
LABORATORIO APPLICAZIONE SW E SICUREZZA INFORMATICA

R



ORGANIZZAZIONE DEL CORSO

Docenti: Roberto Beraldì, Marco Console

Orario del corso:

Martedì 16:00-19:00, Via Tiburtina 205, Esercitazione in laboratorio

Venerdì 17:00-19:00, Aula 204, Edificio Marco Polo Lezione

Ricevimento: mercoledì mattina ore 12. mandare email per prenotazione/conferma

beraldi@diag.uniroma1.it

STRUMENTI WEB DEL CORSO

- Classroom
 - Streaming zoom sincrono del corso
1. Slides ed altro materiale distribuito classroom
 2. Libro di testo : A. Fox, D. Patterson: Engineering Software as a Service: An Agile Approach Using Cloud Computing. ISBN-13: 978-0984881246

MODALITÀ ESAME

- Si tratta di ottenere l'idoneità
- Progetto in Gruppo (2-3 persone)
- Prima fase: invio della descrizione del Progetto mediante google form
 - User Stories
 - Mockup
 - Test da effettuare
- I docenti controllano la descrizione, se deve essere rivista inviano una email, altrimenti si intende 'approvato'
- Seconda Fase:
 - Esame

ESAME

- Il gruppo dovrà dimostrare (**mediante una demo**) che il **progetto realizzato funzioni correttamente**
- Durante la discussione del progetto, il docente verifica la conoscenza generale dei contenuti del corso attraverso domande, ispirate (ma non limitate) al progetto in discussione
- Se si decide di scrivere la relazione finale per la laurea (cfr. modalità della prova finale sul sito del CAD), bisogna produrre in quella sede una relazione tecnica sul progetto secondo lo schema che verrà discusso nel corso

PROGRAM OUTLINE

- Software as a Service
- Ruby
- Rails (basic)
- Rails (advanced)
- Security

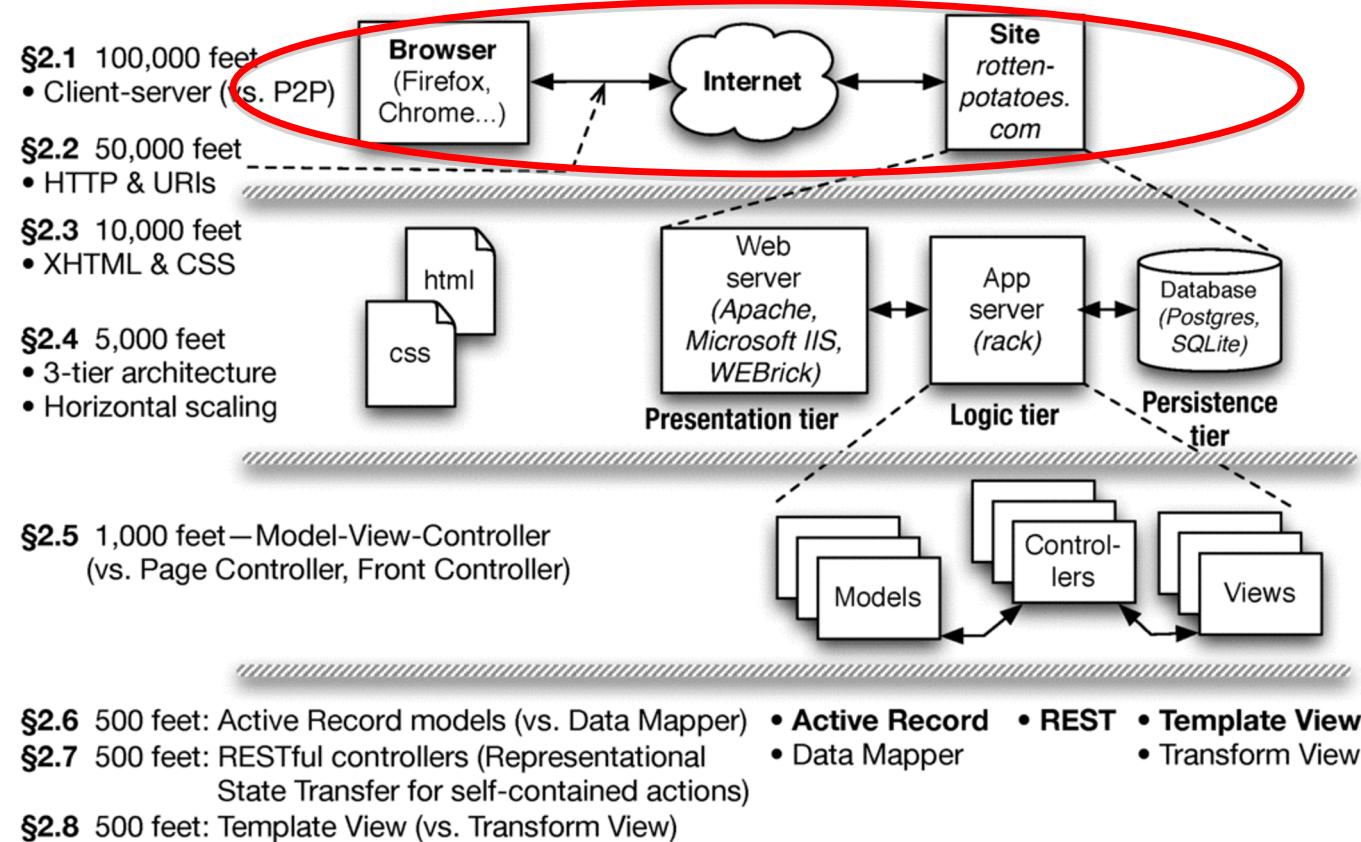


IL WEB COME SISTEMA CLIENT-SERVER

LASSI-I

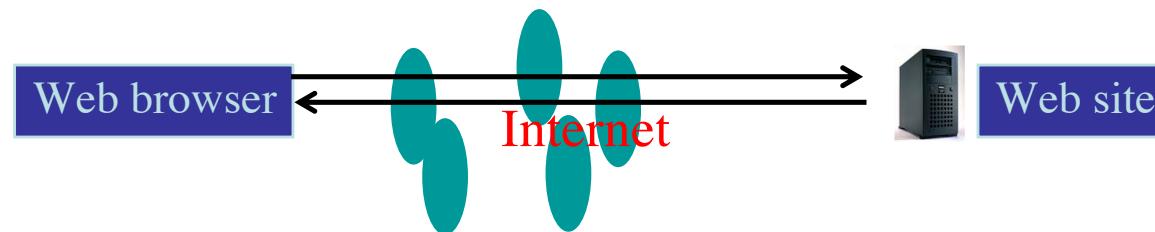


INTERAZIONE C/S

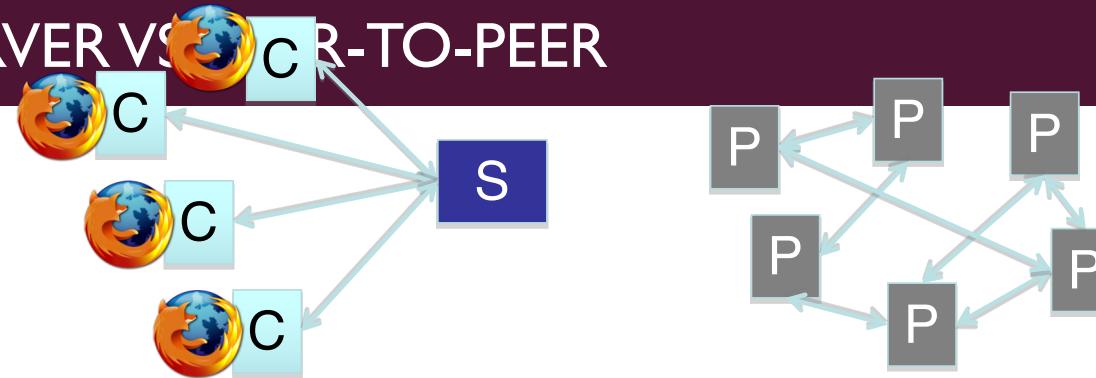


IL WEB DALL'ALTO

- Il Web è un'architettura client/server
- È fondamentalmente orientato ad un modello di interazione del tipo “richiesta/risposta” (request/reply)



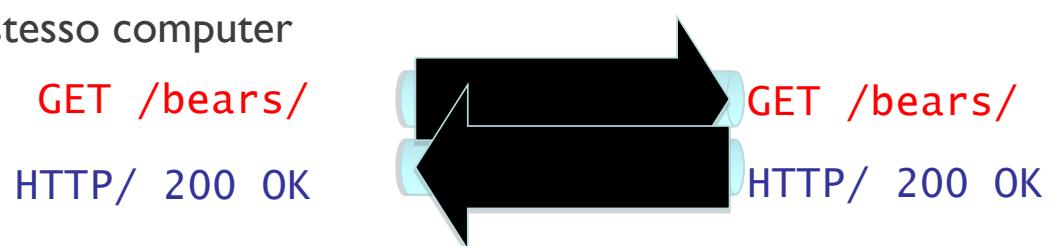
CLIENT-SERVER VS C2C-TO-PEER



- Architettura di alto livello del sistema complessivo
- Presto parleremo di architettura "dentro" scatole Client & server ciascuno specializzato per le proprie attività
- Cliente: porre domande per conto degli utenti
- Server: attendere e rispondere alle domande, servire molti clienti I modelli di progettazione catturano soluzioni strutturali comuni a problemi ricorrenti
- Client-Server è un modello architetturale

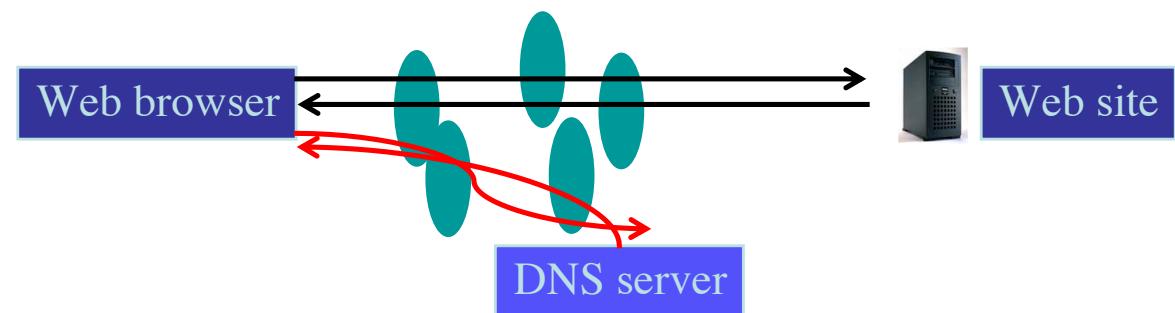
TCP/IP PROTOCOLS

- L'indirizzo IP (Internet Protocol) identifica un'interfaccia di rete fisica con quattro ottetti, ad esempio 128.32.244.172
- L'indirizzo speciale 127.0.0.1 è "questo computer", denominato localhost, anche se non connesso a Internet!
- TCP/IP (Transmission Control Protocol/Internet Protocol)
- IP: servizio senza garanzia e con il massimo sforzo che fornisce pacchetti da un indirizzo IP all'altro
- TCP: rendere l'IP affidabile rilevando pacchetti "eliminati", dati che arrivano fuori servizio, errori di trasmissione, reti lente, ecc., e rispondere in modo appropriato
- Le porte TCP consentono più app TCP nello stesso computer



DNS

- Dns (Domain Name System) è un altro tipo di server che esegue il mapping dei nomi agli indirizzi IP



HYPertext Transfer Protocol (HTTP)

- Un protocollo di richiesta/risposta basato su ASCII per il trasferimento di informazioni sul Web
- La richiesta HTTP include:
- metodo di richiesta (GET, POST, ecc.)
- URI (Uniform Resource Identifier)
- Versione del protocollo HTTP compresa dal client
- informazioni aggiuntive sulla richiesta di trasferimento

HYPertext Transfer Protocol

- Risposta HTTP dal server
- Versione protocollo
- Codice stato
- Corpo di risposta => PAGINA WEB

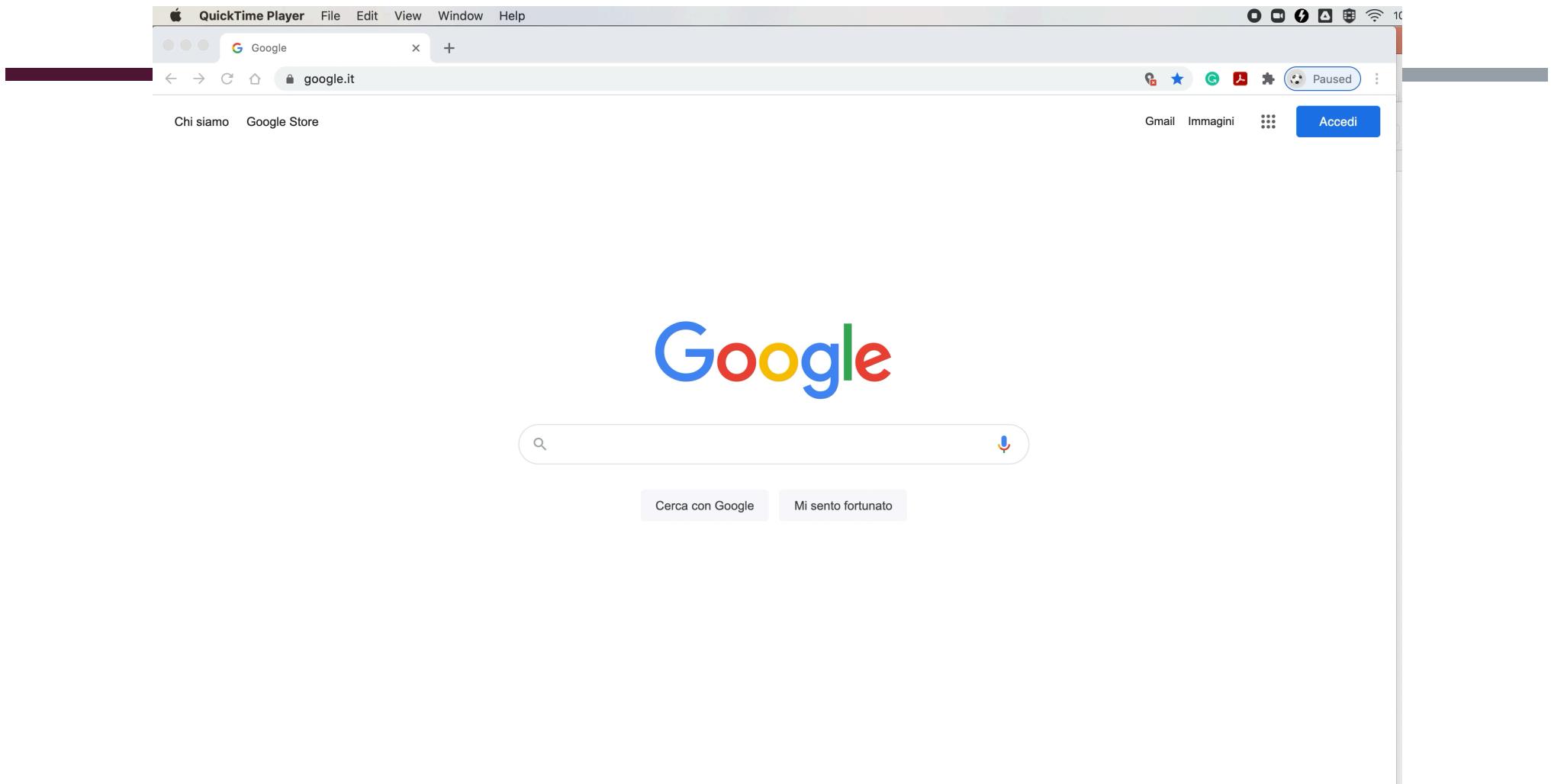
HTTP status codes:

2xx — *all is well*

3xx — *resource moved*

4xx — *access problem*

5xx — *server error*

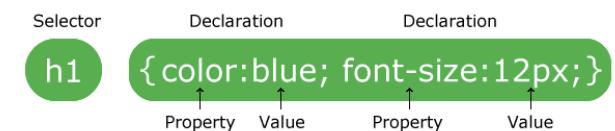


COOKIES

- Osservazione: HTTP è senza stato
- Problema iniziale del Web 1.0: come guidare un utente "attraverso" un flusso di pagine?
- utilizzare l'indirizzo IP per identificare l'utente di ritorno?
- ✗ computer pubblici, utenti che condividono un singolo IP
- incorporare una id differente per utente nella stringa di query URI?
- ✗ la memorizzazione nella cache non condivisibile
- Rapidamente sostituito dai cookie
- Rails gestisce i cookie antimanomissione

CHE C'È DENTRO UNA PAGINA WEB?

- Contenuto + struttura = elemento html (HTML5) definito tramite un set fisso di tag
- Stile (CSS3):
 - Come visualizzare un elemento (colori, tipi di carattere, bordi...)
 - Come impaginare (disporre) degli elementi nell'area di rendering
- Come reagire a un 'evento': JavaScript (JS)
 - Modifica dello stile di un elemento (ad esempio, onMouseOver) Attivare alcuni calcoli Modificare il contenuto della pagina



HTML ESSENTIAL

- Libro:
■ **HTML5: Pocket Reference (Oreilly)**

- Tutorial:
■ <https://www.w3schools.com/html/>

- Lista dei tag HTML:
■ <https://www.w3schools.com/tags/default.asp>
■ <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

COME CREARE UNA PAGINA HTML?

- Visual Studio Code

ESEMPIO DI PAGINA HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- TAG = “marcatura” (o marcatore)
- Un tag viene usato per marcare una parte di testo
- Tag indicano:
 - formattazione (per cambiare l'aspetto ad una parte di testo) (es.
 - “semantica”
 - per dare un “significato” ad una parte di testo

ELEMENTI HTML

- L'elemento HTML è tutto ciò che va dal tag iniziale al tag di fine:
- `<tagname>Il contenuto va qui...</tagname>`
- ESEMPI:
- `<h1>My First Heading</h1>`
- `<p>My first paragraph.</p>`



DISPLAY ELEMENTI

- Rispetto al modo in cui sono visualizzati gli elementi sono classificati in due modi, blocco e livello
- Un elemento a **livello di blocco** inizia sempre su una nuova linea e occupa l'intera larghezza disponibile (si estende a sinistra e a destra il più lontano possibile).
- Un **elemento in linea** non inizia su una nuova riga e occupa solo la larghezza necessaria.

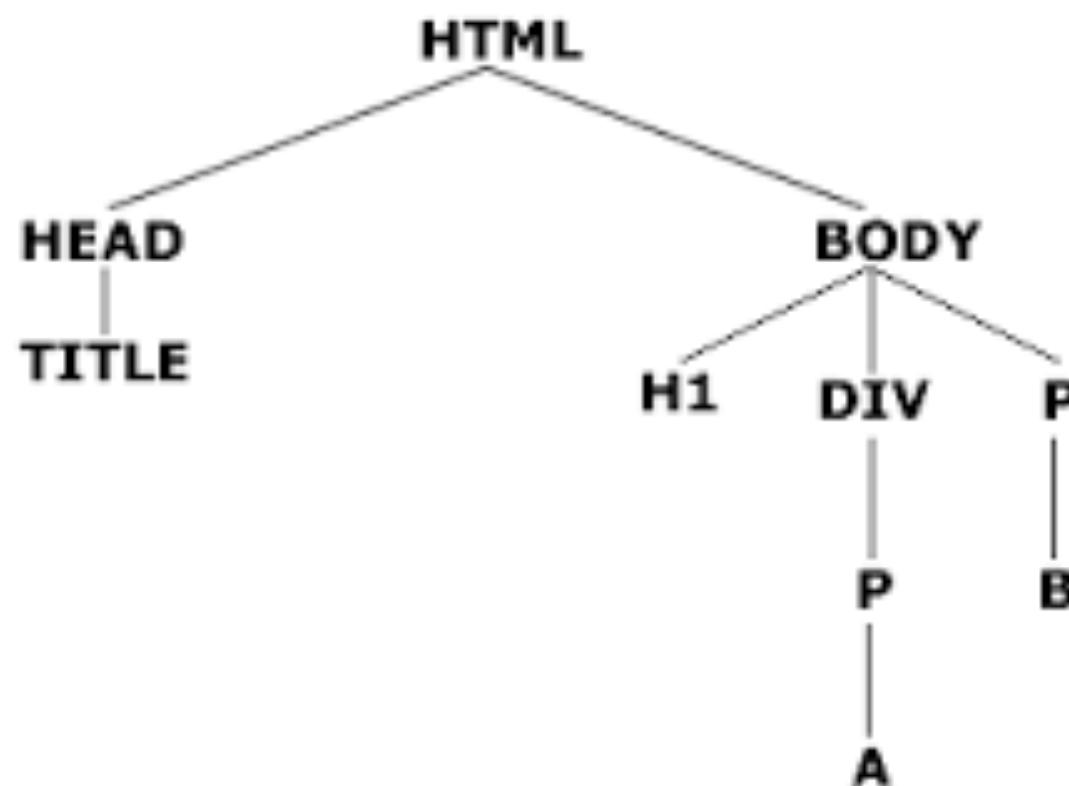
ATTRIBUTI DEI TAG

- Ogni occorrenza di tag (di apertura) può contenere assegnazioni di attributi
- Ogni tag ha un diverso insieme di possibili attributi
- Assegnazione valore ad attributo: → nome-attributo = valore
- Esempio: `Visit W3Schools`
- Struttura del tag con attributi:

- `<nome-tag attributo1 = valore1 attributo2 = valore2>`

- ``

STRUTTURA DOCUMENTO HTML



ELEMENTI PRINCIPALI DELL'INTESTAZIONE

- DOCTYPE
- HTML
- HEAD
- TITLE
- META

ESEMPIO DI INTESTAZIONE

- <!DOCTYPE html>
- <html>
- <head>
- <meta charset="utf-8">
- <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
- <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
- <title>Home | Dipartimento di Ingegneria informatica, automatica e gestionale</title>
- <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700,800">

ELEMENTO BODY E PRINCIPALI ATTRIBUTI

- È il **CONTENITORE** di altri elementi
- **Background.** Permette di inserire una immagine come sfondo del documento html.
- **Text.** Questo attributo permette di cambiare il colore del testo del documento html.
- **Link.** È l'attributo che consente di cambiare il colore dei link del documento.
- **Alink.** Quando si clicca su un link quest'ultimo cambia colore fin quando l'utente non alza il dito dal pulsante del mouse.
 - L'attributo Alink consente di personalizzare questo colore
- **Vlink.** È l'attributo che modifica il colore dei link del documento già cliccati da parte dell'utente.



HEADINGS

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

ELEMENTO PARAGRAPH

- Un paragrafo inizia sempre su una nuova riga e i browser aggiungono automaticamente uno spazio vuoto (un margine) prima e dopo un paragrafo.
- Gli eventuali “andate a capo” nel contenuto del paragrafo sono ignorate
- `<p>This is a paragraph.</p>`
`<p>This is another paragraph.</p>`

(ALCUNI) ELEMENTI DI FORMATTAZIONE

- <pre> preformatted
- - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- ..
- <sub> - Subscript text
- <sup> - Superscript text

ELEMENTO DIV

- L'elemento HTML content DIVision (<div>) è il contenitore generic
- Non ha alcun effetto sul contenuto o sul layout fino a quando non viene applicato in qualche modo utilizzando CSS (vedi dopo)
 - ad esempio, lo stile viene applicato direttamente ad esso o un qualche tipo di modello di layout come Flexbox viene applicato al suo elemento padre

ELEMENTO LISTA

-
- Milk
- Cheese
-
- Blue cheese
- Feta
-
-
-

ANCORE <A>

- 2 tipi di tag :
- con attributo **HREF**: – aggiungono un link ipertestuale (esterno o interno al documento)
- con attributo **NAME**: – definiscono uno specifico punto del documento come un link interno –
- tale punto può essere direttamente acceduto attraverso un tag
- altri attributi: TARGET, TITLE

ESEMPIO I

- ` zona 1 del documento raggiungibile direttamente `
- `....`
- `torna alla zona 1 del documento`

- ` vai alla zona 1 del documento index.html del sito diag.uniroma1.it `

ESEMPIO 2

 **HTML Demo: <a>** Reset

HTML

CSS

```
1 <p>You can reach Michael at:</p>
2
3 <ul>
4   <li><a href="https://example.com">Website</a></li>
5   <li><a href="mailto:m.bluth@example.com">Email</a></li>
6   <li><a href="tel:+123456789">Phone</a></li>
7 </ul>
8
```

Output

You can reach Michael at:

- [Website](https://example.com)
- [Email](mailto:m.bluth@example.com)
- [Phone](tel:+123456789)

ELMENTO TABLE



HTML Demo: <table>

Reset

HTML CSS

```
1 <table>
2   <thead>
3     <tr>
4       <th colspan="2">The table header</th>
5     </tr>
6   </thead>
7   <tbody>
8     <tr>
9       <td>The table body</td>
10      <td>with two columns</td>
11    </tr>
12  </tbody>
13 </table>
```

Output

The table header

The table body | with two columns

LABI

- Visual Studio Code
- HTML Snippets extension
- Salvare il file come primo.html
- Usare snippet **html:5**



CASCADING STYLE SHEET

- La specifica CSS definisce i modi con cui scrivere *regole di stile* ed applicarle ad una pagina, determinandone così la visualizzazione.
- Una **regola di stile** è un insieme di definizioni (**comandi di stile**) che, assegnata ad uno o più elementi, determina le caratteristiche di visualizzazione complessiva di quegli elementi.
`<body style="background-color: yellow; color: blue">`
- Un comando di stile può essere applicato direttamente all'elemento (**stili inline**).
`<h1 style="text-align: center; color: purple; position: absolute; font-family: Arial, sans serif">stili inline </h1>`
- Un comando di stile può essere applicato all'interno di un elemento (**stili interni**).
`<p style="color: black; font-size: 120%; text-align: right">
The End.
</p>`

Comandi
separati da ;

Il documento sarà visualizzato con testo blu su fondo giallo

Header centrato, testo viola di font arial (o, in mancanza, una qualunque sans-serif disponibile)

Questo elemento p modifica il colore testo da blue (ereditato dall'elemento di cui è figlio: body) a nero; il testo sarà più grande del valore normale (ereditato) e sarà allineato a destra

inlineStylesheet.html

```
<body style="background-color: yellow; color: blue">
<h1>stili inline</h1>anche h1 e' block-level
<h1 style="text-align: center; color: purple; font-family: Arial, sans serif">stili inline ... </h1>

<table border="1" cellspacing="3" cellpadding="5">
<thead style="background-color: green">
<tr><td></td>
<th>item</th>
<th>costo</th> </tr> </thead>
<tbody>
<tr><td rowspan="2" style="vertical-align: top">
verdura</td>
<td>zucchine</td>
<td>€2,50</td> </tr>
<tr>
<td>asparagi</td>
<td>€3,05</td> </tr>
<tr>
<th>frutta</th>
<td>fragole</td>
<td>€6,20</td>
</tr> </tbody> </table>

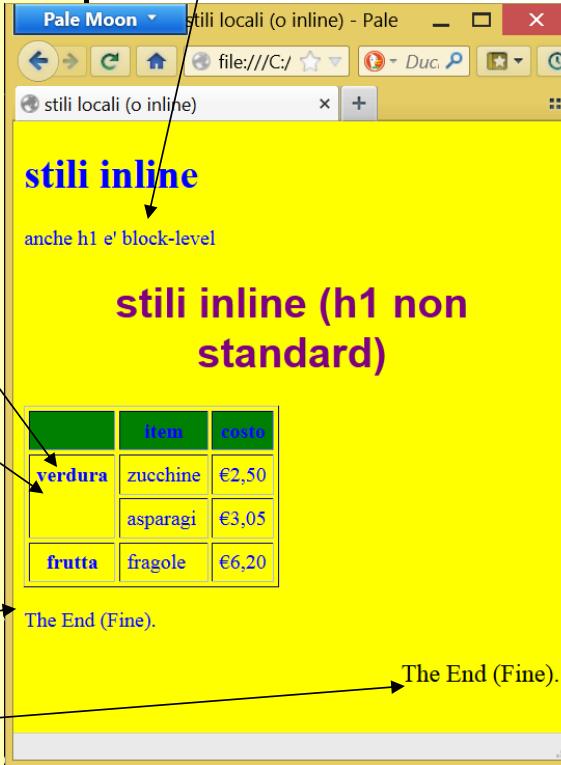
<p>The End (Fine)</p>
<p style="color: black; font-size: 120%; text-align: right">The End...</p>
```

h1 normale (eredita le impostazioni di body)

!testo fuori markup ...

Il blue del body resta

Un p normale e uno che modifica alcune proprietà



LAB2

≡ Preview X

□ ...

Periodo	Filosofo	Filosofia	Storia coeva
625 - 545 ca.	Talete di Mileto	Predisse un' eclissi Acqua come archè (origine di tutto)	c.a. 624 leggi di Dracone 594 Riforma di Solone

REGOLE DI STILE CSS

- **Internal stylesheet:** insieme di regole definite nella **<head>** del documento, con il tag **<style>**;
- **External stylesheet:** insieme di regole definite in **file a parte**, che poi viene riferito con un elemento **<link>** del documento;
- Per definire una regola si associa
 - - un **selettore** (elemento o “tipo di elemento” cui applicare la regola)
 - - a un **insieme di comandi di stile** racchiuso tra **{...}** (la regola:).
- **selettore {property: value; ...; property: value}**

REGOLE DI STILE CSS

Il documento che usa **queste regole** ha

- <body> formattato come indicato qui
- **ogni** elemento <h1> formattato come
- **ogni** elemento <p> formattato come

... se **non** vogliamo assegnare un certo stile a **tutte** le occorrenze di un certo elemento, dobbiamo usare selettori piu' particolareggiati.

```
body {background: yellow;
      color: blue
}

h1 {text-align: center;
    color: purple;
    font-family: Arial,sans serif;
}

p {color:black;
   font-size: 120%;
   text-align: right;
}
```

external.css

SELETTORI CSS

con

le regole {...} si applicano a

`h1 {...}`

tutte le occorrenze dell'elemento **h1**

`h1, h2 {...}`

tutte le occorrenze di **h1** e tutte quelle di **h2**

`h1 em {...}`

tutte le occorrenze dell'elemento **em** dentro ad un **h1**

`h1.special {...}`

solo ad una **classe** di elementi h1:

cioe' alle occorrenze dell'elemento `<h1 class="special">...</h1>`

`.centrato {...}`

le occorrenze di qualsiasi elemento con class="centrato"

esempio: `<p class="centrato">...</p>`

`xx#crossed {...}`

elemento **XX** con **id="crossed"** (NB. unicita' di id)

`th[rowspan] {...}` elemento **th** in cui l'attributo **rowspan** e' assegnato

`th[rowspan="2"] {...}` elemento **th** in cui l'attributo **rowspan** e' assegnato a **2**

`h1 > em {...}`

occorrenze di un elemento **em** in un **h1**, come figlio

`em + h1 {...}`

occorrenze di **h1** subito dopo un **em**

externalStylesheet.html

```
<?xml version = "1.0"?>
  ...
<html ...>
<head><title>stili esterni</title>
<link rel="stylesheet" href="external.css"
      type="text/css" /></head>

<body>
<h1>...external</h1>anche h1 e' block-level
<h1 class="speciale">(h1 non standard)</h1>

<table border="1" cellspacing="3" cellpadding="5">
<thead>
  <tr>...</tr>
</thead>

<tbody>
  <tr>
    <th rowspan="2">verdura</th>
    <td>zucchine</td>
    <td>€2,50</td>
  </tr>
  ...
</tbody>
</table>

<p>The End.</p>
<p class="spostato">The End.</p>
</body></html>
```



```
body {background-color: yellow;
       color: blue}
      /* COMMENTO */
thead {background-color: green;}
```

The End.

```
h1.speciale {
  text-align: center;
  color: purple;
font-family: Arial,sans serif;
}
/* COMMENTO */
p.spostato {color:black;
  font-size: 120%;
  text-align: right;}
```

```
th[rowspan="2"] {
  vertical-align: top; }
```

external.css

... esempi di stylesheet properties

- font-family (es. con un nome, come Times New Roman, Arial, ... o generiche come serif, sans serif, monotype, cursive, fantasy)
- font-size (dimensione font)
- font-style (valori definiti, es. normal, oblique, italics)

- color (text color: specifica di un colore)
- background-color
- background-image
- background-attachment (scroll o fixed immagine di background)

- text-align (valori definiti – es. *left*, ...)
- text-transform (es: lowercase, uppercase)
- list-style-type (es. disk, square, lower-roman)
- letter-spacing, word-spacing

- width, height, margin, padding, border (misure - dimensione - layout)

- top (y coord. della cima dell'elemento)
- left (x coord. della cima dell'elemento)
- z-index (precedenza: se sovrapposti, quali elementi appaiono sopra e quali sotto?)
- float (allineamento rispetto ad altri elementi – es. testo intorno ad un elemento)

dimensioni

Ci sono misure cosiddette assolute e cosiddette relative, basate su valori numerici o enumerati...

Le definizioni di dimensione riguardano la grandezza dei caratteri, o caratteristiche di resa grafica di elementi (es. la larghezza di un elemento block-level).

Absolute rispetto a che cosa? Relative a che cosa?

- px pixel (evidentemente dipende dallo schermo ...)
- em font size della **font in uso nell'elemento padre (*relevant*)**
- ex altezza della x nella relevant font
- % percentuale della dimensione associata all'elemento padre

- in inches
- mm
- cm
- pt point (*1/72th of an inch*)
- pica 12 punti (tradizionale nell'editoria)

- xx-small | x-small | small | medium | large | x-large | xx-large
valori predefiniti, sensibili alle impostazioni dell'applicazione utente
- larger | smaller
(dipendono dalla dimensione del carattere nell'elemento padre)

Specifica colori

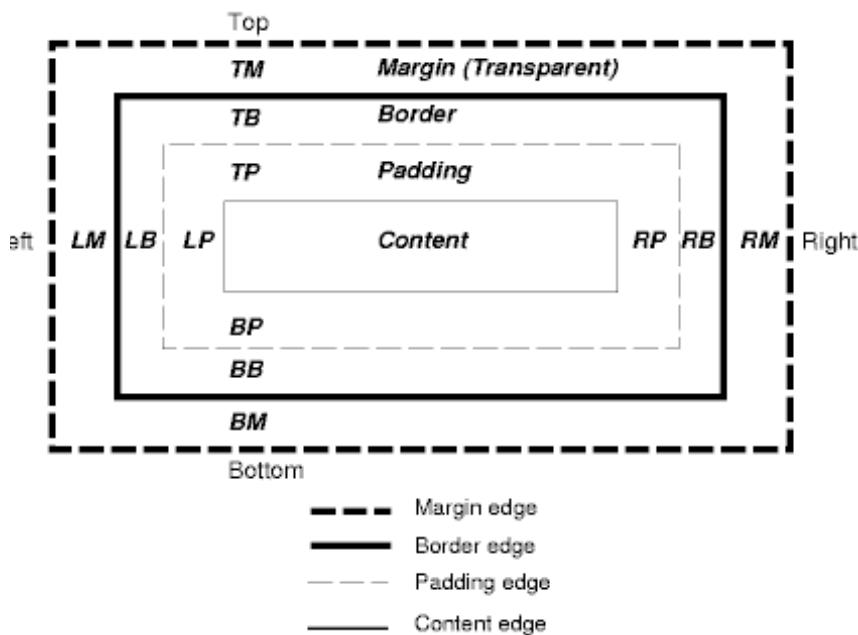
specifica base rgb

#**xxxxxx** 6 cifre esadecimali (2 per ciascun colore base)

16 milioni di colori

```
p {color: #000000} /* nero */    p {color: #FF0000}  
p {color: #f00} /* notazione veloce per ff0000 */  
p {color: rgb(0,0,255)}          /* 0-255 */
```

MODELLO DI BOX



Specifica CSS box

Legenda

TM/TB/PP = Top
Margin/Border/Padding
B=Bottom
L=Left
R=Right

Ogni elemento e' racchiuso in una box, in cui
border = parte potenzialmente visibile, intorno al contenuto dell'elem.
padding = area tra il contenuto e il bordo
margin = area tra l'elem. e gli altri elem. vicini

CSS VANTAGGI

- Chiara **separazione** tra contenuto, strutturazione, e regole di presentazione
- Regole di presentazione
 - definibili piu' **rigorosamente**, e piu' **riccamente** rispetto ad HTML
 - condivisibili da piu' file (si uniforma il *look-and-feel* del sito web e lo si rende piu' **facilmente mantenibile e modificabile**)
 - permettono adattamento al viewport del device

positioning.html (1/4)

positioning.css

```
.navbuttons {  
    position: absolute;  
    top: 0%;  
    left: 0%;  
    background: #09F;  
    border: 5px outset;  
    margin: 10px;  
    z-index: 7  
}  
  
.button {  
    border: 5px outset #339933;  
    margin: 4px  
}
```

Posizionamento assoluto di oggetti



bottoni di navigazione

applicazione del posizionamento assoluto: i bottoni sono resi in questo documento html usando la classe navbutton

```
<div class="navbuttons">  
    <a href="dummy.html">  
        </a>  
    <a href="dummy.html">  
        </a>  
    <a href="dummy.html">  
        </a>  
</div>
```

la classe navbutton e quella button (notare che il posizionamento della caption e' "relative", volendo cosi' dire che le specifiche date vengono attuate "a partire dalla posizione naturale dell'elemento":

```
.navbuttons {  
    position: absolute;  
    top: 0%;  
    left: 0%;  
    background: #09F;  
    border: 5px outset;  
    margin: 10px;  
    z-index: 7  
}  
  
.button {border: 5px outset #339933; margin: 4px}
```

```
<div class="navbuttons">  
    <a href="dummy.html">  
        </a>  
    <a href="dummy.html">  
        </a>  
    <a href="dummy.html">  
        </a>  
</div>
```

50/42

MEDIA QUERY

1. Permettono di usare fogli di stile differenti
2. Il particolare foglio di stile è determinato dal media (ossia stampante, schermo e risoluzione)

BOOTSTRAP4

- Foglio di stile predefinito
- Principale caratteristica: **Sistema a griglia**

BS4

Be aware of the limitations and bugs around flexbox, like the inability to use some trivial elements as flex containers.

Grid options

While Bootstrap uses `ems` or `rems` for defining most sizes, `pxs` are used for grid breakpoints and container widths.

This is because the viewport width is in pixels and does not change with the `font size`.

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	Extra small ≤576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				