

XML, XHTML e CSS



Moreno Marzolla
Dipartimento di Informatica
Università Ca' Foscari di Venezia
marzolla@dsi.unive.it
<http://www.dsi.unive.it/~marzolla>

Testo di riferimento

- Erik T. Ray, *Learning XML*, O'Reilly, First Edition, January 2001 ISBN: 0-59600-046-4, 368 pages

Cos'è XML?

- Extensible Markup Language (XML) è una notazione per rappresentare informazioni in maniera strutturata e platform-independent
 - Standard aperto
 - Supporta il set di caratteri Unicode
 - Sintassi chiara, semplice e non ambigua
 - XML può essere combinato con stylesheet per produrre documenti con stili arbitrari
- XML non è un linguaggio di markup, ma una notazione per definire linguaggi di markup

Markup

- Col termine *markup* si intende qualsiasi informazione aggiunta ad un documento con lo scopo di migliorarne il significato
 - Il *markup* identifica parti del documento, e come queste sono in relazione con altre
- Esempio

```
<message>
  <exclamation>Hello, world!</exclamation>
  <paragraph>XML is <emphasis>fun</emphasis> and
  <emphasis>easy</emphasis> to use.
  <graphic fileref="smiley_face.pict"/></paragraph>
</message>
```

Concetti fondamentali / 1

- Boundaries
 - I tag <message> e </message> definiscono l'inizio e la fine di una collezione di testo e markup
- Roles
 - Quale è il ruolo di una regione di testo? Nell'esempio i tag <paragraph> e </paragraph> identificano del testo come paragrafo, anziché una lista, un titolo o altro
- Positions
 - Esiste un ordinamento (totale) tra frammenti di documento. Ad esempio, il paragrafo appare dopo il testo etichettato come <exclamation>, quindi verrà probabilmente stampato o visualizzato in quest'ordine

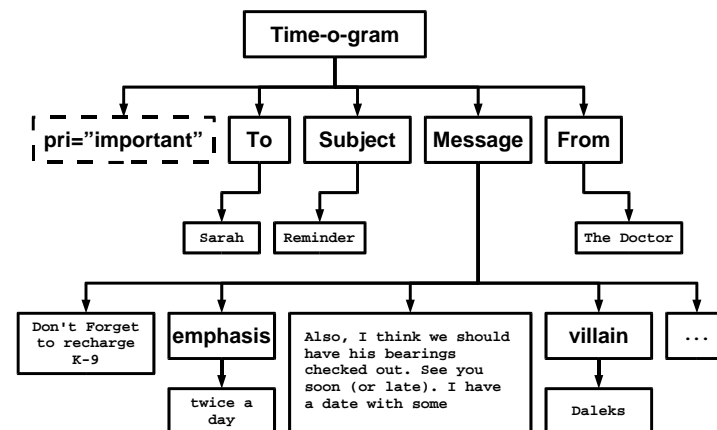
Concetti fondamentali / 2

- Containment
 - Il testo "fun" si trova in un elemento <emphasis>, il quale si trova dentro ad un <paragraph>, che è dentro a <message>. L'annidamento degli elementi può essere utilizzato dal software che processa il frammento XML, che può subire elaborazioni diverse in base a dove si trova
- Relationships
 - Un frammento di testo può essere collegato ad una risorsa che si trova in una locazione diversa. Ad esempio, il tag <graphic fileref="smiley_face.pict"/> potrebbe definire una relazione tra un frammento XML e il file chiamato smiley_face.pict.

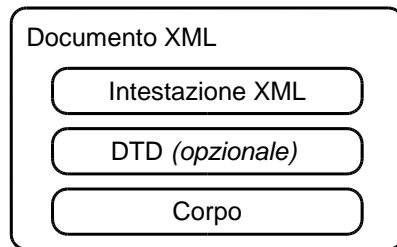
Anatomia di un documento XML

```
<?xml version="1.0"?>
<time-o-gram pri="important">
  <to>Sarah</to>
  <subject>Reminder</subject>
  <message>Don't forget to recharge K-9
  <emphasis>twice a day</emphasis>. Also, I think we
  should have his bearings checked out. See you soon
  (or late). I have a date with some
  <villain>Daleks</villain>...
  </message>
  <from>The Doctor</from>
</time-o-gram>
```

Diagramma ad albero di un documento XML



Sintassi generale



Dichiarazione XML

```
<?xml [ name1="val1" [ name2="val2" ... ] ] ?>
```

- Proprietà che è possibile settare:
 - **version**
 - Numero di versione (al momento "1.0")
 - **encoding**
 - Definisce il set di caratteri utilizzato nel documento, ad esempio "US-ASCII" o "iso-8859-1", "UTF-8" (set standard di caratteri latini)
 - **standalone**
 - Se standalone="yes" non ci sono file esterni da includere. Altrimenti, standalone="no"

- **Esempi:**

```
<?xml version="1.0" standalone="no" encoding="UTF-8"?>  
<?xml version="1.0"?>
```

Dichiarazione DTD

```
<!DOCTYPE root-element uri-of-dtd  
[  
  internal-subset  
]>
```

- **Esempio:**

```
<!DOCTYPE time-o-gram  
  PUBLIC "-//LordsOfTime//DTD TimeOgram 1.8//EN"  
  "http://www.lordsoftime.org/DTDs/timeogram.dtd"  
  [  
    <!ENTITY sj "Sarah Jane">  
    <!ENTITY me "Doctor Who">  
  ]>
```

Elementi XML

- **Elemento generico**

```
<name attr1="val1" attr2="val2" >  
  contenuto  
</name>
```

- **Elemento vuoto**

```
<name attr1="val1" attr2="val2" />
```

Entità interne

```
<?xml version="1.0"?>
<!DOCTYPE message
  SYSTEM "/xmlstuff/dtds/message.dtd"
 [
  <!ENTITY client "Mr. Rufus Xavier Sasperilla">
  <!ENTITY agent "Ms. Sally Tashuns">
  <!ENTITY phone "<number>617-555-1299</number>">
 ]>
<message>
  <opening>Dear &client;</opening>
  <body>
    We have an exciting opportunity for you! A set of
    ocean-front cliff dwellings in Pi&#241;ata, Mexico
    have been renovated as time-share vacation homes.
    They're going fast! To reserve a place for your
    holiday, call &agent; at &phone;. Hurry, &client;.
    Time is running out!
  </body>
</message>
```

Entità predefinite

- &#241; &
- ' '
- > >
- < <
- " “
- Numbered character entities
 - { indica il carattere di posizione decimale 123 nel set di caratteri Unicode; indica il carattere di posizione esadecimale 20 nel set di caratteri Unicode
- Named character entities
 - ` ´ ecc.

Entità esterne

```
<?xml version="1.0"?>
<!DOCTYPE doc
  SYSTEM "http://www.dtds-r-us.com/generic.dtd"
 [
  <!ENTITY part1 SYSTEM "p1.xml">
  <!ENTITY part2 SYSTEM "p2.xml">
  <!ENTITY part3 SYSTEM "p3.xml">
 ]>
<longdoc>
  &part1;
  &part2;
  &part3;
</longdoc>
```

```
<longdoc>
  &part1;
  &part2;
  &part3;
</longdoc>
```

Entità esterne

- La parola chiave SYSTEM è seguita da un URI
 - ```
<!ENTITY catalog SYSTEM "http://www.bobsbolts.com/catalog.xml">
```

    - Però la risorsa può essere spostata altrove, rendendo non valida l'entità
- Per risolvere il problema, è possibile utilizzare un identificatore PUBLIC
  - ```
<!ENTITY faraway PUBLIC "-//BOB//FILE Catalog//EN" "http://www.bobsbolts.com/catalog.xml">
```

 - Però il processore XML deve sapere come usare l'identificatore PUBLIC; per tale ragione è opportuno inserire anche un URI come fallback

Commenti e CDATA

- Commenti

```
<!-- testo del commento -->
```

- CDATA (Character Data)

- Utile per visualizzare blocchi di testo così come scritto, senza che il processore XML interpreti i caratteri speciali

```
<![ CDATA [ testo ]]>
```

- Esempio

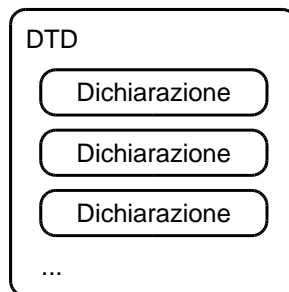
```
<para>&Egrave; possibile visualizzare frammenti di codice come <![CDATA[if (&x < &y)]]> in documenti XML.</para>
```

Documenti ben formati (well-formed documents)

- Sono i documenti XML che rispettano le regole sintattiche dei documenti XML
 - Ogni elemento non vuoto deve avere un tag di apertura e uno di chiusura
 - Ogni elemento vuoto deve avere una barra (/) prima della parentesi angolata di chiusura
 - Tutti i valori degli attributi vanno racchiusi tra virgolette (“)
 - I tag di apertura e chiusura devono essere bilanciati
 - Elementi di markup (<,]> ecc) non possono apparire isolati
 - I nomi di elementi iniziano con lettere o underscore, e contengono solo lettere, numeri, punto, underscore (_), hyphens (-)

DTD

- Un DTD definisce la “grammatica” che un documento XML deve rispettare
- Formato generale:



Element Declaration / 1

```
<!ELEMENT name content-model>
```

- Elementi vuoti

```
<!ELEMENT graphic EMPTY>
```

- Elementi senza restrizioni sul contenuto

```
<!ELEMENT contain-anything ALL>
```

- Elementi che contengono solo caratteri

```
<!ELEMENT emphasis (#PCDATA)>
```

Element Declaration / 2

- Elementi che contengono solo altri elementi

```
<!ELEMENT article (para+)>
<!ELEMENT article (title, (para | sect1)+)>
<!ELEMENT article (title, subtitle?,
                  ((para+, sect1*) | sect1+))>
```

- Elementi con contenuto misto

```
<!ELEMENT para (#PCDATA | emphasis | xref)*>
```

Simboli usati

- , (virgola)
 - Indica una sequenza obbligatoria di elementi
- | (barra)
 - Indica una alternativa
- (content)
 - Raggruppa il contenuto in modo tale che l'operatore che segue si applichi al tutto
- ?
 - Rende l'elemento che precede opzionale
- +
 - Richiede una o più istanze dell'elemento che precede
- *
 - Richiede zero o più istanze dell'elemento che precede

Dichiarazione di liste di attributi

```
<!ATTLIST name
  attname1 atttype1 attdsc1
  attname2 atttype2 attdsc2
>
```

- Specifica:
 - Nome del tag cui si applica l'attributo
 - Nome dell'attributo
 - Tipo dell'attributo
 - Comportamento dell'attributo (obbligatorio/opzionale, valore di default ecc.)

```
<!ATTLIST memo
  id ID #REQUIRED
  security (high | low) "high"
  keywords NMTOKENS #IMPLIED
>
```

Attribute Datatypes / 1

- CDATA
 - Character data: rappresenta una generica sequenza di caratteri

```
<!ATTLIST circle radius CDATA "12 inches">
```

- Esempi

```
dimensions="35x12x9 mm"
company="O'Reilly & Associates"
text=" 5 + 7 = 3 * 4 "
```

Attribute Datatypes / 2

- NMTOKEN

- Name Token: Stringa di caratteri che inizia con una lettera e può contenere numeri, lettere e certi simboli di punteggiatura

```
<!ATTLIST part number NMTOKEN #REQUIRED>
```

- Esempi

```
skin="reptilian"  
file="README.txt"  
version="v3.4-b"
```

Attribute Datatypes / 3

- NMTOKENS

- Named Tokens List: lista di token separati da spazi

```
<!ATTLIST article keywords NMTOKENS #IMPLIED>
```

- Esempi

```
name="Greg Travis"  
format="thin blue border"
```

Attribute Datatypes / 4

- ID

- Identificatore univoco: non possono esistere elementi diversi nel documento XML con lo stesso ID

```
<!ATTLIST record num ID #REQUIRED>
```

Attribute Datatypes / 5

- IDREF

- Riferimento ad un ID

```
<!ATTLIST relatedword ref IDREF #REQUIRED>
```

- IDREFS

- Lista di riferimenti ad IDs

```
<!ATTLIST bookset refs IDREFS #REQUIRED>
```

Attribute Datatypes / 6

- ENTITY

- Questo tipo accetta un nome di entità come argomento

```
<!ATTLIST bulletlist icon ENTITY #IMPLIED>  
<!ENTITY bluedot SYSTEM "icons/bluedot.png">
```

- Esempio

```
<bulletlist icon="bluedot">
```

- ENTITIES

- Stessa cosa, con una lista di entità

Attribute Datatypes / 7

- Enumerate Value List

```
<!ATTLIST part instock ( true | false ) #IMPLIED>  
<!ATTLIST schedule day  
  ( mon | tue | wed | thu | fri | sat | sun )  
  #REQUIRED >  
<!ATTLIST shape type  
  ( circle | square | triangle ) "square">
```

- Esempio

```
<schedule day="fri">...</schedule>
```

Attribute Behavior

- Valore di default

- Se l'utente non specifica un valore, viene messo quello di default, che deve essere specificato nella dichiarazione dell'attributo

- Attributo opzionale (**#IMPLIED**)

- Non c'è un valore di default, se l'utente non indica l'attributo, questo viene ritenuto assente

- Attributo obbligatorio (**#REQUIRED**)

- L'attributo non può essere omesso, e non c'è alcun valore di default predefinito.

- Attributo con valore fisso non modificabile (**#FIXED**)

Tipi di entità / 1

- General Entity

- `<!ENTITY abc "The ABC Group">`
- Richiamato come `&abc;`

- External General Entity

- `<!ENTITY man SYSTEM "/pub/docs/manuals/prod23.htm">`
- Richiamato come `&man;`

- Nonparsed External Entity

- `<!ENTITY logo SYSTEM "images/logo.gif" NDATA gif>`
- Richiamato come `%logo;`

Tipi di entità / 2

- Parameter Entity

- `<!ENTITY % paratext "(#PCDATA | emph | acronym)*">`
- Richiamato come `%paratext`;
- Individua un frammento di DTD

- External Parameter Entity

- `<!ENTITY % tables SYSTEM "http://www.xmljunk.org/dtds/tables2.1.dtd">`
- Richiamato come `%tables`;
- Individua un frammento esterno di DTD

Esempio Parameter Entities

```
<!ENTITY % content "para | note | warning">
<!ENTITY % id.att "id ID #REQUIRED">
<!ELEMENT chapter (title, epigraph, (%content;)+)>
<!ATTLIST chapter %id.att;>
<!ELEMENT appendix (title, (%content;)+)>
<!ATTLIST appendix %id.att;>
```

Equivale a

```
<!ELEMENT chapter
  (title, epigraph, (para | node | warning)+)>
<!ATTLIST chapter id ID #REQUIRED>
<!ELEMENT appendix
  (title, (para | node | warning)+)>
<!ATTLIST appendix id ID #REQUIRED>
```

Esempio Versione super-semplificata di XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html SYSTEM "html.dtd">
<html>
  <h1>Esempio</h1>

  <p>Questo documento rappresenta un esempio
  molto semplice di file <u>XHTML</u>.</p>

  <p><b>Attenzione:</b> oltre a cambiare la
  formattazione dei caratteri, dobbiamo dare
  la possibilit&egrave; di inserire
  <a href="home.html">link</a></p>

  <p>Di tutti gli elementi &egrave; possibile cambiare
  lo <span style="color: blue">stile</span></p>
</html>
```

Esempio Versione super-semplificata di XHTML

```
<!ELEMENT html ( h1 | p ) * >
<!ENTITY % testo "#PCDATA">
<!ELEMENT h1 ( %testo; | a | span ) * >
<!ELEMENT p ( %testo; | a | b | u | span ) * >
<!ELEMENT a ( %testo; ) >
<!ELEMENT u ( %testo; ) >
<!ELEMENT b ( %testo; ) >
<!ATTLIST a href CDATA #REQUIRED>
<!ELEMENT span ( %testo; ) >
<!ATTLIST span style CDATA #IMPLIED>
<!ENTITY egrave "&#232;" >
<!ENTITY agrave "&#224;" >
```

Documenti XML Validi

- Documenti XML Ben Formati (well-formed)
 - Rispettano la sintassi di XML
- Documenti XML Validi
 - Sono Ben Formati
 - In più, rispettano la struttura sintattica descritta dal proprio DTD
- Documenti Ben Formati possono *non* essere Validi

XHTML

- Versione di HTML in notazione XML
- Sostanzialmente simile a HTML 4.01, con alcune differenze importanti
 - I documenti XHTML devono iniziare con l'intestazione standard `<?xml version="1.0" ?>`
 - Tutti i tag aperti devono essere chiusi
 - Tutti i valori degli attributi vanno racchiusi tra virgolette
 - I documenti XHTML devono importare un DTD esterno

XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en">
  <head>
    <title>Virtual Library</title>
  </head>
  <body>
    <p>This is a test page.</p>
  </body>
</html>
```

XHTML: Livelli di conformità

- *Strict XHTML*
 - Presenta notevoli differenze con HTML, e rimuove molti elementi HTML utilizzati ai soli scopi di presentazione. E' necessario usare CSS per dare alle pagine l'aspetto voluto
- *Transitional XHTML*
 - Per coloro che desiderano che le proprie pagine siano compatibili con browser datati che non supportano stylesheet
- *Frameset XHTML*
 - Come strict XHTML, con in più la possibilità di utilizzare i frames

XHTML: Livelli di conformità

- Strict

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Transitional

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- Frameset

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Differenze principali tra XHTML e HTML / 1

- I documenti XHTML devono essere ben formati
- I nomi dei tag e degli attributi devono essere in minuscolo
- Per *tutti* gli elementi è indispensabile indicare il tag di chiusura
 - Anche per quelli vuoti: `
` al posto di `
`
- I valori degli attributi devono sempre essere racchiusi tra virgolette
 - `<table columns="1">` anziché `<table columns=1>`

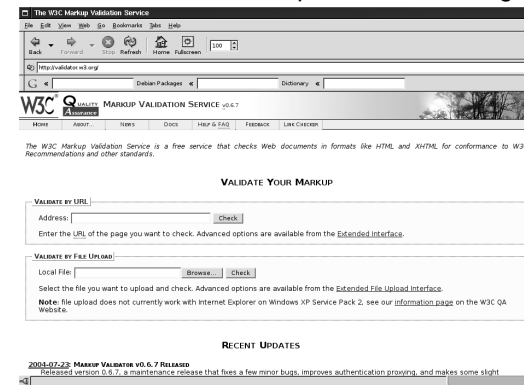
Differenze principali tra XHTML e HTML / 2

- Attenzione ad includere gli script: i caratteri `<` e `&` hanno un significato speciale in XML

```
<script type="text/javascript">
<![CDATA[
... unescaped script content ...
]]>
</script>
```

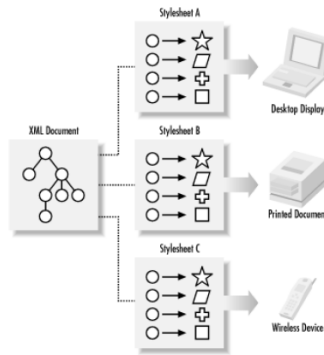
Come faccio a sapere se la mia pagina è XHTML valido?

- W3C Validation Service: <http://validator.w3.org>



CSS

- Cos'è uno stylesheet?
- Pagine XHTML e CSS sono complementari
 - XHTML definisce il contenuto di un documento
 - CSS ne definisce l'aspetto, o la presentazione
 - Lo stesso documento XHTML può essere presentato in modi diversi, in base allo stile applicato



CSS e HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en">
  <head>
    <title>Moreno Marzolla Teaching Activities</title>
    <link rel="stylesheet" type="text/css"
      href="css-main.css" media="screen"/>
    <link rel="stylesheet" type="text/css"
      href="css-print.css" media="print"/>
  </head>
  <body>
    ....
  </body>
</html>
```

CSS e HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en">
  <head>
    <title>Moreno Marzolla Teaching Activities</title>
    <style type="text/css">
      ...
    </style>
  </head>
  <body>
    ....
  </body>
</html>
```

Tipi di media / 1

- screen
 - Intended for non-paged computer screens.
- tty
 - dispositivi con font a spaziatura fissa
- tv
 - dispositivi con bassa risoluzione, basso numero di colori, limitata possibilità di scroll (es. televisori)
- projection
 - videoproiettori

Tipi di media / 2

- handheld
 - dispositivi portatili (schermo piccolo, spesso b/n, grafica bitmap, banda limitata)
- print
 - stampanti
- braille
 - dispositivi con feedback braille
- aural
 - sintetizzatori vocali
- all
 - tutti i dispositivi

Sintassi generale dei CSS

- I CSS si compongono di una serie di *regole*
 - Per ogni elemento HTML si controlla se ad esso si applica una regola
 - In caso affermativo, a quell'elemento (ed eventualmente al suo contenuto) si applicano le formattazioni indicate nel corpo della regola

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

Esempio

```
p {  
  font-weight: bold; /* carattere grassetto */  
  color: #0000ff; /* colore bianco */  
  margin-left: 2em; /* margine sinistro */  
}
```

- Tutti i paragrafi (tag 'p') vengono visualizzati in grassetto, testo in bianco e margine sinistro pari alla larghezza di due lettere 'm'

Esempio

```
h1 {  
  margin: 10pt 0pt 10pt 0pt;  
  display: block;  
  padding: 6pt;  
  background-color: #00ffff; /* aqua */  
  color: #000000;  
  text-align: center;  
  border: thin solid #00bec1;  
}
```

Corso di Tecnologie Web @ UniFE 2004/2005

Selettori / 1

- **h1 { ... }**
 - Tutti i tag 'h1'
- **h1, h2, h3 { ... }**
 - Tutti i tag 'h1', oppure 'h2', oppure 'h3'
- **p.citazione**
 - Tutti i tag 'p' con attributo 'class' settato a "citazione"
 - `<p class="citazione">blah blah blah</p>`
- **.citazione**
 - Tutti i tag con attributo 'class' settato a "citazione"
 - `<div class="citazione">...</div>`
 - `<p class="citazione">...</p>`

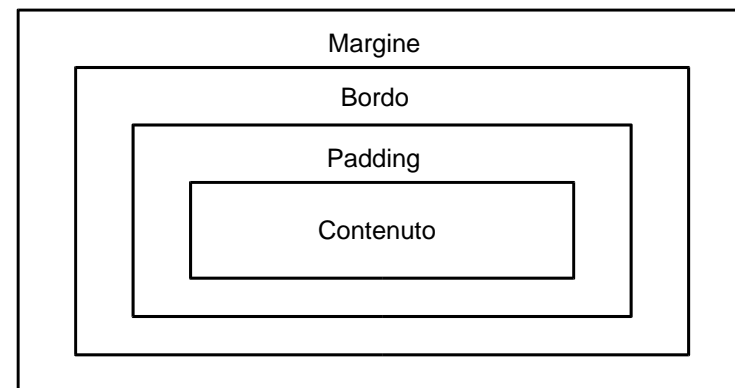
Selettori / 2

- **#navbar**
 - Tutti i tag con attributi 'id' settato a "navbar"
 - `<div id="navbar">...</div>`
- **p#autore**
 - Tutti i tag 'p' con attributo 'id' settato a "autore"
 - `<p id="autore">Moreno Marzolla</p>`
- **div p { ... }**
 - Tutti i tag 'p' che sono discendenti di un tag 'div'
 - `<div>...<p>blah</p>...</div>`

Selettori / 3

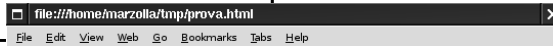
- **.navbar a { ... }**
 - Tutti i tag 'a' che sono discendenti di un tag qualsiasi con attributo 'class' settato a "navbar"
 - `<div class="navbar">.........</div>`
- **a[href] { ... }**
 - Tutti i tag 'a' che hanno un attributo "href" settato ad un qualche valore
- **Direttamente all'interno dei tag, utilizzando l'attributo "style"**
 - `<p style="color: #ffffff">blah</p>`

Modello di formattazione



Esempio

```
p.quote {
  margin: 2ex 6ex 2ex 6ex;
  border: 2pt solid black;
  padding: 1ex;
  font-style: italic;
  background-color: #00ffff;
  text-align: right;
}
```



All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy. All work and no play makes Jack a dull boy.

*In theory, theory equals practice
In practice, they differ*

Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca. Il mattino ha l'oro in bocca.

Unità di misura

- Assolute

```
h1 { margin: 0.5in } /* inches, 1in = 2.54cm */
h2 { line-height: 3cm } /* centimeters */
h3 { word-spacing: 4mm } /* millimeters */
h4 { font-size: 12pt } /* points, 1pt = 1/72 in */
h4 { font-size: 1pc } /* picas, 1pc = 12pt */
h4 { font-size: 12px } /* 12 pixel */
```

Unità di misura

- Relative

```
h1 { margin: 0.5em }
/* ems, the height of the element's font */
h1 { margin: 1ex }
/* x-height, ~ the height of the letter 'x' */
h1 { font-size: 120% }
/* 120% la dimensione del font standard */
```

- Usare le unità di misura relative, quando possibile

- In questo modo, le misure verranno automaticamente scalate se la finestra viene ridimensionata.

Colori

```
EM { color: #f00 } /* #rgb */
EM { color: #ff0000 } /* #rrggbb */
EM { color: rgb(255,0,0) } /* integer range 0 - 255 */
EM { color: rgb(100%, 0%, 0%) } /* float range 0.0% - 100.0% */
```