i titoli testuali della pagina



# ELEMENTI HTML GENERICI

# Elementi generici

Se nessun elemento descrive il vostro contenuto

- `<div>` block element generic
  - divisione di contenuto
- `<span>` inline element generico
  - identifica una parola o una frase
- Assumono significato insieme agli attributi **id** e **class**
  - usati per motivi grafici
  - o per legare azioni con js

# Esempi di <div> e <span>

```
<div class="listing">
  
  <p><cite>The Complete Manual of Typography</cite>, James Felici</p>
  <p>A combination of type history and examples of good and bad type
design.</p>
</div>
```

```
<div id="news">
  <h1>New This Week</h1>
  <p>We've been working on...</p> <p>And last but not least,... </p>
</div>
```

```
<ul>
  <li>John: <span class="tel">999.8282</span></li>
  <li>Paul: <span class="tel">888.4889</span></li>
  <li>George: <span class="tel">888.1628</span></li>
  <li>Ringo: <span class="tel">999.3220</span></li>
</ul>
```



# identificatori per id e class

- **id**
  - il valore assegnato va usato una sola volta nel documento
- **class**
  - classifica gli elementi in gruppi concettuali
- **identificatori validi**
  - deve iniziare con una lettera A–Z o a–z o un underscore
  - non possono contenere spazi
  - può contenere lettere, numeri, trattini, underscore, due punti, punti,
  - sono case sensitive



# SELETTORI

# Selettori Semplici

- Tutti gli elementi
- Selezionare con il tag

Es.: `*{text-decoration: none}`

Es.: `p{text-align: center;color: red}`

- Selezionare con la classe

- Es.: `.center-red{text-align: center;color: red}`

- Es.: `p.center-red{text-align: center;color: red}` → Elemento + Classe

- `<p class="center-red">...</p>`

- `<ul class="center-red">...</ul>`

- Selezionare **un solo** elemento con id

- Es.: `#news{text-align: center;color: red}`

- `<div id="news">...</div>`

- Nota: l'ID nella pagina deve essere unico!!!

# Ereditarietà

- Alcune proprietà sono ereditate dai **discendenti**
  - la dichiarazione color viene passa allo span
  - le altre altre no

```
<!DOCTYPE html>
<html>
<head>
<style>

p{
  color:white;
  background:grey;
  border: medium solid black;
}

</style>
</head>
<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph <span>with a span</span> inside.</p>
</body>
</html>
```

**This is a Heading**

This is a paragraph with a span inside.

# DOM - Albero degli elementi

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Titolo</title>
```

```
  <style>
```

```
    h1{ color: red;}
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Titolo 1</h1>
```

```
  <p>paragrafo</p>
```

```
  <p>paragrafo con un elemento <em>importante</em></p>
```

```
  <h2>Titolo 2</h2>
```

```
  <p>paragrafo con immagine<img src=""></p>
```

```
  <p>paragrafo</p>
```

```
  <p>paragrafo con due <em>elementi</em> <em>importanti</em></p>
```

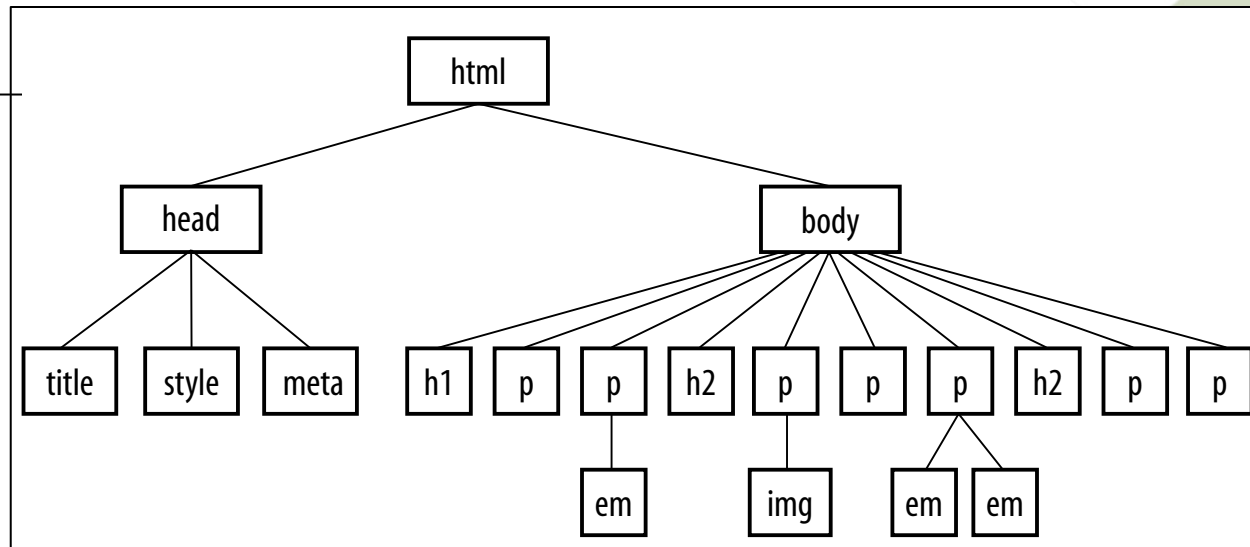
```
  <h2>Titolo 2</h2>
```

```
  <p>paragrafo</p>
```

```
  <p>paragrafo</p>
```

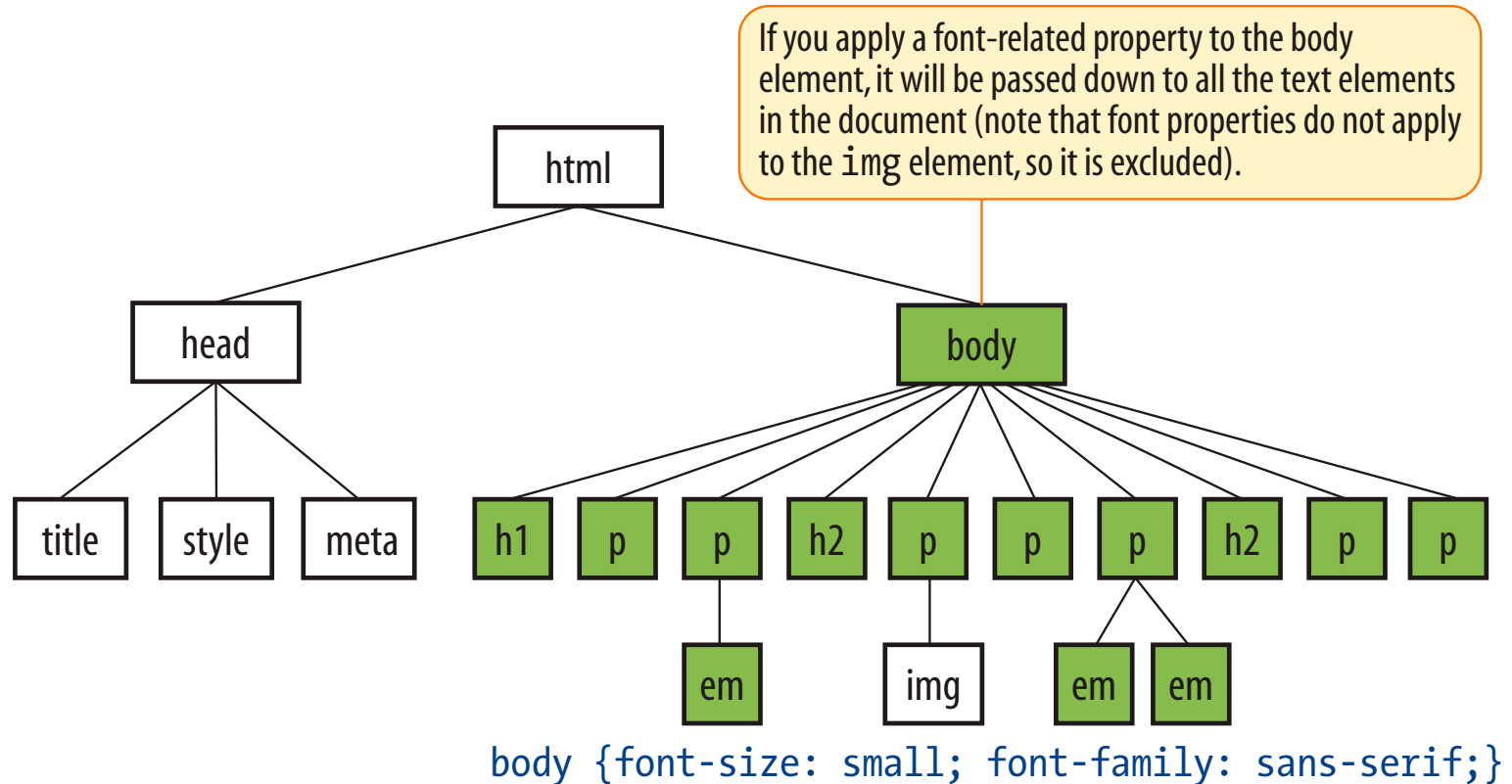
```
</body>
```

```
</html>
```



# Dichiarazioni ereditate

- Gli stili relativi ai font sono ereditati dai descendant



# Proprietà ereditate

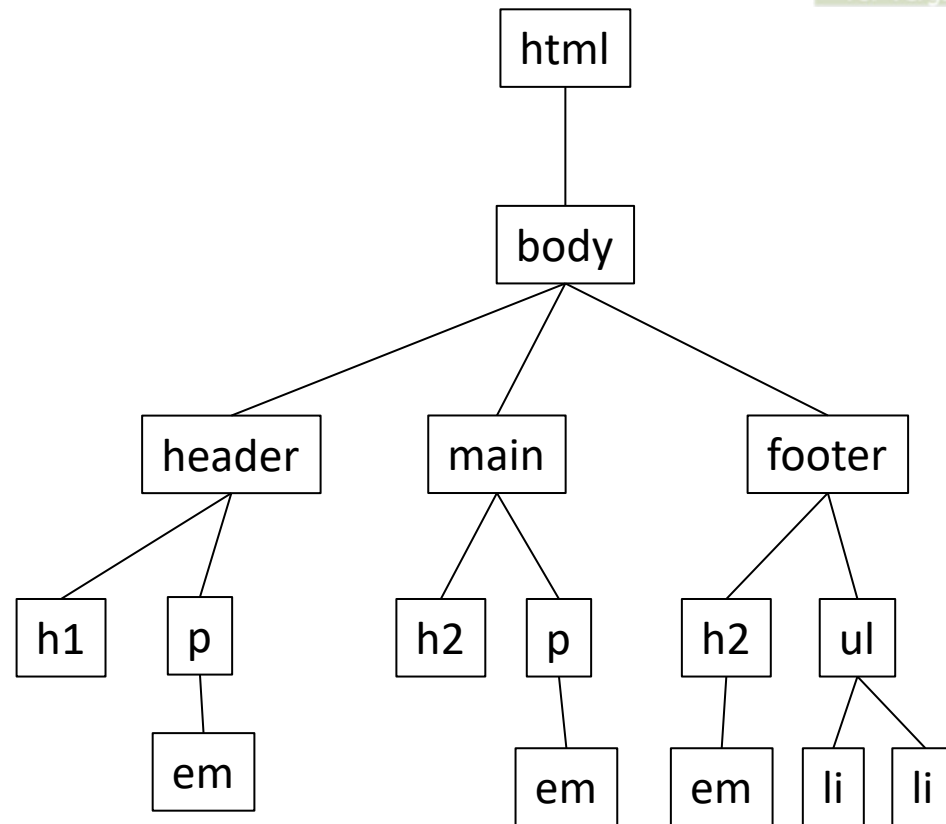
- azimuth
- border-collapse
- border-spacing
- caption-side
- color
- cursor
- direction
- elevation
- empty-cells
- font-family
- font-size
- font-style
- font-variant
- font-weight
- font
- letter-spacing
- line-height
- list-style-image
- list-style-position
- list-style-type
- list-style
- orphans
- pitch-range
- pitch
- quotes
- stress
- text-align
- text-indent
- text-transform
- visibility
- voice-family
- white-space
- widows
- word-spacing



# SELETTORI AVANZATI

# Relazioni nel DOM

- **Descendant** - discendenti
  - Gli elementi contenuti in un elemento sono i suoi discendenti
- **Child** - figli
  - Discendenti diretti e viceversa si dice genitore (parent)
- **Ancestor** - antenati
  - Gli elementi sopra nell'albero
- **Parent** – genitore
  - Elementi direttamente sopra
- **Sibling** – fratelli
  - Elementi con lo stesso parent



# Selettori composti

1. Selettori per descendant (spazio)
2. Selettori per child (>)
3. Selettori per Adjacent sibling (+)
  - il fratello immediatamente successivo
4. Selettori per General sibling (~)
  - tutti i fratelli successivi

# Selezionare i discendenti 1

`p em {color: grey;}`

`header p {font-size: 20px;}`

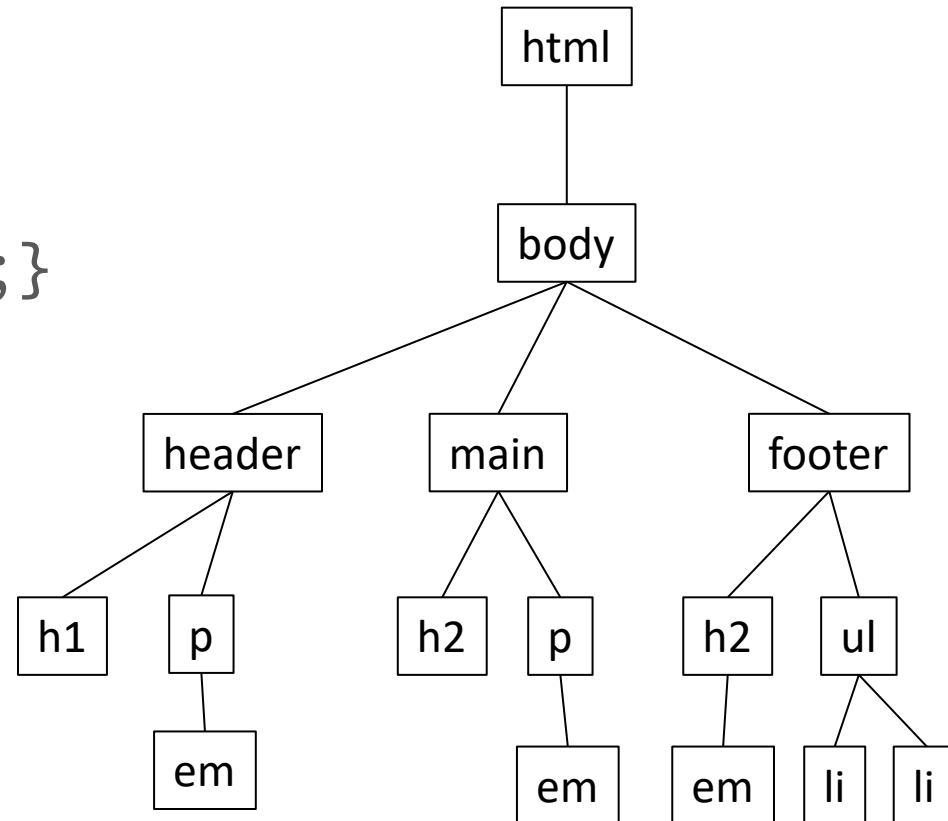
`main p {font-size: 16px;}`

`footer h2 {...}`

`main h2 {...}`

`header em {...}`

`main em {...}`



# Selezionare i discendenti 2

```
p.ingredienti {...}
div.ricetta {...}
```

```
.ricetta p {...}
.ricetta h2 {...}
```

```
.ricetta .ingredienti {...}
div.ricetta p.ingredienti {...}
```

```
<div class="ricetta">
  <h2>...</h2>
  <p>...</p>
  <p class="ingredienti"> ...</p>
</div>

<div class="ricetta">
  <h2>...</h2>
  <p>...</p>
  <p class="ingredienti"> ...</p>
</div>
```

# Selezionare i child

```
div > p {  
    background-color: yellow;  
}
```

- Tutti i **p** figli (child) di un **div**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_gt](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_gt)

# Selezionare fratelli

```
div + p {  
    background-color: yellow;  
}
```

- Tutti gli elementi **p** immediatamente successivi ad un **div**  
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_pluss](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_pluss)

```
div ~ p {  
    background-color: yellow;  
}
```

- Tutti gli elementi **p** fratelli successivi degli elementi **div**  
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_tilde](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_tilde)

# Pseudo Classi

- Una pseudo-classe viene utilizzata per definire uno stato speciale di un elemento.
  - mouse over
  - visited e unvisited link
  - focus

Esempio: stili ancora

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_link](https://www.w3schools.com/css/tryit.asp?filename=trycss_link)

Elenco

[https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

```
selector:pseudo-class {  
    property:value;  
}
```

```
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}
```



# Altre pseudo classi

- :root
- :only-child
- :empty
- :first-of-type
- :first-child
- :last-of-type
- :last-child
- :only-of-type
- :nth-child()
- :nth-last-child()
- :nth-of-type()
- :nth-last-of-type()

[https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

# Pseudo elementi

- Uno pseudo-elemento viene utilizzato per dare uno stile alle parti specifiche di un elemento.
  - Disegna la prima lettera o riga di un elemento
  - Inserire un contenuto prima o dopo un elemento

- Selettori

- ::after
- ::before
- ::first-letter
- ::first-line
- ::selection

```
selector::pseudo-element {  
    property:value;  
}
```

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

[https://www.w3schools.com/css/css\\_pseudo\\_elements.asp](https://www.w3schools.com/css/css_pseudo_elements.asp)

# Selettori con attributi

- È possibile impostare lo stile su elementi HTML con attributi o valori di attributo specifici.

```
[attribute] {  
  property: value;  
}
```

```
img[alt] {  
  background-color: grey;  
}
```

```
[attribute=value] {  
  property: value;  
}
```

```
a[target="_blank"] {  
  color: red;  
}
```

- è possibile selezionare parte del valore:
  - parola, inizio, fine, etc.

[https://www.w3schools.com/css/css\\_attribute\\_selectors.asp](https://www.w3schools.com/css/css_attribute_selectors.asp)



# Cascade

- Il Cascade "**cascata**" è un algoritmo che definisce come combinare valori di proprietà provenienti da fonti diverse.
- Stili:
  - **browser** - user-agent stylesheets
  - **author** - quello che scriviamo noi
  - **reader** - the user of the browser, may have a custom style sheet to tailor its experience.

<https://developer.mozilla.org/it/docs/Web/CSS/Cascade>

# Conflitti

- I **conflitti** di dichiarazione sono la normalità
  - Posso applicare più stili allo stesso elemento
  - Alcune proprietà le eredito
- Es: conflitto diretto fra due regole o proprietà

```
p{
  color: blue;
  margin-left: 30px;
}

p{
  color: green;
}
```

```
p{
  color: blue;
  color: green;
}
```

La seconda dichiarazione cancella la prima!!!

- Domina quello più vicino all'elemento
- Il p sarà verde.

ATTENZIONE  
Non è sempre così!!!!

# Esempio cascade 1

```
<!DOCTYPE html>
<html>
<head>
<style>

p{
  color: blue;
  margin-left: 30px;
}

p{
  color: green;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

- I paragrafi sono verdi
- Il margine rimane

## This is a Heading

This is a paragraph.

This is another paragraph.

```
element.style {
}

p {
  color: ■ green;
}

p {
  color: ■ blue;
  margin-left: 30px;
}
```

devtool

# Esempio cascade 2

```
<!DOCTYPE html>
<html>
<head>
<style>

p{
  color: blue;
  margin-left: 30px;
}

p{
  color: green;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p style="color: red;">This is another par

</body>
</html>
```

- Il secondo paragrafo è rosso!!
  - domina lo style inline

## This is a Heading

This is a paragraph.

This is another paragraph.

```
element.style {
  color: red;
}

p {
  color: green;
}

p {
  color: blue;
  margin-left: 30px;
}
```

devtool



# Esempio cascade 3

```
<!DOCTYPE html>
<html>
<head>
<style>

#para3{
  color: orange;
}

.red{
  color: red;
}

p{
  color: green;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p class="red">This is a paragraph with class.</p>
<p id="para3" class="red">This is a paragraph with id and class.</p>

</body>
</html>
```

**This is a Heading**

This is a paragraph.

This is a paragraph with class.

This is a paragraph with id and class.

Non tutti i selettori sono uguali, alcuni hanno più "peso" di altri

# Esempio 1

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
  color: red;
}
p span{
  color: green;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p>This is a <span>paragraph with a span</span> inside.</p>

</body>
</html>
```

## This is a Heading

This is a paragraph.

This is a paragraph with a span inside.

# Esempio 2

```
<!DOCTYPE html>
<html>
<head>
<style>
.news p{
  color: green;
}

.blog p{
  color: red;
}

</style>
</head>
<body>
<h1>This is a Heading</h1>
<section class="news">
  <p>This is a paragraph in sec news.</p>
  <p>This is a another paragraph in sec news.</p>
</section>
<section class="blog">
  <p>This is a paragraph in sec blog.</p>
  <p>This is a another paragraph in sec blog.</p>
</section>
</body>
</html>
```

## This is a Heading

This is a paragraph in sec news.

This is a another paragraph in sec news.

This is a paragraph in sec blog.

This is a another paragraph in sec blog.

# Esempio 3

```
<!DOCTYPE html>
<html>
<head>
<style>
#news p{
    color: green;
}

#blog p{
    color: red;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<section id="news">
    <p>This is a paragraph in sec news.</p>
    <p>This is a another paragraph in sec news.</p>
</section>
<section id="blog">
    <p>This is a paragraph in sec blog.</p>
    <p>This is a another paragraph in sec blog.</p>
</section>
</body>
</html>
```

## This is a Heading

This is a paragraph in sec news.

This is a another paragraph in sec news.

This is a paragraph in sec blog.

This is a another paragraph in sec blog.

# Esempio 4

```
<!DOCTYPE html>
<html>
<head>
<style>
section{
  color: grey;
}

.news .mark{
  font-weight: bold;
}

.blog .mark{
  color: orange;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<section class="news">
  <p>This is a paragraph in sec news.</p>
  <p class="mark">This is a another paragraph in sec news.</p>
</section>
<section class="blog">
  <p class="mark">This is a paragraph in sec blog.</p>
  <p>This is a another <span class="mark">paragraph</span> in sec blog.
</p>
</section>
</body>
</html>
```

## This is a Heading

This is a paragraph in sec news.

**This is a another paragraph in sec news.**

**This is a paragraph in sec blog.**

This is a another **paragraph** in sec blog.

# Esempio 5

```
<!DOCTYPE html>
<html>
<head>
<style>
.news.mark{
  font-weight: bold;
}

.news .mark{
  color: orange;
  font-weight: normal;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<section class="news mark">
  <p>This is a paragraph in sec news.</p>
  <p class="mark">This is a another paragraph in sec news.</p>
</section>
</section>
</body>
</html>
```

**This is a Heading**

**This is a paragraph in sec news.**

**This is a another paragraph in sec news.**



# Classi multiple

```
<!DOCTYPE html>
<html>
<head>
<style>
.btn {
  font-size: 18px; color: white; margin: 10px;
  padding: 10px; display: block; width: 100px;
}

.btn-danger {
  background: red;
}

.btn-success {
  background: green;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<a class="btn btn-danger" href="">Pericolo</a>
<a class="btn btn-success" href="">Hai vinto</a>

</body>
</html>
```

## This is a Heading

Pericolo

Hai vinto





# Calcolo della specificità

- Ad ogni dichiarazione è attribuita una specificità misurata con quattro valori [a,b,c,d]
  - ogni valore è indipendente dagli altri
  - "a" è il valore più importante e "d" il meno importante
  - si confrontano i valori più importanti e se uguali si passa a quelli successivi altrimenti termina
    - Es. 0,1,0,0 è più specifico di 0,0,5,5
- Calcolo dei valori
  - a. 1 se la dichiarazione è inline, 0 altrimenti
  - b. numero di selettori id
  - c. numero di selettori di classe, attributo o pseudo-classe
  - d. numero di selettori elemento o pseudo elemento

# Esempi

- Dallo standard **(6.4.3 Calculating a selector's specificity)**

<code>* {}</code>	<code>/* a=0 b=0 c=0 d=0 -&gt; specificity = 0,0,0,0 */</code>
<code>li {}</code>	<code>/* a=0 b=0 c=0 d=1 -&gt; specificity = 0,0,0,1 */</code>
<code>li::first-line {}</code>	<code>/* a=0 b=0 c=0 d=2 -&gt; specificity = 0,0,0,2 */</code>
<code>ul li {}</code>	<code>/* a=0 b=0 c=0 d=2 -&gt; specificity = 0,0,0,2 */</code>
<code>ul ol+li {}</code>	<code>/* a=0 b=0 c=0 d=3 -&gt; specificity = 0,0,0,3 */</code>
<code>h1 + *[rel=up]{}</code>	<code>/* a=0 b=0 c=1 d=1 -&gt; specificity = 0,0,1,1 */</code>
<code>ul ol li.red {}</code>	<code>/* a=0 b=0 c=1 d=3 -&gt; specificity = 0,0,1,3 */</code>
<code>li.red.level {}</code>	<code>/* a=0 b=0 c=2 d=1 -&gt; specificity = 0,0,2,1 */</code>
<code>#x34y {}</code>	<code>/* a=0 b=1 c=0 d=0 -&gt; specificity = 0,1,0,0 */</code>
<code>style=""</code>	<code>/* a=1 b=0 c=0 d=0 -&gt; specificity = 1,0,0,0 */</code>

# Esempi

- Dallo standard (6.4.3 Calculating a selector's specificity)

<code>* {}</code>	<code>/* a=0 b=0 c=0 d=0 -&gt; specificity = 0,0,0,0 */</code>
<code>li {}</code>	<code>/* a=0 b=0 c=0 d=1 -&gt; specificity = 0,0,0,1 */</code>
<code>html body div#pagewrap ul#summer-drinks li.favorite</code>	<code>/* a=0 b=0 c=0 d=4 -&gt; specificity = 0,0,0,4 */</code>
<code>ul li {}</code>	<code>/* a=0 b=0 c=0 d=2 -&gt; specificity = 0,0,0,2 */</code>
<code>ul ol+li {}</code>	<code>/* a=0 b=0 c=0 d=3 -&gt; specificity = 0,0,0,3 */</code>
<code>h1 + *[rel=up]{}</code>	<code>/* a=0 b=0 c=1 d=1 -&gt; specificity = 0,0,1,1 */</code>
<code>ul ol li.red {}</code>	<code>/* a=0 b=0 c=1 d=3 -&gt; specificity = 0,0,1,3 */</code>
<code>li.red.level {}</code>	<code>/* a=0 b=0 c=2 d=1 -&gt; specificity = 0,0,2,1 */</code>
<code>#x34y {}</code>	<code>/* a=0 b=1 c=0 d=0 -&gt; specificity = 0,1,0,0 */</code>
<code>style=""</code>	<code>/* a=1 b=0 c=0 d=0 -&gt; specificity = 1,0,0,0 */</code>

# !important

```
<!DOCTYPE html>
<html>
<head>
<style>

#para3{
  color: orange;
}

.red{
  color: red;
}

p{
  color: green !important;
}

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p class="red">This is a paragraph with class.</p>
<p id="para3" class="red">This is a paragraph with
id and class.</p>

</body>
</html>
```

- Se non voglio che una regola venga sovrascritta la posso dichiarare **!important**
- Viene usato in casi molto particolari
  - di solito è sempre possibile evitarlo

## This is a Heading

This is a paragraph.

This is a paragraph with class.

This is a paragraph with id and class.

# Cascading Order

	Origin	Importance
1	user agent	normal
2	user agent	!important
3	user	normal
4	author	normal
5	CSS Animations	<i>see below</i>
6	author	!important
7	user	!important

[https://developer.mozilla.org/it/docs/Web/CSS/Cascade#cascading\\_order](https://developer.mozilla.org/it/docs/Web/CSS/Cascade#cascading_order)