

INTERNET

I MATTONI FONDAMENTALI



Come è fatta Internet

Internet è una rete ad accesso pubblico che connette vari dispositivi in tutto il mondo. Dalla sua nascita rappresenta il principale mezzo di comunicazione di massa, che offre all'utente una vasta serie di contenuti potenzialmente informativi e di servizi.

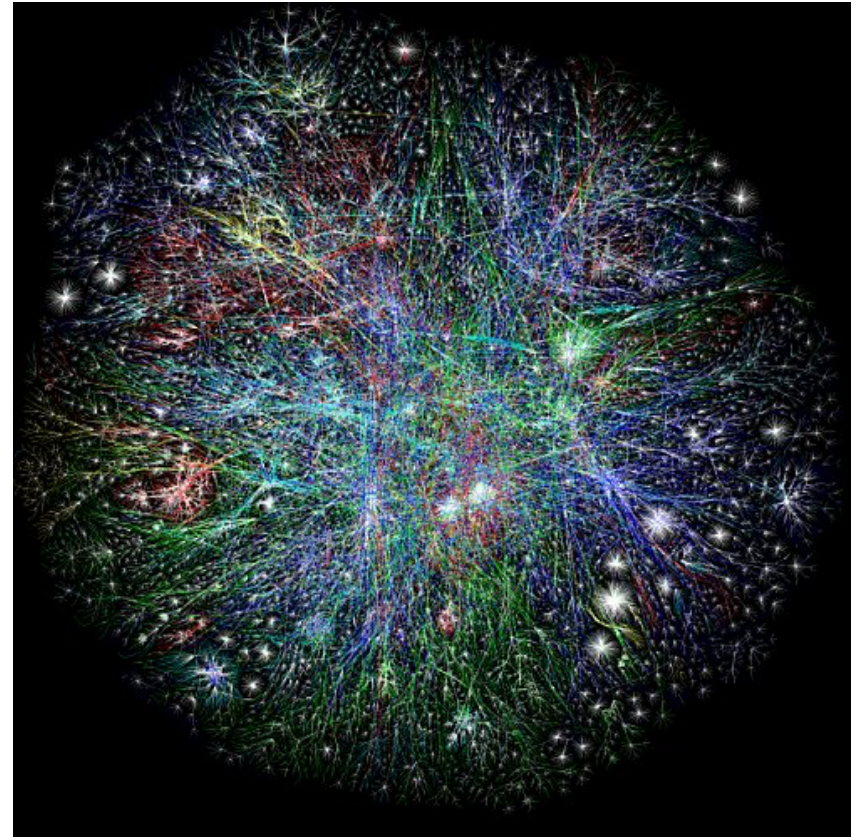
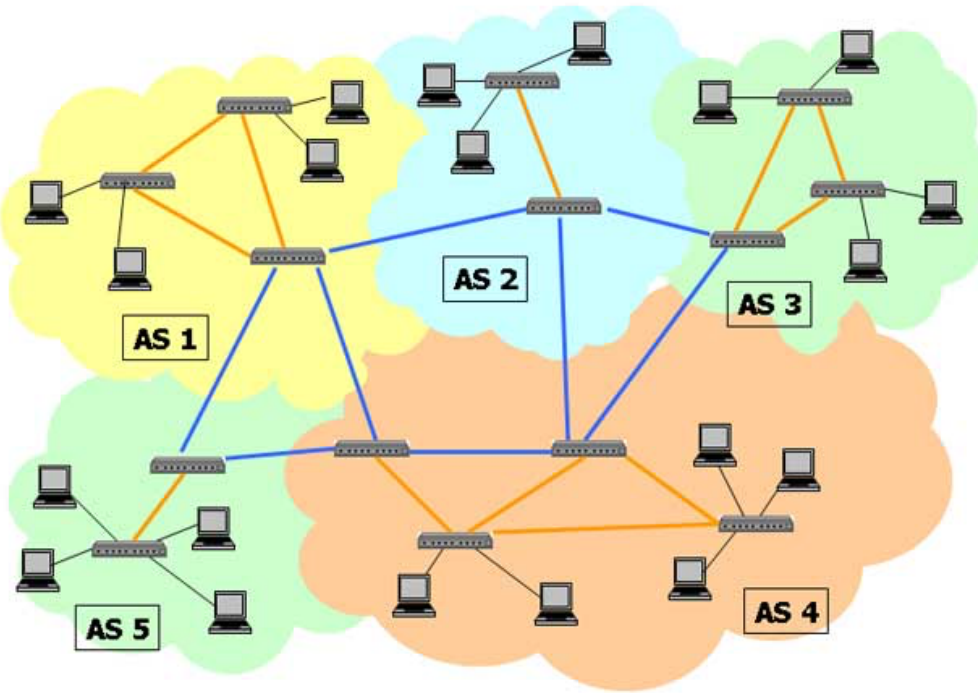
L'origine di Internet risale agli anni sessanta, su iniziativa degli Stati Uniti, che misero a punto durante la guerra fredda un nuovo sistema di difesa e di controspionaggio. ARPANET nel 1969, finanziato da Defence Advanced Research Projects Agency.

L'Italia fu il terzo Paese in Europa a connettersi in rete, dopo Norvegia e Inghilterra, grazie ai finanziamenti del Dipartimento della Difesa degli Stati Uniti. La connessione avvenne dall'Università di Pisa. Il collegamento avvenne il 30 aprile 1986, alle 18 circa.

Il precursore del servizio di posta elettronica fu ideato nel 1973, su ARPANET. Le emoticon vennero isituite nel 1979 per suggerire lo stato d'animo nelle email.



Qualche immagine



Esempio di una rete

LAN: Local Area Network, copre un'area limitata come una scuola, un'azienda

WLAN: LAN Wireless

WAN: Wide Area Network, è una rete che copre una o più regioni geografiche

BAN: Body Area Network se la rete si estende intorno al corpo dell'utilizzatore con una estensione dell'ordine del metro

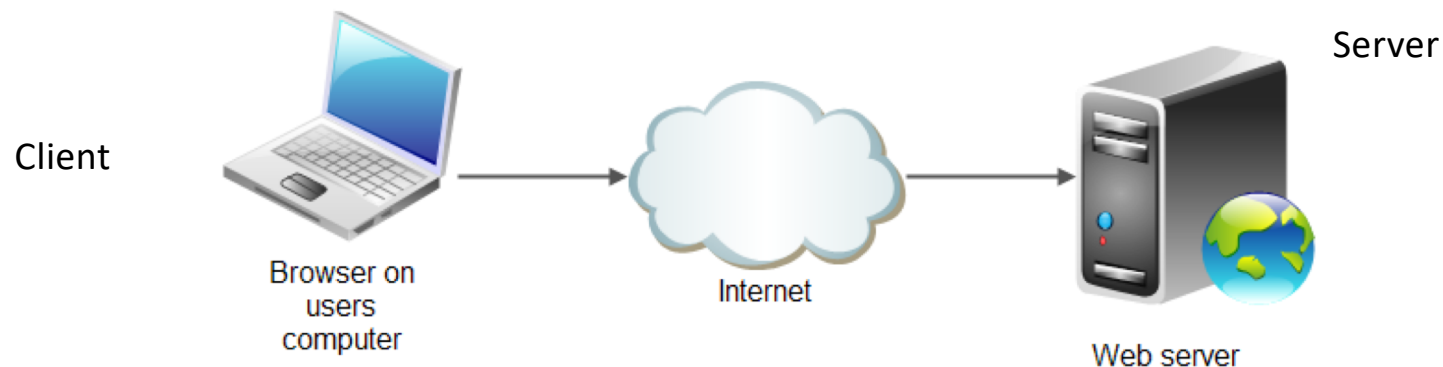
PAN: Personal Area Network, se la rete si estende intorno all'utilizzatore con una estensione di alcuni metri

GARR è la rete telematica italiana dell'Università e della Ricerca ed il suo principale obiettivo è quello di fornire connettività ad altissime prestazioni e servizi avanzati alla comunità scientifica ed accademica italiana. La rete GARR è ideata e gestita dal Consortium GARR, un'associazione senza fini di lucro fondata con il patrocinio del Ministero dell'Istruzione, dell'Università e della Ricerca.



World Wide Web

Nel 1991 presso il CERN di Ginevra il ricercatore Tim Berners-Lee definì il protocollo HTTP (**HyperText Transfer Protocol**), un sistema che permette una lettura ipertestuale, non-sequenziale dei documenti, saltando da un punto all'altro mediante l'utilizzo di rimandi (link o, più propriamente, **hyperlink**). Il primo browser con caratteristiche simili a quelle attuali, il Mosaic, venne realizzato nel 1993. Esso rivoluzionò profondamente il modo di effettuare le ricerche e di comunicare in rete. Nacque così il **World Wide Web (WWW)**. In particolare, il 6 agosto 1991 Berners-Lee pubblicò il primo sito Web al mondo, presso il CERN, all'indirizzo <http://info.cern.ch/hypertext/WWW/TheProject.htm>



Cos'è un browser

In informatica, il **Web browser** è un'applicazione per il recupero, la presentazione e la navigazione di risorse sul Web. Tali risorse (come pagine web, immagini o video) sono messe a disposizione sul **World Wide Web** (la rete globale che si appoggia su **Internet**), su una rete locale o sullo stesso computer dove il browser è in esecuzione.

Il programma implementa

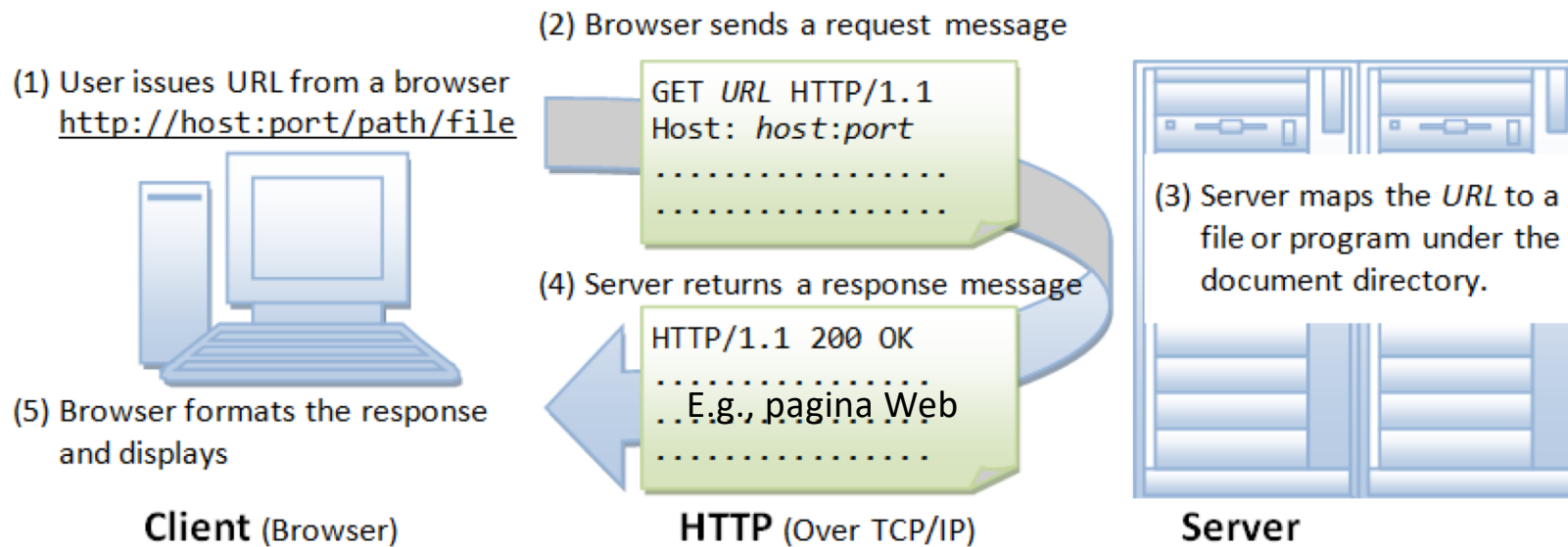
1. da un lato le funzionalità di client per il **protocollo HTTP**, che regola lo scaricamento delle risorse dai server web a partire dal loro indirizzo URL;
2. dall'altro quelle di visualizzazione dei contenuti ipertestuali (solitamente all'interno di documenti HTML) e di riproduzione di contenuti multimediali.

Tra i browser più utilizzati vi sono Google Chrome, Safari, Internet Explorer, Mozilla Firefox, Opera, Microsoft Edge (uscito con Windows 10) e Maxthon.



HTTP

L'**HyperText Transfer Protocol (HTTP)** (protocollo di trasferimento di un ipertesto) è un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul Web ovvero in un'architettura tipica client-server.



Esempio

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

<h1>La mia prima intestazione</h1>

<p>Il mio primo paragrafo.</p>

<h2>La mia seconda intestazione</h2>

<p>Il mio secondo paragrafo.</p>

</body>
</html>
```

La mia prima intestazione

Il mio primo paragrafo.

La mia seconda intestazione

Il mio secondo paragrafo.

HTML

COME STRUTTURARE IL CONTENUTO DI UNA PAGINA



Un po' di storia

L'**HyperText Markup Language (HTML)**; traduzione letterale: linguaggio a marcatori per ipertesti), in informatica è il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine Web.

È un linguaggio di pubblico dominio, la cui sintassi è stabilita dal **World Wide Web Consortium (W3C)**.

L'HTML è stato sviluppato verso la fine degli anni ottanta del XX secolo da Tim Berners-Lee al CERN di Ginevra assieme al protocollo HTTP dedicato al trasferimento di documenti in tale formato.

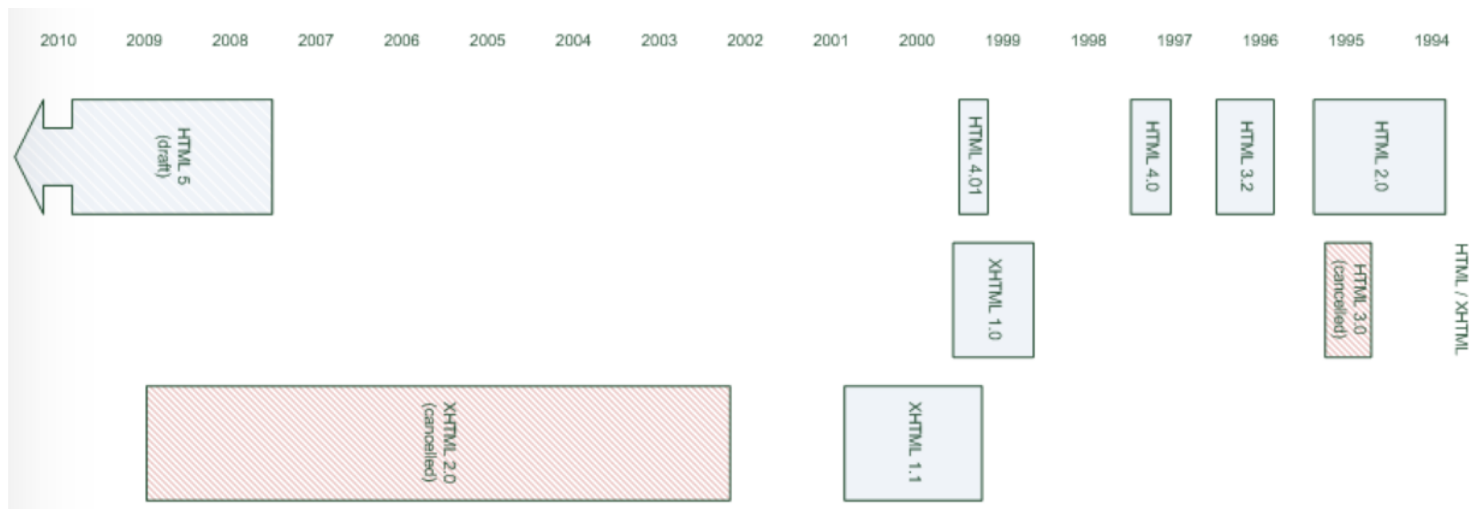
Nel giugno 1993 Berners-Lee ne pubblica una prima definizione formale, firmata insieme a Daniel Connolly e sostenuta dal gruppo di lavoro dell'Internet Engineering Task Force chiamato Integration of Internet Information Resources , per proporla come standard IETF

Standard

Nel gennaio 2000, viene pubblicato come standard **XHTML 1.0**, una variante di **HTML 4.01** che usa **XML 1.0** anziché in SGML come metalinguaggio di markup, per migliorarne l'interoperabilità con altri linguaggi della famiglia.

Il W3C decide di abbandonare HTML e di continuare lo sviluppo solo su XHTML: **XHTML 1.0** diventa W3C recommendation (standard) il 26 Gennaio 2000, e **XHTML 1.1** il 31 Maggio 2001.

L'XHTML è il successore diretto dell'HTML 4.1.




HTML5

Lo sviluppo venne avviato dal gruppo di lavoro **Web Hypertext Application Technology Working Group (WHATWG)** (fondato nel 2004 da sviluppatori appartenenti ad Apple, Mozilla Foundation ed Opera Software) che si pose come obiettivo quello di progettare delle specifiche per lo sviluppo di applicazioni web, focalizzandosi su miglioramenti e aggiunte ad HTML e alle tecnologie correlate. Inizialmente in contrasto con il World Wide Web Consortium per le lungaggini nel processo di evoluzione dello standard html e per la decisione del W3C di orientare la standardizzazione verso l'XHTML 2 che non garantiva retro compatibilità,

Lo stesso W3C ha poi riconosciuto valide tali motivazioni, annunciando di creare un apposito gruppo per la standardizzazione dell'HTML5 e abbandonare l'XHTML 2.0. Dal 2007 il WHATWG ha collaborato con il W3C in tale processo di standardizzazione, per poi decidere nel 2012 di separarsi dal processo di standardizzazione del W3C, creando di fatto due versioni dell'HTML5: la versione del WHATWG viene definita come "HTML Living Standard" e quindi in continua evoluzione, mentre quella del W3C sarà una unica versione corrispondente ad uno "snapshot" del Living Standard.

Il 28 Ottobre 2014, HTML5 è diventato ufficialmente standard W3C
(<http://www.w3.org/TR/html5/>)



Struttura

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

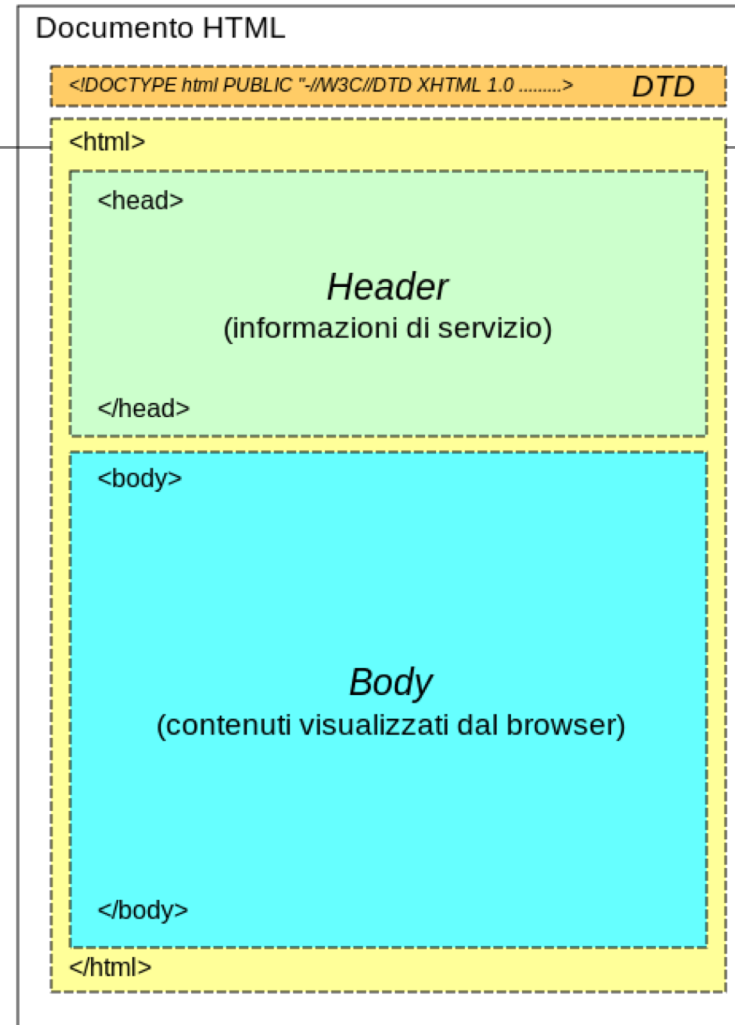
```
<title>Conforming HTML5 Template</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```



Struttura

```
<!DOCTYPE html>
<html>
<body>
  This is my first web page
</body>
</html>
```

La prima linea **<!DOCTYPE html>** è la **dichiarazione del tipo di documento** (che segue). Fa sapere al browser quale tipo di HTML deve utilizzare per interpretare la pagina (in questo caso, HTML5). Va inserito sempre!

Il tag **<html>** è il tag di apertura che dice al browser che tutto quello che si trova tra quello e il corrispondente tag di chiusura **</html>**, è un documento html.

Tra **<body>** e **</body>** va aggiunto tutto quello che si vuole sia visualizzato nella pagina. **ATTENZIONE:** tutto ciò che è fuori questi due tag, non viene visualizzato e non fa parte della pagina vera e propria.

Non tutti i tag hanno il corrispondente tag di chiusura, per esempio **
** (line break). In XHTML il tag vuole comunque uno slash di chiusura (**
**), mentre in HTML5 basta **
**.

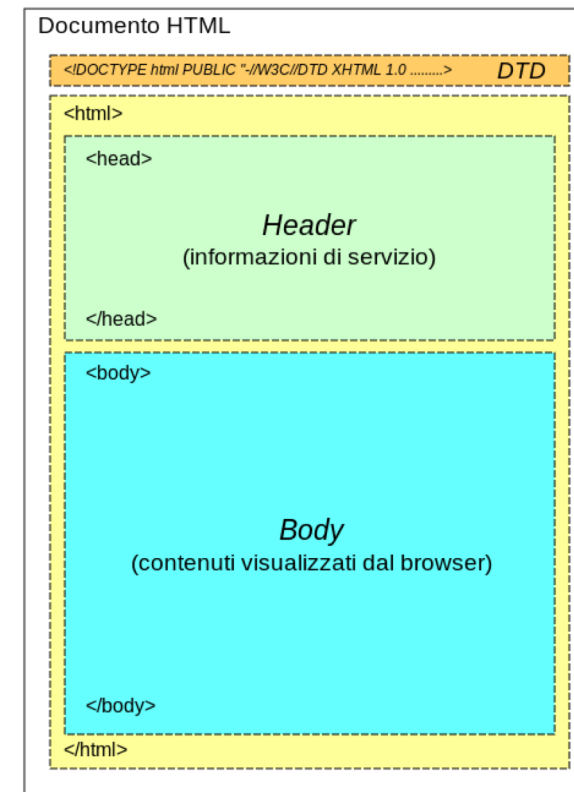
Differenti DTD

DTD = Document Type Definition. Il Document Type Definition (definizione del tipo di documento) è uno strumento utilizzato dai programmatori il cui scopo è quello di definire le componenti ammesse nella costruzione di un documento XML

HTML5

```
<!DOCTYPE html>
```

```
<html>  
<head>  
<title>Conforming HTML5 Template</title>  
</head>  
<body>  
</body>  
</html>
```

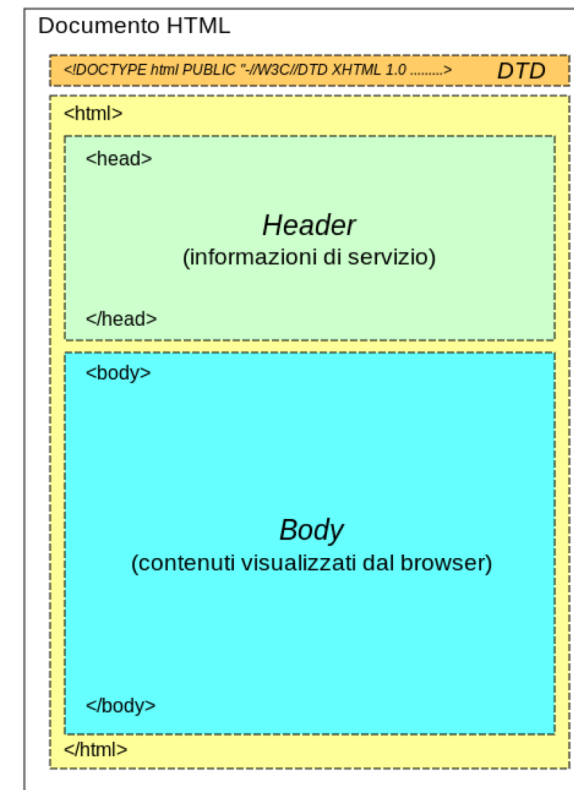


Differenti DTD

DTD = Document Type Definition. Il Document Type Definition (definizione del tipo di documento) è uno strumento utilizzato dai programmatori il cui scopo è quello di definire le componenti ammesse nella costruzione di un documento XML

HTML 4.1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">  
<html>  
<head>  
<title>Conforming HTML 4.1 Template</title>  
</head>  
<body>  
</body>  
</html>
```



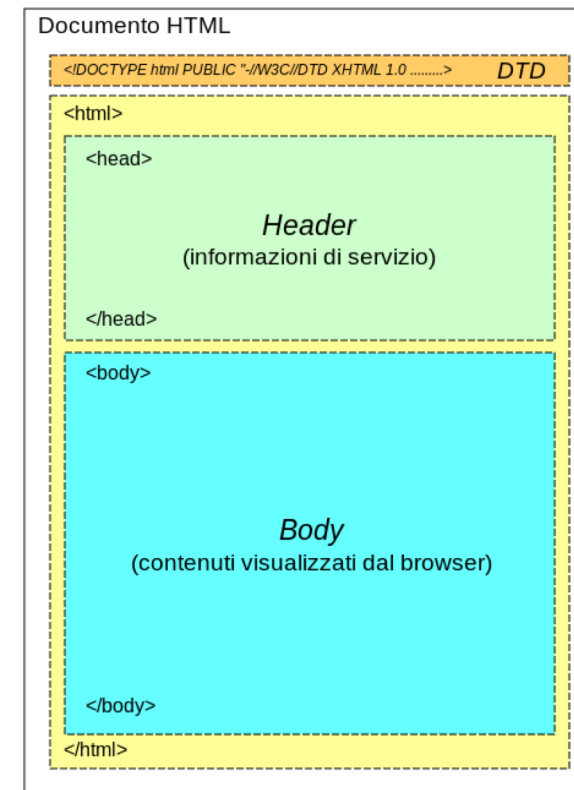
Differenti DTD

DTD = Document Type Definition. Il Document Type Definition (definizione del tipo di documento) è uno strumento utilizzato dai programmatori il cui scopo è quello di definire le componenti ammesse nella costruzione di un documento XML

XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html>  
<head>  
<title>Conforming XHTML 1.1 Template</title>  
</head>  
<body>  
</body>  
</html>
```



Head

L'elemento `<head>` (che inizia con il tag di apertura `<head>` e termina con il tag di chiusura `</head>`) appare prima dell'elemento `body`, e contiene informazioni a proposito della pagina. Questa informazione non appare nella finestra del contenuto della pagina, nel browser.

Il titolo comparirà come riferimento della pagina se la salvate tra i preferiti.

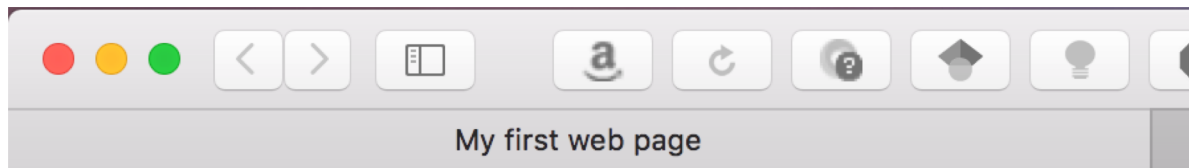


Contenuto della pagina



Contenuto in body

Il browser non interpreta il testo come lo avete scritto: se andate a capo, oppure se mettete più spazi, il risultato visualizzato sarà sempre:



This is my first web page. How exciting

Per avere due linee serve

```
<p> This is my first web page </p>  
<p> How exciting </p>
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>My first web page</title>  
</head>  
<body>  
  This is my first web page.  
  How exciting  
</body>  
</html>
```

I paragrafi <p>


Il paragrafo <p> è un elemento contenitore che al suo interno prevede l'inserimento di testo e di altri tag.

<p>Nel primo paragrafo di questa trattazione, ci occuperemo dell'importanza del testo nel Web. Grazie ad un semplice esempio possiamo sperimentare molte cose.</p>

<p>Per default il browser manderà a capo il contenuto di questo secondo paragrafo.</p>

Nel primo paragrafo di questa trattazione, ci occuperemo dell'importanza del testo nel Web. Grazie ad un semplice esempio possiamo sperimentare molte cose.

Per default il browser manderà a capo il contenuto di questo secondo paragrafo.



Altri tag

Si può enfatizzare il testo, e dargli maggiore importanza

`<p>Yes, that really is exciting. Warning: level of excitement may cause head to explode.</p>`

*Yes, that really is exciting. **Warning:** level of excitement may cause head to explode.*

Tradizionalmente il browser interpreta `` in italico, e `` in grassetto. Esistono anche `<i>` e ``, ma è meglio non utilizzarli, la presentazione non deve far parte di HTML (ma di CSS).

`
`, come già intravisto serve per separare le linee. Da usare con parsimonia: probabilmente il tag che si vuole è `<p>`.

```
<!DOCTYPE html>
<html> <head> <title>Titoli</title> </head>

<body>
This is my first web page <br> How exciting
</body></html>
```

This is my first web page
How exciting

Titoli (headings)

Esistono 6 differenti livelli di titolo che si ottengono utilizzando i seguenti tag:

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

La “h” sta per “heading”, cioè titolo e ogni livello indica una grandezza diversa.

Dall'`<h1>`, che è il più importante, si va via via degradando fino all'`<h6>` che rappresenta il titolo con grandezza minore.

In genere sono rappresentati in grassetto e con una dimensione del testo ingrandita.

Gli heading, così come altri tag contenitori, appartengono alla famiglia degli elementi di blocco (block). Tali elementi sono così chiamati perché una volta definiti creano un blocco attorno a sé, di conseguenza vanno a capo e risultano distanziati dagli elementi che li circondano.

Titoli (esempio)

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Titoli</title>
```

```
</head>
```

```
<body>
```

```
<h1>Esame di informatica</h1>
```

```
<h2>Esame di informatica</h2>
```

```
<h3>Esame di informatica</h3>
```

```
<h4>Esame di informatica</h4>
```

```
<h5>Esame di informatica</h5>
```

```
<h6>Esame di informatica</h6>
```

```
</body>
```

```
</html>
```

Esame di informatica

Esame di informatica

Esame di informatica

Esame di informatica

Esame di informatica

Esame di informatica

Citazioni

Gli elementi **blockquote** e **q** sono utilizzati per delimitare parte del testo come citazione.

Solitamente **q** è utilizzato per citazioni in linea, più corte, rispetto a **blockquote**:

<body>

<p>So I asked Bob about quotations on the Web and he said **<q>**I know as much about quotations as I do about pigeon fancying**</q>**. Luckily, I found HTML Dog and it said:**</p>**

<blockquote>

<p>blockquote and q are used for quotations. blockquote is generally used for standalone often multi-line quotations whereas q is used for shorter, in-line quotations.**</p>**

</blockquote>

</body>

Titoli, paragrafi, blocchi testo | Gui...

Text: Abbreviations, Quotations,...

Untitled.html

So I asked Bob about quotations on the Web and he said "I know as much about quotations as I do about pigeon fancying". Luckily, I found HTML Dog and it said:

blockquote and q are used for quotations. blockquote is generally used for standalone often multi-line quotations whereas q is used for shorter, in-line quotations.

Liste

Esistono le liste ordinate (``) e non ordinate (``). Gli elementi delle prime vengono solitamente rappresentate con un numero sequenziale, mentre per le seconde, con un pallino (bullet). Ogni elemento delle liste viene contenuto in due tag ` ... `.

```
<!DOCTYPE html>
<html>
<head>
<title>Liste</title>
</head>

<body>
<ul>
<li>To learn HTML</li>
<li>To show off</li>
<li>Because I've fallen in love with my computer and want
to give her some HTML loving.</li>
</ul>
</body>
</html>
```

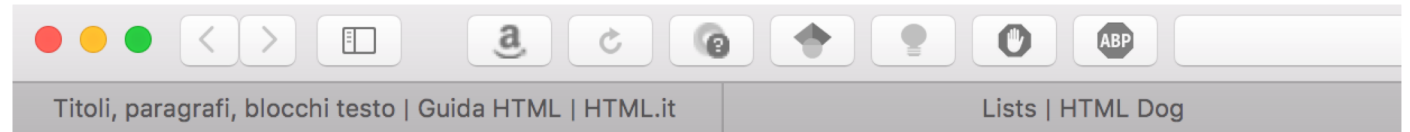


- To learn HTML
- To show off
- Because I've fallen in love with my computer and want to give her some HTML loving.

Liste ordinate

```
<!DOCTYPE html>
<html>
<head>
<title>Liste ordinate</title>
</head>
```

```
<body>
<ol>
<li>To learn HTML</li>
<li>To show off</li>
<li>Because I've fallen in love with my computer and want to give her some
HTML loving.</li>
</ol>
</body>
</html>
```



1. To learn HTML
2. To show off
3. Because I've fallen in love with my computer and want to give her some HTML loving.

Collegamenti, immagini, e tabelle

ALTRI ELEMENTI

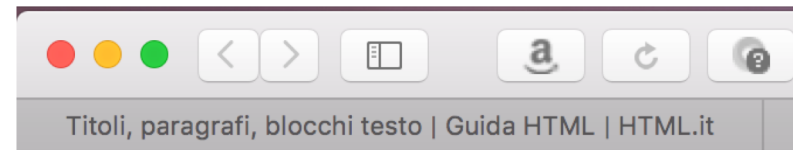


Collegamenti (link)

Per ora abbiamo visto com'è fatta una pagina, ma “Hyper Text” in HTML significa ipertesto, cioè più documenti di testo collegati insieme (questo è il WWW).

Il tag ancora (anchor) `<a>` serve a delimitare la parte del testo che, una volta cliccata, ci porta ad un altro documento. In più deve esser eperò anche specificato un link di destinazione.

```
<h2>Where to find the tutorial</h2>
<p>
<a href="http://www.htmldog.com">HTML Dog</a>
</p>
```



Where to find the tutorial

[HTML Dog](http://www.htmldog.com)

La destinazione è specificata tramite attributo href dell'elemento ancora. In questo caso è assoluto, ma può essere anche relativo se si riferisce ad un pagina sullo stesso computer.

```
<a href="miapagina.html">Altra pagina sul mio computer</a>
```

Collegamenti ad altre pagine sul proprio computer

La destinazione è specificata tramite attributo **href** dell'elemento ancora. In questo caso è assoluto, ma può essere anche relativo se si riferisce ad un pagina sullo stesso computer.

Altra pagina sul mio computer

Con solamente il nome, per esempio **miapagina.html**, il browser cerca l'immagine nella stessa cartella della pagina in cui si sta inserendo il link.

Se l'immagine non si trova nella stessa cartella, **ESSA NON VERRÀ VISUALIZZATA.**

Se l'immagine si trova altrove, è necessario inserire il percorso dove si trova l'immagine, e non solo il nome. Lo vedremo più avanti.

Salti all'interno della stessa pagina

Può essere associato un identificatore, tramite attributo **id**, ad un elemento della pagina.

```
<h2 id="dogs">Dogs</h2>
```

...

```
<h2 id="cats">Cats</h2>
```

Poi, quando si vuole saltare in quel punto da un'altra parte della stessa pagina

```
<a href="#dogs">Go to Dogs</a>
```

Alcune precisazioni risp. standard

HTML5 non richiede che gli attributi siano specificati in minuscolo: per esempio **ALT** oppure **Alt** sono attributi validi. W3C raccomanda comunque di scriverli minuscoli e richiede (obbligatoriamente) che siano minuscoli per XHTML.

Stessa cosa per quanto riguarda gli apici. W3C raccomanda di utilizzare gli apici per quotare il valore dell'attributo, e richiede la loro presenza se si usa XHTML. Altrimenti HTML5 permette anche di scrivere

```
<a href=http://www.w3schools.com>
```

invece di

```
<a href="http://www.w3schools.com">
```

A volte comunque è necessario utilizzarli: quando ci sono degli spazi nel valore:

NO

```
<p title=About W3Schools>
```

SI

```
<p title="About W3Schools">
```

Immagini

Le immagini si comportano come i link, nel senso che è sufficiente menzionare (con un tag differente da quello dell'ancora *a* il nome del file dell'immagine).

Se l'immagine si trova nella stessa cartella della pagina, è sufficiente utilizzare il tag **img**.

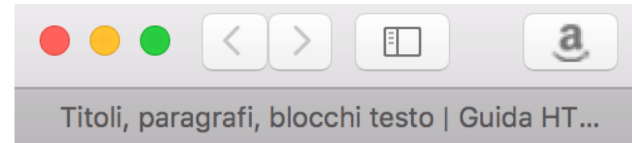
Per esempio se si vuole inserire l'immagine che sta nel file **nome_immagine.jpg**, si può utilizzare

```

```


Immagini

In questo esempio, utilizziamo una immagine che si trova su Internet ad un dato indirizzo, e non sul nostro computer.



```
<!DOCTYPE html>
<html>
<head>
<title>Titoli</title>
</head>
```

```
<body>
<h3>Pagina qui sotto</h3>

</body>
```

```
</html>
```

Pagina qui sotto



Alcuni di tipi di file immagine

JPEG (Joint Photographic Expert Group) è prodotta tramite un algoritmo di compressione, che modifica (più o meno) l'immagine iniziale. È di solito utilizzate per le fotografie.

GIF (Graphics Interchange Format): non può avere più di 256 colori. Meno colori portano ad una dimensione minore. Non è un formato compresso ma bitmap. Utilizzata in genere per icone o loghi. Un pixel può essere anche trasparente.

PNG (Portable Network Graphics): come GIF ma consente 16 milioni di colori. Anche questo formato può essere utilizzato per foto.



Attributi di elementi

Tutti gli elementi HTML possono avere attributi. Essi aggiungono ulteriore informazione a proposito dell'elemento in questione.

Vengono specificati sempre nel tag di apertura e solitamente sono definiti tramite la coppia **name="value"** (nome dell'attributo e suo valore).

```
<!DOCTYPE html>  
<html lang="en-US">  
<body>  
...  
</body>  
</html>
```

```
<p title="I'm a tooltip">  
This is a paragraph.  
</p>
```

```

```

```

```

```
<a href="http://www.w3schools.com">This is a link</a>
```

Tabelle

Storicamente le tabelle sono state utilizzate (in modo abusato) per impaginare tutta una pagina; per esempio l'intestazione era una riga, la colonna di sinistra era una colonna della tabella, e così via. Per la visualizzazione ci penseremo quando faremo CSS.

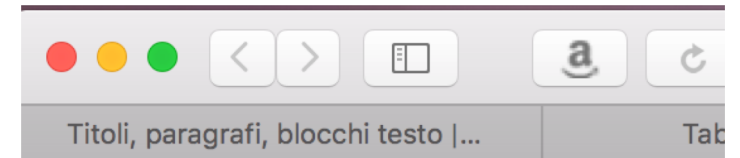
```
<body>
<table>
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
<td>Row 1, cell 3</td>
</tr>

<tr>
<td>Row 2, cell 1</td>
<td>Row 2, cell 2</td>
<td>Row 2, cell 3</td>
</tr>

<tr>
<td>Row 3, cell 1</td>
<td>Row 3, cell 2</td>
<td>Row 3, cell 3</td>
</tr>

<tr>
<td>Row 4, cell 1</td>
<td>Row 4, cell 2</td>
<td>Row 4, cell 3</td>
</tr>
</table>
</body>
```

table = tabell
td = data cell
tr = table row



Row 1, cell 1 Row 1, cell 2 Row 1, cell 3
Row 2, cell 1 Row 2, cell 2 Row 2, cell 3
Row 3, cell 1 Row 3, cell 2 Row 3, cell 3
Row 4, cell 1 Row 4, cell 2 Row 4, cell 3

Tabelle con intestazioni

La prima riga può avere elementi `<th>` invece che `<td>`.

Rappresentano gli header della tabella, solitamente in grassetto.

| | Firstname | Lastname | Age |
|------|------------------|-----------------|------------|
| Jill | Smith | | 50 |
| Eve | Jackson | | 94 |
| John | Doe | | 80 |

```
<table>
<tr>
  <th>Firstname</th>
  <th>Lastname</th>
  <th>Age</th>
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

Tabelle con rowspan e colspan

```
<table>
<tr>
<td>Name 1</td>
<td>Name 2</td>
<td colspan="2">Name 3</td>
<td>Name 4</td>
</tr>
```

```
<tr>
<td>name</td>
<td>price</td>
</tr>
```

```
<tr>
<td rowspan="3">ITEM 1</td>
<td rowspan="3">ITEM 2</td>
<td>name</td>
<td>price</td>
<td rowspan="3">ITEM 4</td>
</tr>
```

```
<tr>
<td>name</td>
<td>price</td>
</tr>
</table>
```

| | | | |
|--------|--------|--------|--------|
| Name 1 | Name 2 | Name 3 | Name 4 |
| ITEM 1 | ITEM 2 | name | price |
| | | name | price |
| | | name | price |



Tabelle con dentro tabelle

```
<table>
<tr>
<td> Elem1 Elem2
      Elem3 Elem4 The cell next to this one has a smaller table inside of it, a table inside a table.
<table>
<tr>
<td>2nd Table</td>
<td>2nd Table</td>
</tr>
<tr>
<td>2nd Table</td>
<td>2nd Table</td>
</tr>
</table>
</td>

<td> The cell next to this one has a smaller table inside of it, a table inside a table.</td>
</tr>
</table>
```



Percorsi (path)

COME ORGANIZZARE IL CONTENUTO DELLE DIRECTORY



Percorsi

Certe volte, per organizzare meglio tutti i file che compongono il sito Web che si sta costruendo, è necessario suddividerli in più cartelle. Per esempio i file delle immagini possono essere spostati in una cartella a parte.

Tutto quello che abbiamo detto sui link e sulle immagini deve essere modificato. Se mettiamo solo il nome, il browser cerca pagine html immagini nella stessa cartella della pagina in cui stiamo mettendo l'immagine od un link ad un'altra pagina.

Per questo motivo è necessario non specificare solo il nome, ma un percorso verso quella particolare immagine o pagina web a cui ci vogliamo riferire.

Esempio

Immagini

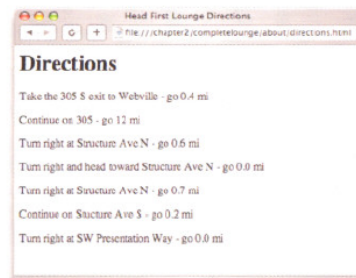
Here's the new and improved page.

We've added links to two new pages, one for elixirs and one for driving directions.



The "elixirs" link points to a page with a full list of elixir selections.

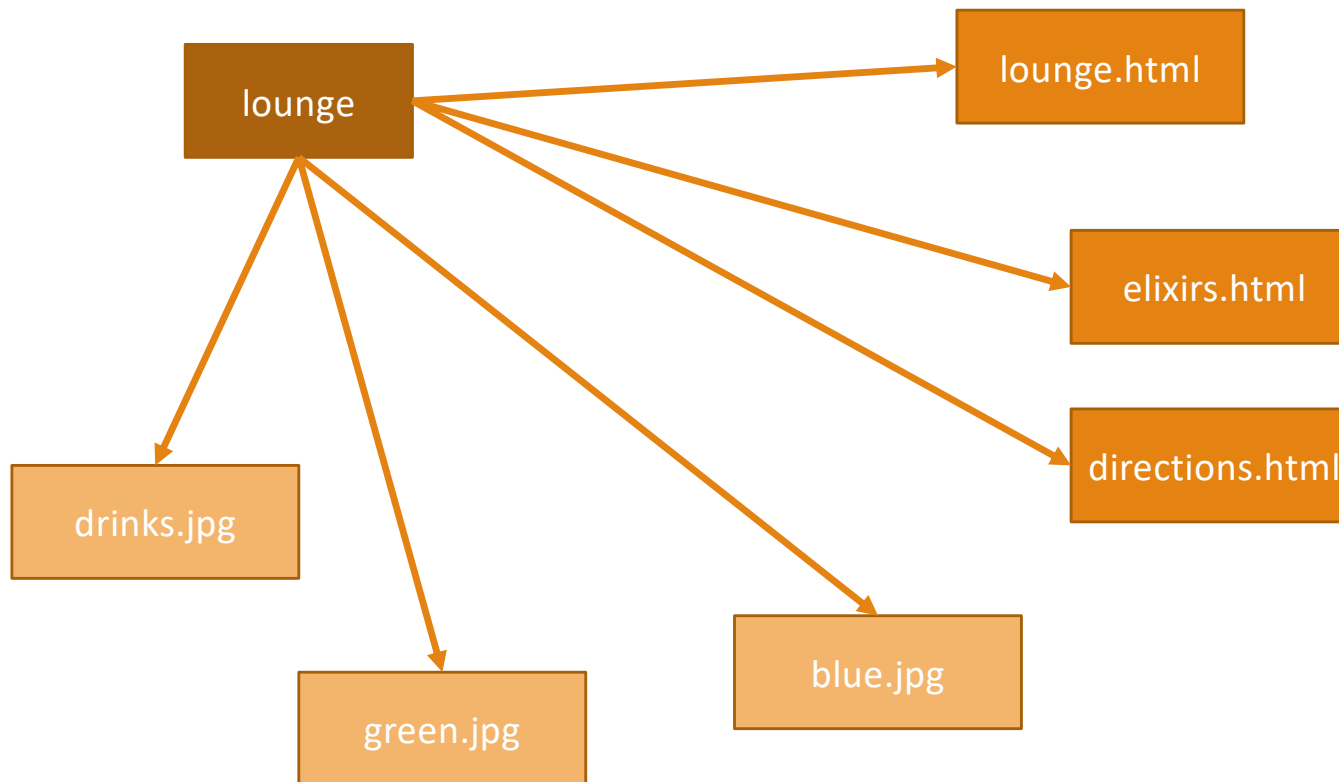
Altra pagina come Directions



The "detailed directions" link leads to an HTML page with driving directions.

directions.html

Organizzare le directory!



Una cartella di progetto, lounge, con dentro tutti i file: tutte le pagine html e tutte le immagini.

Come l'Esercizio 2 su Sushi

```
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for
      refreshing <a href="elixir.html">elixirs</a>,
      conversation and maybe a game or two of
      <em>Dance Dance Revolution</em>.
      Wireless access is always provided;
      BYOWS (Bring your own web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown Webville.
      If you need help finding us, check out
      our <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>
```

Let's add "New and Improved" to the heading.

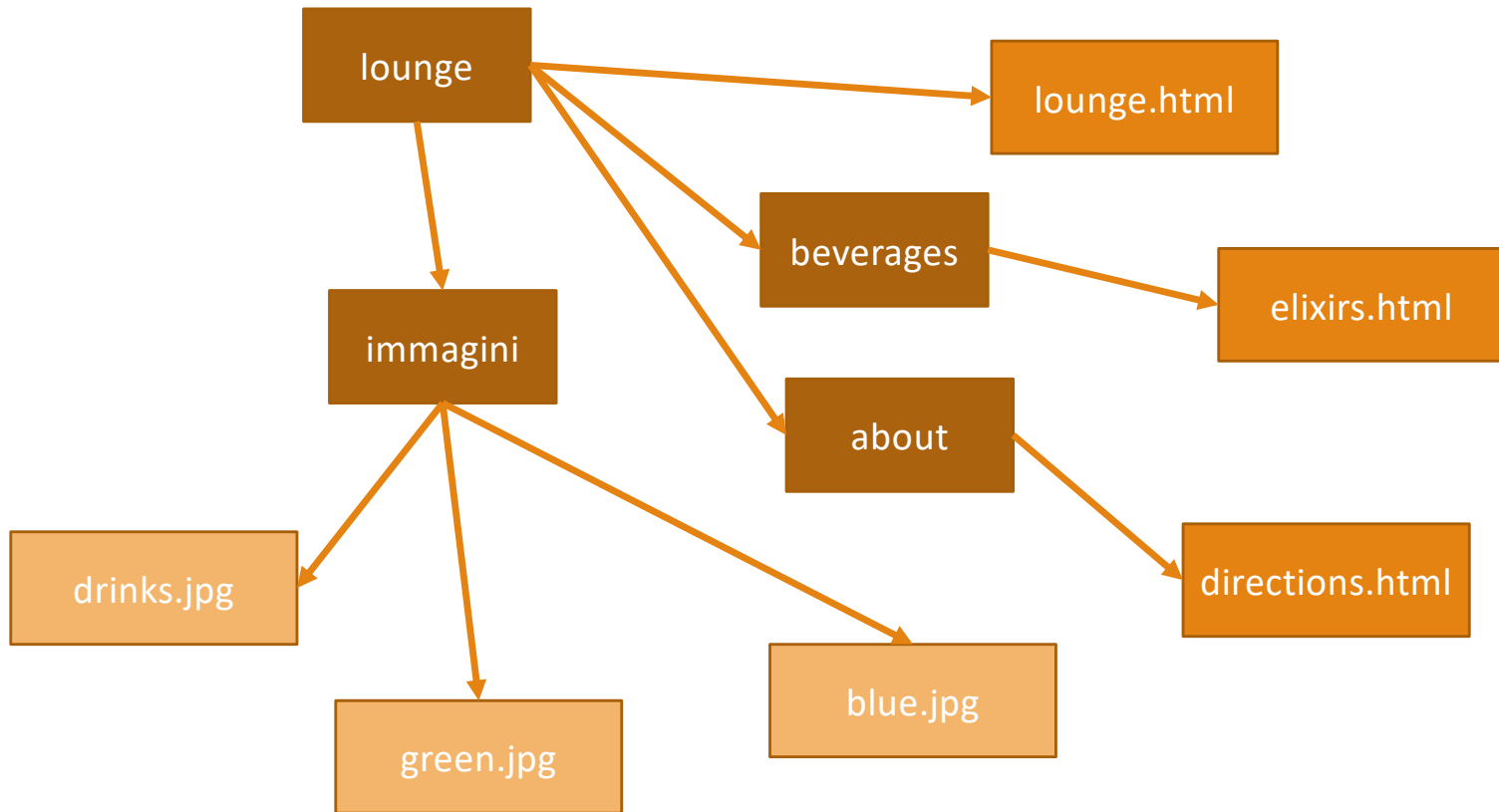
Here's where we add the HTML for the link to the elixirs.

To create links we use the <a> element; we'll take a look at how this element works in just a sec...

We need to add some text here to point customers to the new directions.

And here's where we add the link to the directions, again using an <a> element.

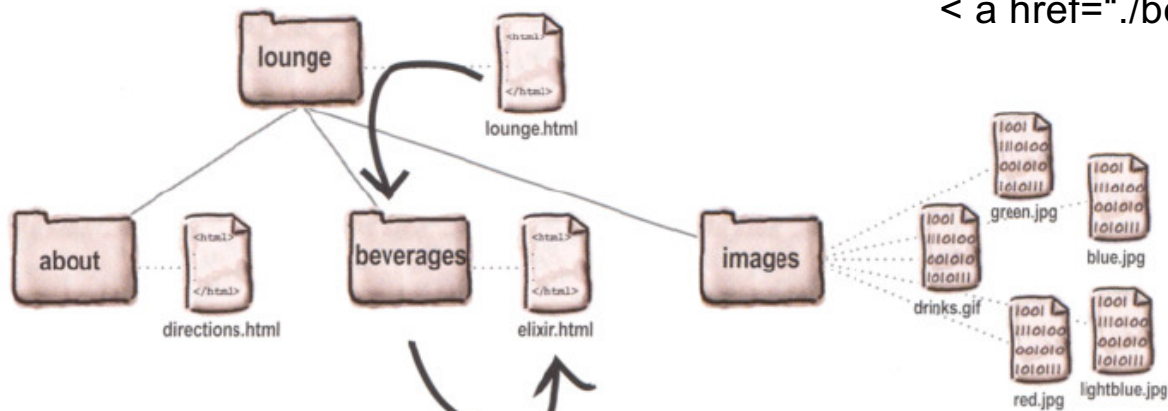
Ancora più ordine!



In questo caso organizziamo i file in più sottocartelle, come in Esercizio 4 sushi

Come cambiare link

First we need to go down into the "beverages" folder.

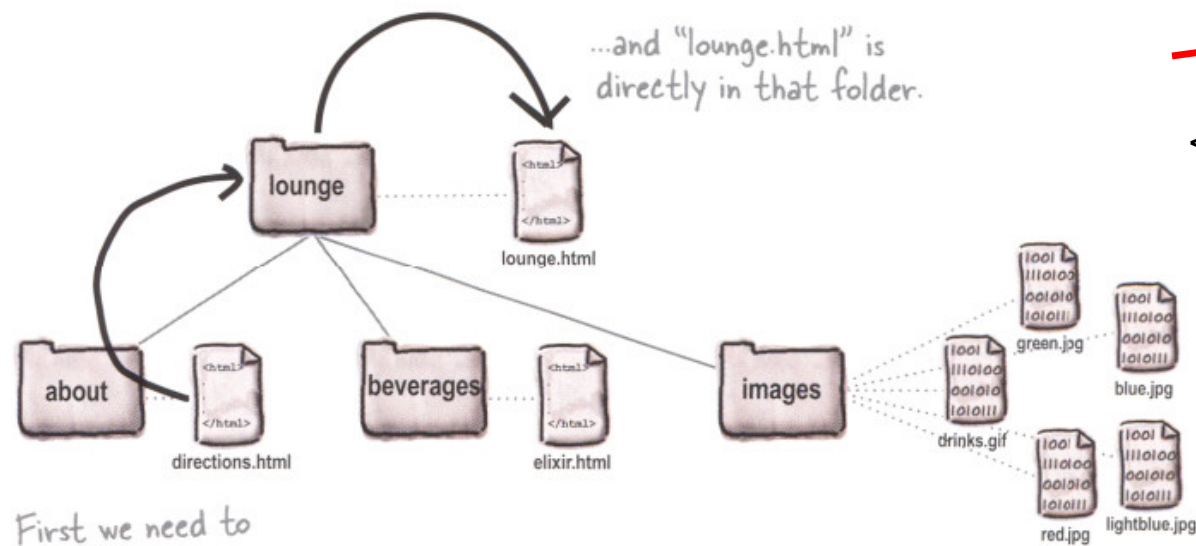


And "elixir.html" is directly in that folder.

~~< a href="elixir.html" > elixirs~~

< a href="./beverages/elixir.html" > elixirs

Per tornare alla home



~~< a href="lounge.html">lounge~~

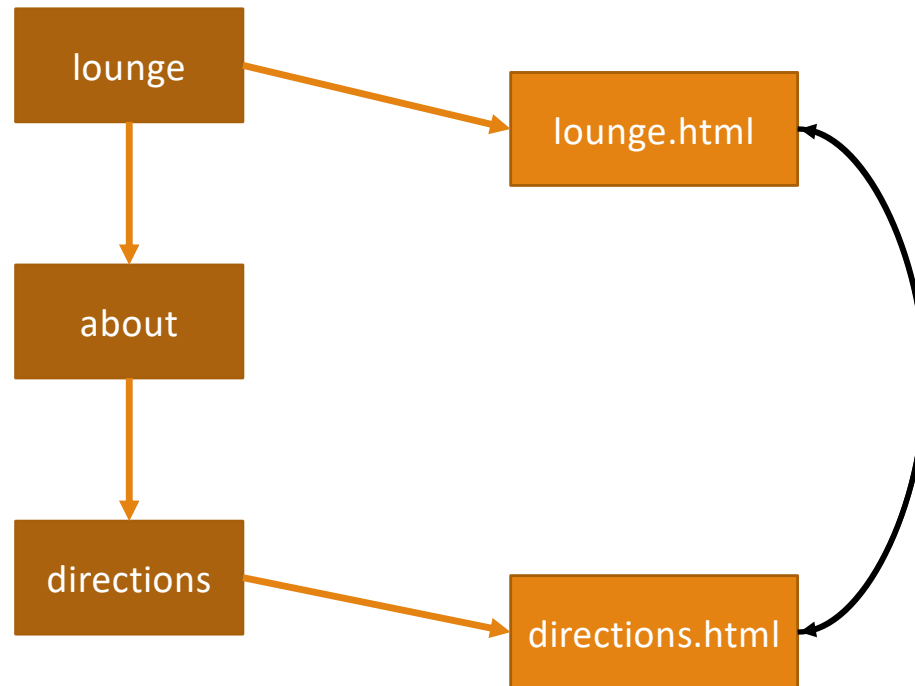
< a href=" ../lounge.html">lounge

./ significa "cerca nella directory dove si trova il file che contiene questo link"

../ significa "cerca nella directory superiore a dove si trova il file che contiene questo link"

Altro esempio

`< a href=“./about/directions/directions.html”>directions`



`< a href=“../../lounge.html”>lounge`



Commenti finali



Elementi block e Inline

Un elemento blocco inizia sempre su una nuova linea e occupa tutta la larghezza possibile a destra e sinistra.

Esempi sono `<h1>` (e gli altri titoli) e `<p>`

Gli elementi Inline non iniziano su una nuova linea e occupano solo lo spazio necessario.

Esempi sono `<a>` (per definire un link), `` (per un'immagine), e `` (utilizzato per delimitare parte di testo; poi vediamo degli esempi).

Quindi esistono ELEMENTI, che sono definiti da tag, in certi casi sia di apertura che di chiusura.

Span

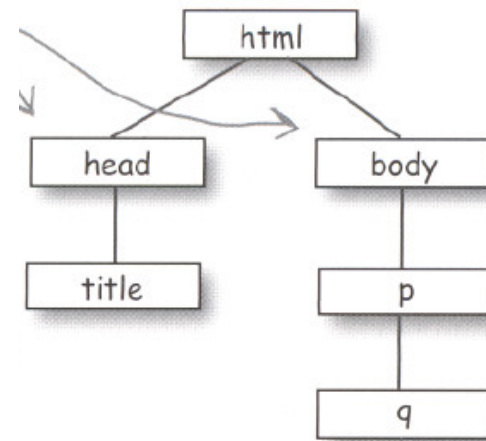
Il tag span è utilizzato per raggruppare elementi inline, anche semplicemente testo.

Di per sé non apporta nessun cambiamento di visualizzazione.

`<p>My mother has blue eyes, and blonde hair.</p>`

Annidamento (nesting)

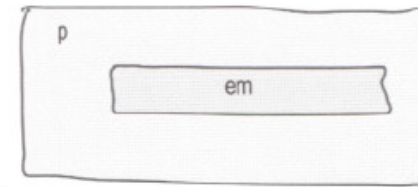
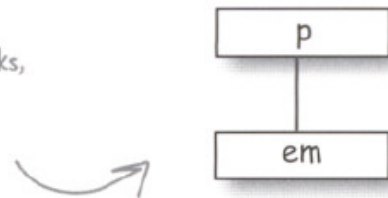
```
<html>
  <head>
    <title>Musings</title>
  </head>
  <body>
    <p>
      To quote Buckaroo,
      <q>The only reason
      for time is so
      that everything
      doesn't happen
      at once.</q>
    </p>
  </body>
</html>
```



Esempio corretto e sbagliato

`<p>I'm so going to blog this</p>`

Here's how this HTML looks,
 is nested inside <p>.

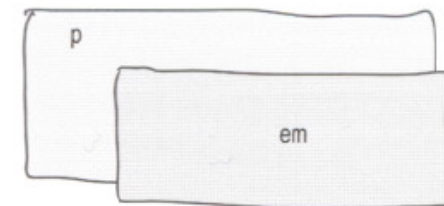


GOOD: here the element is nested inside the <p>.

looks more like this:

`<p>I'm so going to blog this</p>`

WRONG: the <p> tag ends before the tag! The element is supposed to be inside the <p> element.



BAD: here the element has leaked outside of the <p> element, which means it's not properly nested inside it.

RICORDARSI DI CHIUDERE I TAG!

Commenti

Si possono aggiungere commenti che non vengono visualizzati nel browser.

```
<!-- Write your comments here -->
```

Si possono scrivere commenti per ricordarsi di qualcosa di importante o per annotare il codice HTML utilizzato.

Un altro motivo è per commentare temporaneamente delle linee di codice, senza cancellarle e poi riscriverle nuovamente.

```
<!-- Do not display this at the moment
```

```
<a href="http://www.ciao.it">Sito ciao</a>
```

```
-->
```



Risorse e strumenti



Risorse

W3C School Tutorial

<http://www.w3schools.com/html/default.asp>

Quando non vi ricordate qualcosa, ad esempio il nome del tag, la sua sintassi, i suoi attributi, controllate subito lì

Altro tutorial, sia CSS che html, HTML Dog:

<http://htmldog.com/guides/>



Strumenti

Brackets, Mac, Windows

<http://brackets.io>

HTML-Kit 292 (versione free), Windows

<http://www.htmlkit.com/download/next/?download292=1&thankyou=1>

Coffee cup, Windows

<http://www.coffeecup.com/free-editor/>



W3C Validator

COME VALIDARE I DOCUMENTI RISPETTO AGLI STANDARD



Linguaggio e insieme caratteri

```
<!DOCTYPE html>
<html lang="it">
  <head>
    <meta charset="UTF-8">
    |   <title>CIAO</title>
  </head>
  <body>

    <h1>CIAO</h1>

    <p> La metà </p>

  </body>
</html>
```

HTML versione 5

Linguaggio, italiano

Set di caratteri, UTF8: si possono utilizzare anche le lettere accentate!

Utilizziamo questo schema per tutte le pagine che scriviamo: riportare i tag indicati dalle 3 frecce in tutte le pagine

Validiamo HTML5

All'indirizzo


https://validator.w3.org/#validate_by_upload

Validatore HTML (per passaggio di file)

Validate by **URI** **Validate by File Upload** Validate by **Direct Input**

Validate by File Upload

Upload a document for validation:

File: no file selected 

▼ **More Options**

Character Encoding (detect automatically) Only if missing

Document Type (detect automatically) Only if missing

List Messages Sequentially Group Error Messages by Type

Show Source Clean up Markup with HTML-Tidy

Show Outline Validate error pages Verbose Output

Passi da seguire

Per ognuna delle pagine:

1. Seguire la struttura di due slide fa, per ognuna delle pagine che compongono il sito
2. Aprire il validatore (indirizzo precedente) e passare la pagina html, sfogliando e caricandola
3. Se il risultato è una lista di errori, capire il primo errore della lista guardando nell'html alla linea segnalata, correggere, e rivalidare per essere sicuri che l'errore sia stato corretto
4. Ripetere il punto 3 fino a che non ci sono errori per ognuna delle pagine del sito