



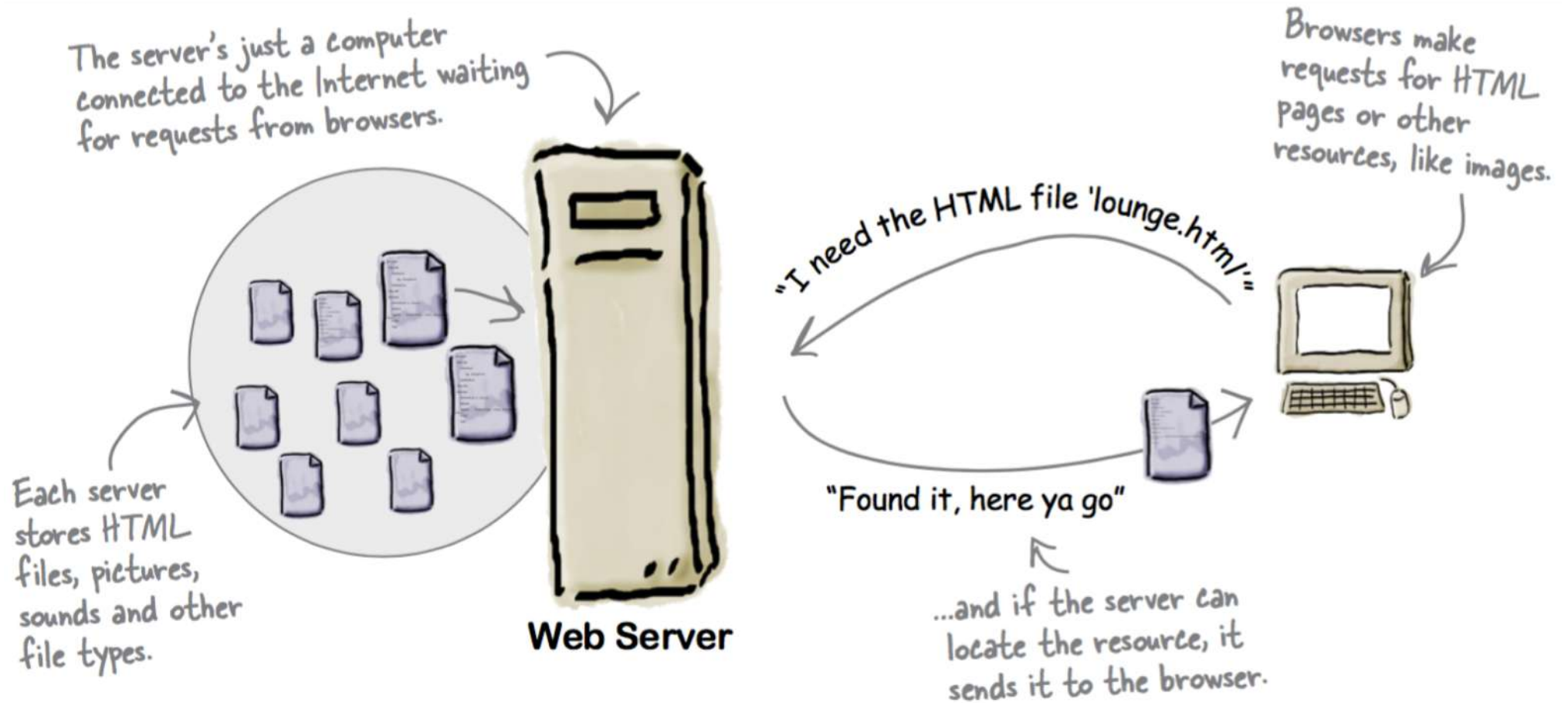
CORSO DI LAUREA IN INFORMATICA

# PROGRAMMAZIONE WEB

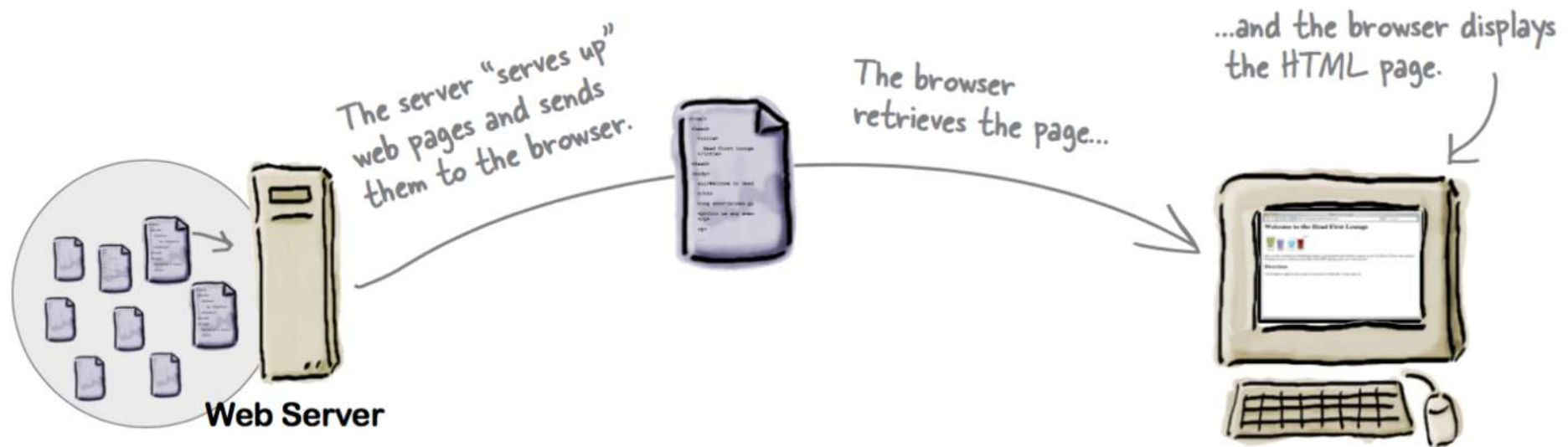
HTML - PART 1

a.a 2018-2019

# What does the web server do?



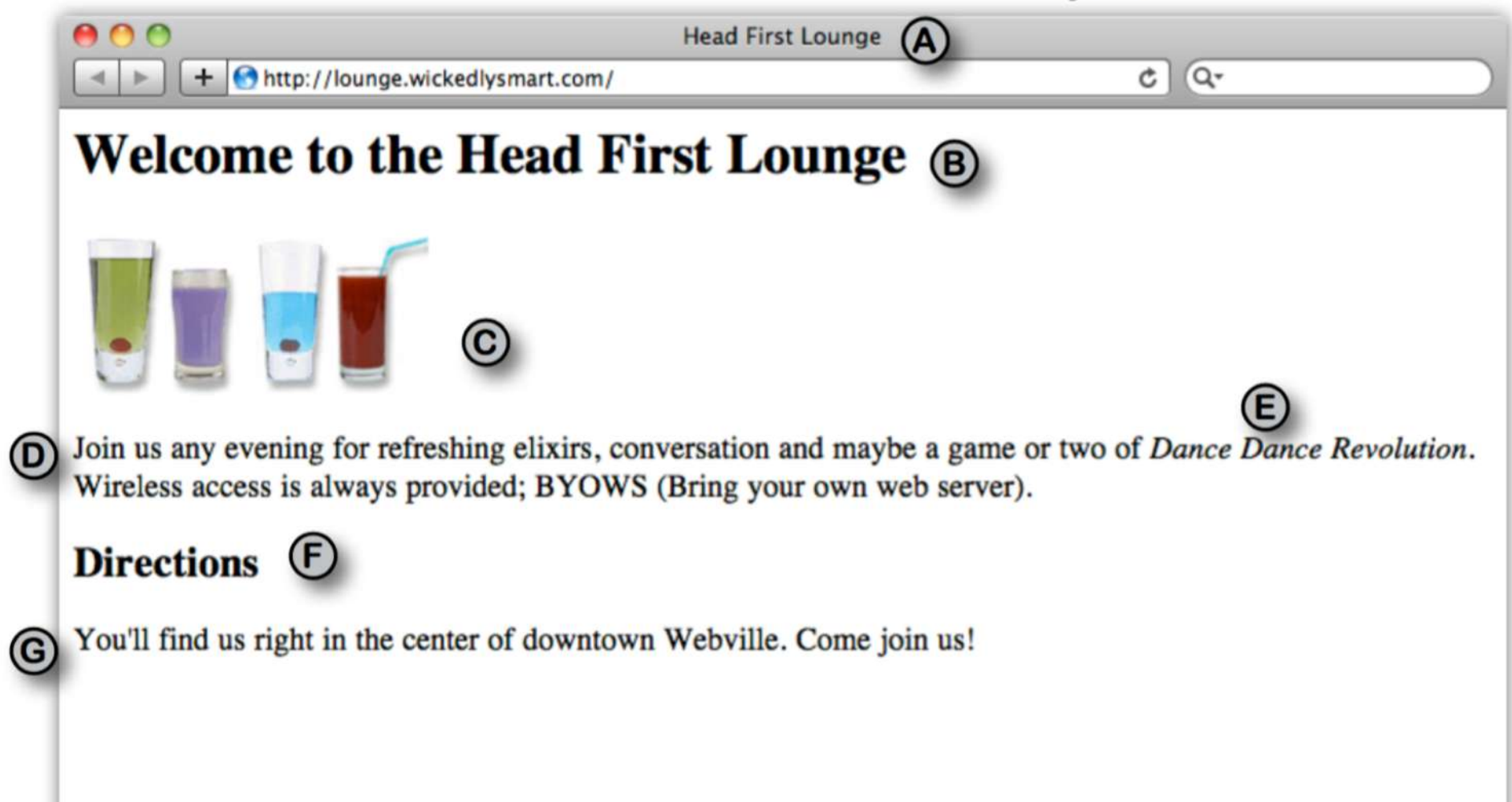
# What does the web browser do?



```
<html>
  <head>
    <title>Head First Lounge</title> ①
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1> ②
     ③
    <p>
      ④ Join us any evening for refreshing elixirs,
        conversation and maybe a game or
        two of <em>Dance Dance Revolution</em>. ⑤
        Wireless access is always provided;
        BYOWS (Bring your own web server).
    </p>
    <h2>Directions</h2> ⑥
    <p>
      ⑦ You'll find us right in the center of
        downtown Webville. Come join us!
    </p>
  </body>
</html>
```

Let's check out how the browser interprets the tags in the Head First Lounge

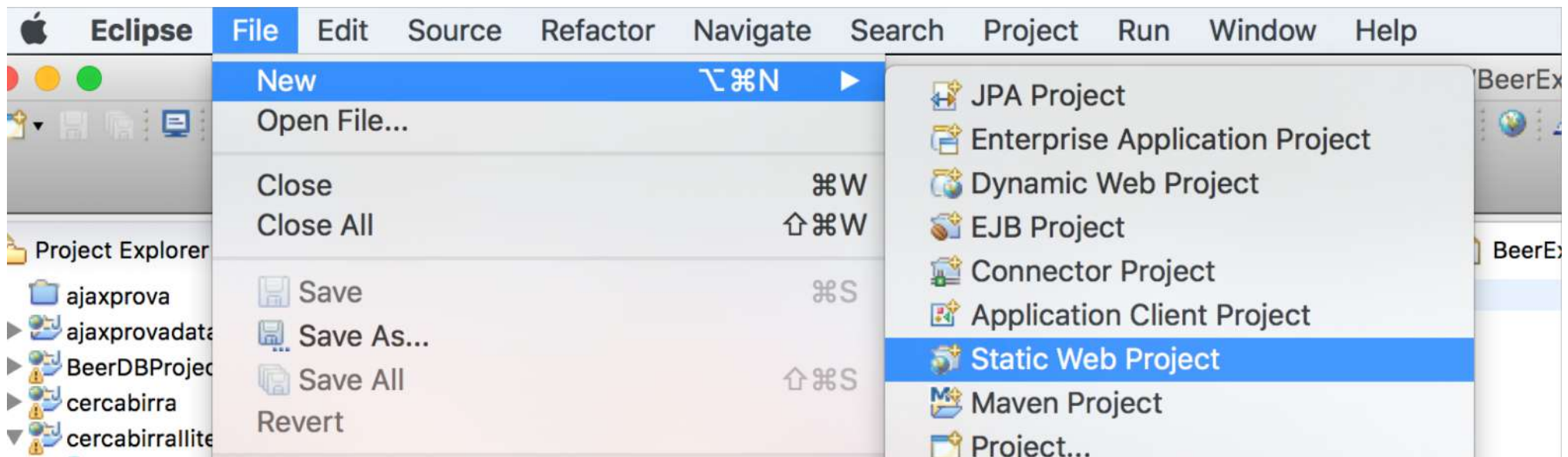
Notice how each tag in the HTML maps to what the browser displays.





# Creating an HTML file


- All HTML files are text files
- you have to create plain text
- *Attention to the text format!*
  - On MAC for the Textedit editor → Set Preferences for **Plain text**
  - On Windows → Notepad or Notepad++
- **We will use Eclipse!!!**



New Static Web Project

### Static Web Project

Create a Web project for simple content (HTML files) only.



Project name:

Project location

☒ Use default location

Location:

Target runtime


Configuration

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

Working sets

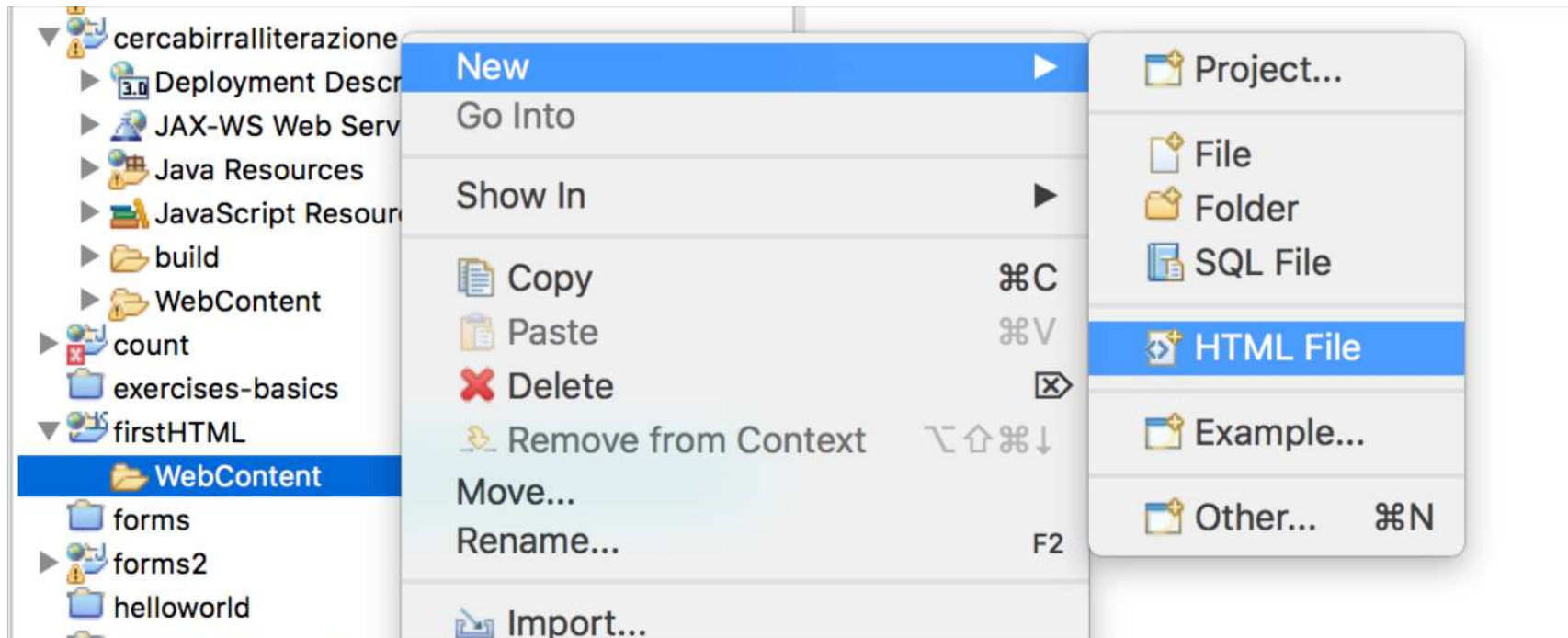
☐ Add project to working sets

Working sets:

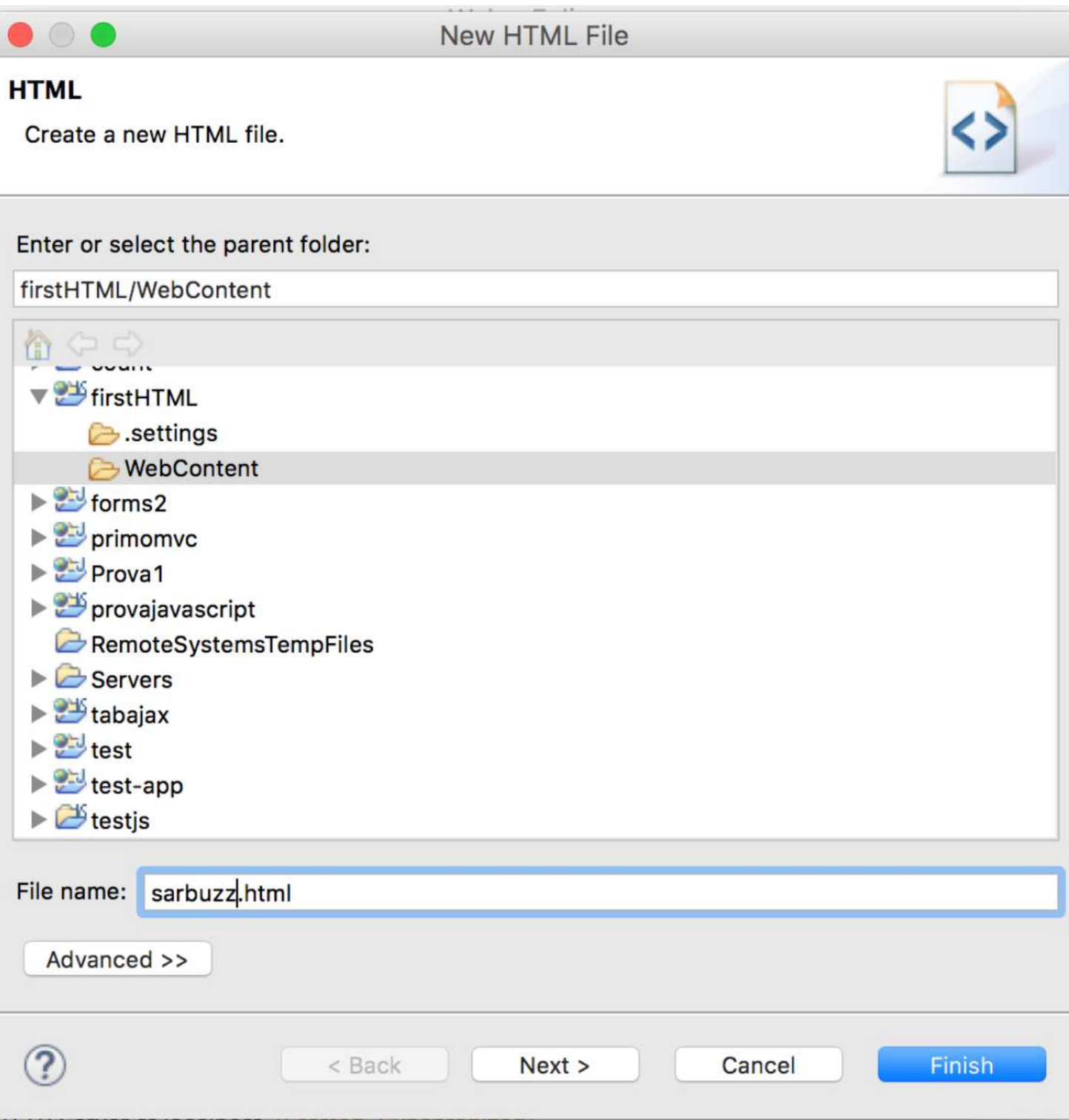


- Press Finish

- R-click on the folder web content of the project firstHTML







## Edit the file

### Starbuzz Coffee Beverages

House Blend, \$1.49

A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.

Mocha Cafe Latte, \$2.35

Espresso, steamed milk and chocolate syrup.

Cappuccino, \$1.89

A mixture of espresso, steamed milk and foam.

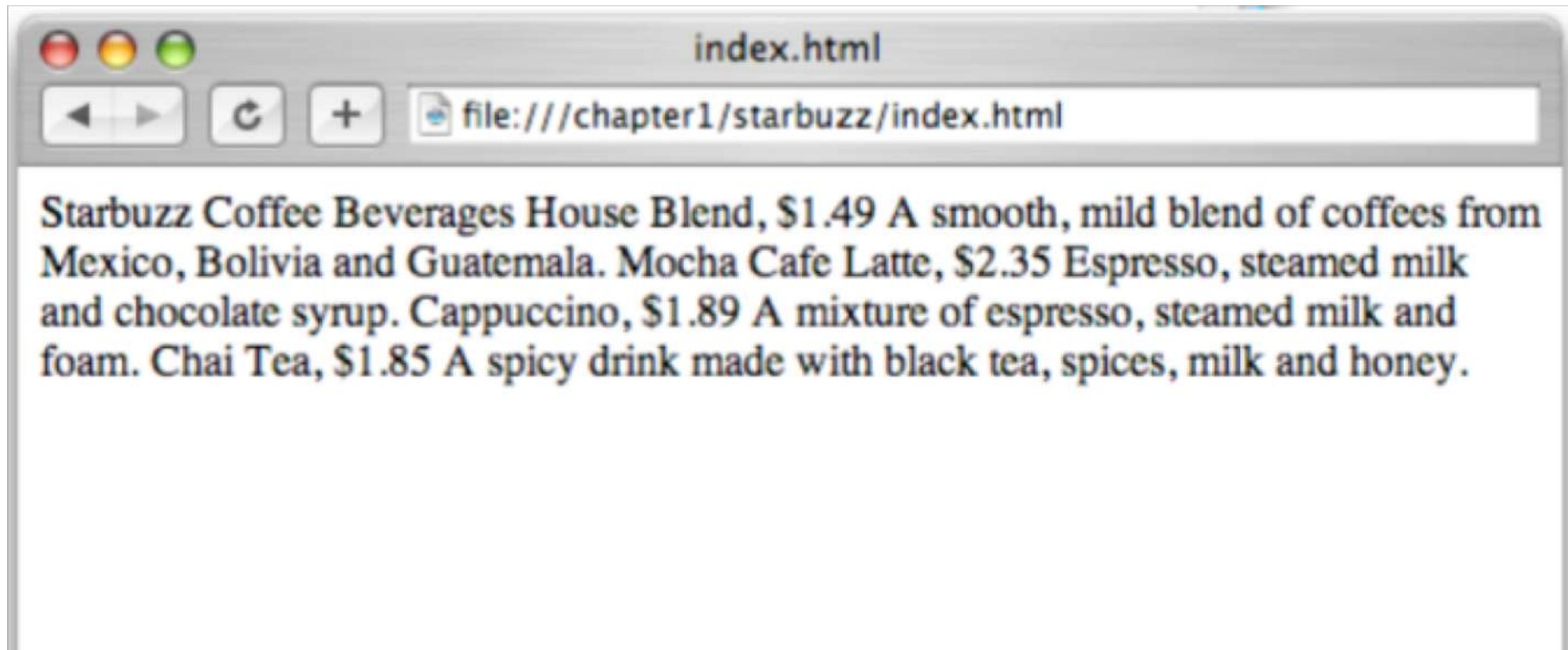
Chai Tea, \$1.85

A spicy drink made with black tea, spices, milk and honey.

- To see it in the browser: R-click on the file open with → Web browser

# Open the file in the browser and test your file

- Select the file out of Eclipse
- And double click on it
- *The result is poor!*



Use the <h1> and </h1> tags to mark headings. All the text in between is the actual content of the heading.

**<h1>Starbuzz Coffee Beverages</h1>**

**<h2>House Blend, \$1.49</h2>**

**<p>A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.</p>**

The <h2> and </h2> tags go around a subheading. Think of an <h2> heading as a subheading of an <h1> heading.

**<h2>Mocha Cafe Latte, \$2.35</h2>**

**<p>Espresso, steamed milk and chocolate syrup.</p>**

**<h2>Cappuccino, \$1.89</h2>**

**<p>A mixture of espresso, steamed milk and foam.</p>**

The <p> and </p> tags go around a block of text that is a paragraph. That can be one or many sentences.

**<h2>Chai Tea, \$1.85</h2>**

**<p>A spicy drink made with black tea, spices, milk and honey.</p>**

Notice that you don't have to put matching tags on the same line. You can put as much content as you like between them.

# You have to add...

- The head

First, surround your HTML with `<html>` & `</html>` tags. This tells the browser the content of the file is HTML.

```
<html>
```

```
<head>
```

```
<title>Starbuzz Coffee</title>
```

```
</head>
```

Next add `<head>` and `</head>` tags. The head contains information about your web page, like its title. For now, think about it this way: the head allows you to tell the browser things about the web page.

Go ahead and put a title inside the head. The title usually appears at the top of the browser window.

The head consists of the `<head>` & `</head>` tags and everything in between.

.....

- and the closure

```
</html>
```

Keep your head and body separate when writing HTML.





Test your file



# Add style to your page

- Add the **<style>** element to the head of the HTML
- Css uses a syntax very different from HTML

```
<html>
  <head>
    <title>Starbuzz Coffee</title>
    <style type="text/css">
      body {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 2px dotted black;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>
```

# Test again!

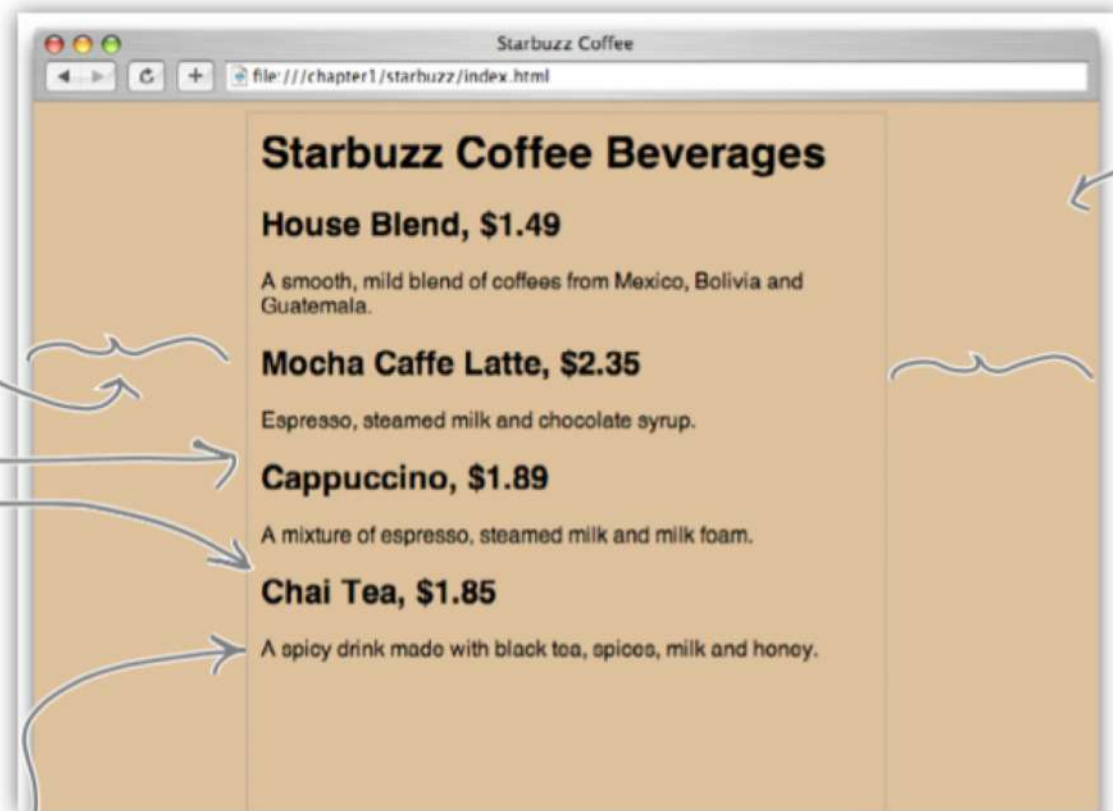
Background color is now tan.

Now we have margins around the content.

We've got a black dotted border around the content.

There's now some padding between the content and the border (on all sides).

We're using a different font for a cleaner look.

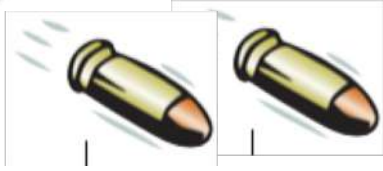
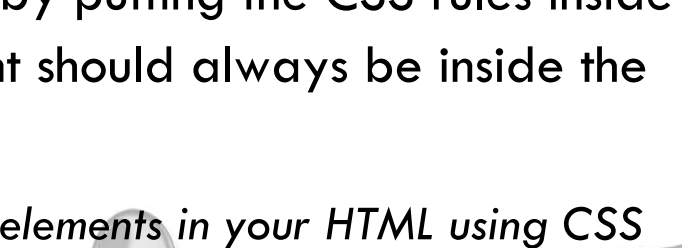


# Bullet point

- *HTML and CSS are the languages we use to create web pages!*
  - Web servers store and serve web pages, which are created from HTML and CSS. Browsers retrieve pages and render their content based on the HTML and CSS
  - HTML is an abbreviation for **HyperText Markup Language** and is used to structure your web page
  - CSS is an abbreviation for **Cascading Style Sheets**, and is used to control the presentation of your HTML
- Using HTML, we mark up content with **tags** to provide structure. We call matching tags, and their enclosed content, elements
- An element is composed of three parts: **an opening tag, content, and a closing tag**. There are a few elements, like `<img>`, that are an exception to this rule
- Opening tags can have attributes. We've seen one already "type" for the tag `<style>`



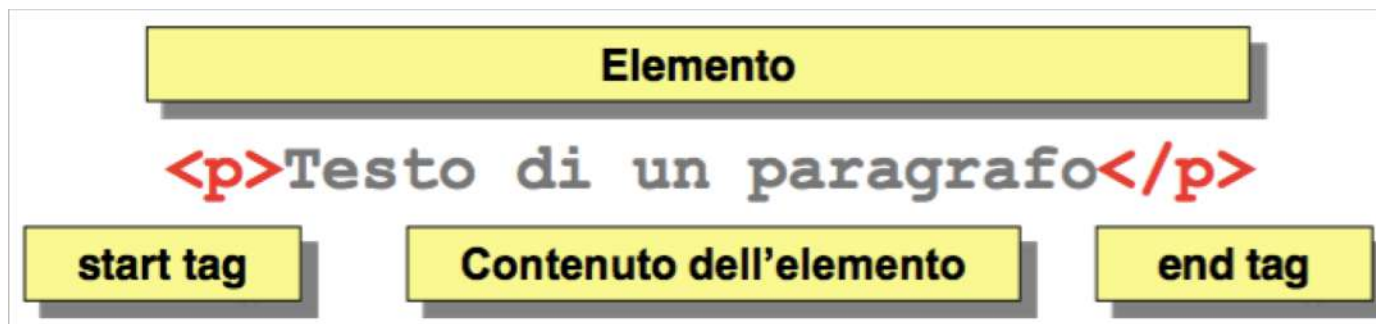
## Bullet point (2)

- 
- Closing tags have a “/” after the left angle bracket, in front of the tag name, to distinguish them as closing tags
  - Your pages should always have an **<html>** element along with a **<head>** element and a **<body>** element
  - Information about the web page goes into the **<head>** element
  - What you put into the **<body>** element is what you see in the browser
  - Most whitespace (tabs, returns, spaces) is ignored by the browser, but you can use it to make your HTML more readable (to you)
  - You can add CSS to an HTML web page by putting the CSS rules inside the **<style>** element. The **<style>** element should always be inside the **<head>** element
    - *You specify the style characteristics of the elements in your HTML using CSS*
- 



# Tag

- I tag HTML sono usati per definire il mark-up di elementi HTML
- Sono preceduti e seguiti rispettivamente da due caratteri “<” e “>” (parentesi angolari)
- Sono normalmente accoppiati; un esempio è dato da: <p> e </p>, detti rispettivamente start tag ed end tag
- Il testo tra start tag ed end tag è detto contenuto dell'elemento
- Un documento HTML contiene quindi elementi composti da testo semplice delimitato da tag



# Grammatica poco rigorosa

- HTML

- Ammette elementi senza chiusura come `<br>`
- I tag non sono case sensitive
- L'apertura e chiusura di tag annidati può essere "incrociata"

`<b><i>Testo corsivo grassetto</b></i>`

- Esistono però delle buone pratiche che è bene rispettare e che diventano un obbligo in una versione più rigorosa del linguaggio chiamata XHTML

- Chiudere sempre anche i tag singoli: `<br></br>` o in forma sintetica `<br/>`
- Tag in minuscolo
- Apertura e chiusura senza incroci (in teoria non ammessi ma tollerati)

`<b><i>...</i></b>`

# Entity

- HTML definisce un certo numero di entità (entity) per rappresentare i caratteri speciali (*escape character*) senza incorrere in problemi di codifica:
  - Caratteri riservati a HTML (<, >, &, “, ecc.)
  - Caratteri non presenti nell'ASCII a 7 bit

Result	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;

💡 Entity names are case sensitive.

- La lista completa di escape character HTML:

<http://www.theukwebdesigncompany.com/articles/entity-escape-characters.php>

# Attributi

- Un elemento può essere dettagliato mediante attributi
  - Gli attributi sono coppie **“nome = valore”** contenute nello start tag con una sintassi di questo tipo

**<tag attrib1='valore1' attrib2='valore2'>**

- I valori sono racchiusi da apici singoli o doppi, es:

**<input type='submit' value="NAME">Ok</input>**

- Gli apici possono essere omessi se il valore non contiene spazi
- I colori vengono espressi con un nome o in formato RGB con la sintassi **#RRGGBB**

	Red = "#FF0000"		Black = "#000000"
	Blue = "#0000FF"		Yellow = "#FFFF00"

## Attributes (2)

- **Attributes** give you a way to specify additional information about an element

`<style type="text/css">`

The type attribute specifies which style language we're using, in this case CSS.

`<a href="irule.html">`

The href attribute tells us the destination of a hyperlink.

``

The src attribute specifies the filename of the picture an img tag displays.

- If `<car>` was an element, then you'd naturally want to write some markup like this:

```
<car>My Red Mini</car>
```

```
<car make="Mini" model="Cooper" convertible="no">My Red Mini</car>
```

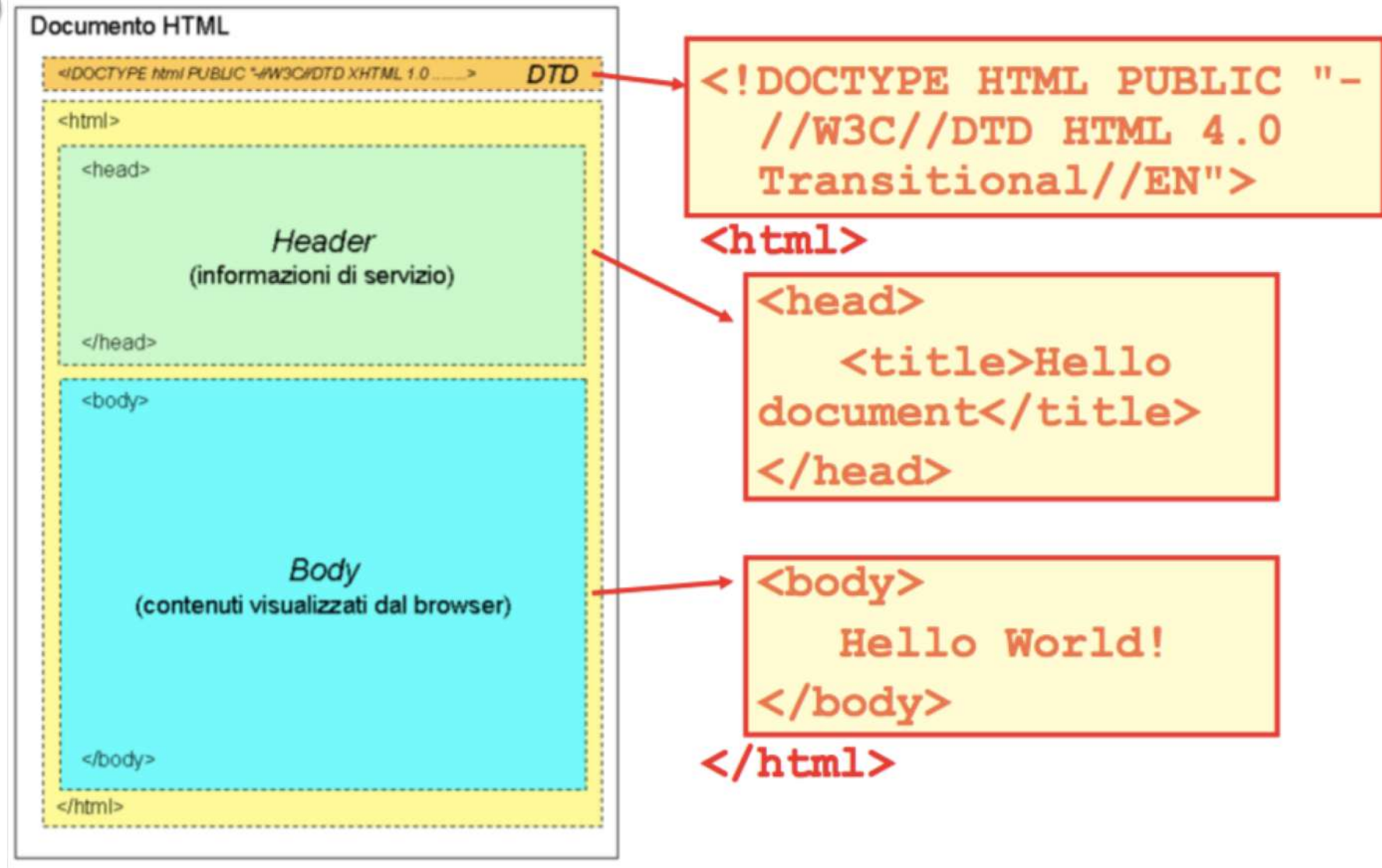


# Commenti

- È possibile inserire commenti in qualunque punto all'interno di una pagina HTML con la seguente sintassi:

**<!-- Questo è un testo di commento -->**

# Struttura base di un documento HTML



<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>



Only the <body> area (the white area) is displayed by the browser.

# DTD

- Il primo elemento di un documento HTML è la definizione del tipo di documento (**Document Type Definition** o DTD)
- Serve al browser per identificare le regole di interpretazione e visualizzazione da applicare al documento, es:
  - **<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"**  
**<http://www.w3.org/TR/html4/loose.dtd>>**
- È costituita da diverse parti:
  - **HTML** il tipo di linguaggio utilizzato è l'HTML
  - **PUBLIC** il documento è pubblico
  - - le specifiche non sono registrate all'ISO (altrimenti +)
  - **W3C** ente che ha rilasciato le specifiche
  - **DTD HTML 4.01 Transitional**: versione di HTML
  - **EN** la lingua con cui è scritta il DTD è l'inglese
  - **<http://...>** URL delle specifiche

# HTML 5 DOCTYPE

- The old HTML 4.01 version of the doctype:
  - **<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">**
- The new doctype for HTML5 is simply:
  - **<!doctype html>**
- ***It is simple to remember it!***



# Header

- È identificato dal tag **<head>**
  - Contiene elementi non visualizzati dal browser (informazioni di servizio)
- **<title>** titolo della pagina (viene mostrato nella testata della finestra principale del browser)
- **<meta>** metadati informazioni utili ad applicazioni esterne (es. motori di ricerca) o al browser (es. lingua, codifica dei caratteri utile per la visualizzazione di alfabeti non latini)
- **<base>** definisce come vengono gestiti i riferimenti relativi nei link
- **<link>** collegamenti verso file esterni: CSS, script, icone visualizzabili nella barra degli indirizzi del browser
- **<script>** codice eseguibile utilizzato dal documento
- **<style>** informazioni di stile (CSS locali)

## Element <meta>

- Gli elementi di tipo **<meta>** sono caratterizzati da una serie di attributi
- Esistono due tipi di elementi meta, distinguibili dal primo attributo: **http-equiv** o **name**
  - Gli elementi di tipo *http-equiv* danno informazioni al browser su come gestire la pagina
  - Hanno una struttura di questo tipo:  
**<meta http-equiv=*nome* content=*valore*>**
  - Gli elementi di tipo *name* forniscono informazioni utili ma non critiche
  - Hanno una struttura di questo tipo:  
**<meta nome=*nome* content=*valore*>**

# Element `<meta>`

- Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

- Define a description of your web page:

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

- Define the character set used:

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

- Define the author of a page:

```
<meta name="author" content="Tim Berners-Lee">
```

- Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

# Adding a meta tag to specify the character encoding

The charset attribute is where we specify the character encoding.

The value of the charset attribute is the type of character encoding we're using.

**<meta charset="utf-8">**

"utf-8" is an encoding in the Unicode family of encoding of several). "utf-8" is the we use for web pages.

## Adding a meta tag to specify the character encoding (2)

```
<!doctype html>
```

```
<html>
```

```
<head>
```

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

```
<title>Head First Lounge</title>
```

```
</head>
```

Here's the `<meta>` tag. We've added it to the `<head>` element above the `<title>` element.

Add this line above any other elements in the `<head>` element.

# Other Elements

- The **<base>** element specifies the base URL and base target for all relative URLs in a page

```
<base href="http://www.w3schools.com/images/" target="_blank">
```

- The **<link>** element defines the page relationship to an external resource

- The **<link>** element is most often used to link to style sheets

```
<link rel="stylesheet" href="mystyle.css">
```

```
<link rel="icon" href="favicon.png">
```

- The **<script>** element is used to define client-side JavaScripts

```
<script>
```

```
function myFunction {  
    document.getElementById("demo").innerHTML = "Hello JavaScript!";  
}
```

```
</script>
```

```
<script src="library.js"></script>
```

- **Inside the <style>** element you specify how HTML elements should render in a browser

- Each HTML document can contain multiple **<style>** tags.

```
<style>
```

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

```
</style>
```



# <html>, <body> and language

- The **<html>** tag tells the browser that this is an HTML document
- The **<html>** tag represents the root of an HTML document
- The **<body>** tag defines the document's body
- The **<body>** element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.
- In HTML5, the lang attribute can be used on **any** HTML element

## **<element lang="language\_code">**

- *language\_code* = Specifies the language code for the element's content (it, en, fr, ...)

# Attributes of <body>

Attribute	Value	Description
<u>alink</u>	<i>color</i>	<b>Not supported in HTML5.</b> Specifies the color of an active link in a document
<u>background</u>	<i>URL</i>	<b>Not supported in HTML5.</b> Specifies a background image for a document
<u>bgcolor</u>	<i>color</i>	<b>Not supported in HTML5.</b> Specifies the background color of a document
<u>link</u>	<i>color</i>	<b>Not supported in HTML5.</b> Specifies the color of unvisited links in a document
<u>text</u>	<i>color</i>	<b>Not supported in HTML5.</b> Specifies the color of the text in a document
<u>vlink</u>	<i>color</i>	<b>Not supported in HTML5.</b> Specifies the color of visited links in a document

# Tipi di elementi del body

- *Intestazioni*: titoli organizzati in gerarchia
- *Strutture di testo*: paragrafi, testo indentato, ecc.
- *Aspetto del testo*: grassetto, corsivo, ecc.
- *Elenchi e liste*: numerati, puntati
- *Tabelle*
- *Form (moduli elettronici)*: campi di inserimento, checkbox e radio button, menu a tendina, bottoni, ecc.
- *Collegamenti ipertestuali e ancore*
- *Immagini e contenuti multimediali*: audio, video, animazioni, ecc.
- *Contenuti interattivi*: script, applicazioni esterne

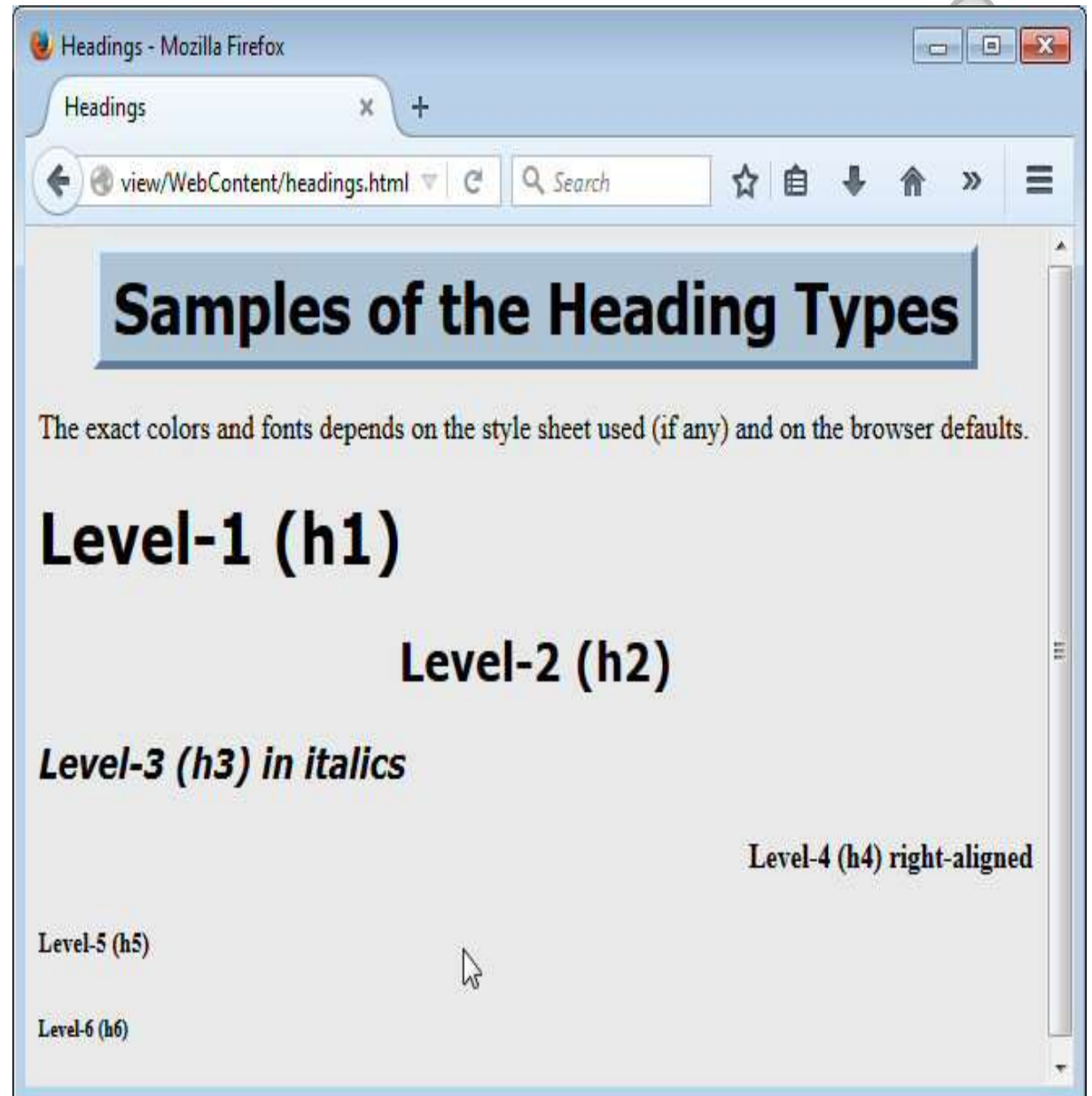
# Elementi blocco, elementi inline e liste

- Dal punto di vista del layout della pagina gli elementi HTML si dividono in 3 grandi categorie:
  - Elementi “**block-level**”: costituiscono un blocco attorno a sé, e di conseguenza vanno a capo (paragrafi, tabelle, form...)
  - Elementi “**inline**”: non vanno a capo e possono essere integrati nel testo (link, immagini,...)
  - Liste: numerate, puntate
- Regole di composizione:
  1. Un elemento block-level può contenere altri elementi dello stesso tipo o di tipo inline
  2. Un elemento inline può contenere solo altri elementi inline

# Headings: Example

```
<!doctype html>
<html>
  <head>
    <title>headings</title>
    <link href="css/styles.css" rel="stylesheet"/>
  </head>
  <body>
    ...
    <h1>level-1 (h1)</h1>
    <h2 align="center">level-2 (h2) centered</h2>
    <h3><i>level-3 (h3) in italics</i></h3>
    <h4 align="right">level-4 (h4) right-aligned</h4>
    <h5>level-5 (h5)</h5>
    <h6>level-6 (h6)</h6>
  </body>
</html>
```

# Headings: Result





## <p> the basic paragraph

- Attribute: align
  - left (default), right, center (same as headings)
    - Again, often better to do alignment from CSS:  
`<p class="some-css-name-that-gives-alignment">...</p>`
- Behavior
  - Whitespace ignored (use `<br/>` if you want line breaks)
  - Consecutive `<p></p>`'s do not yield multiple blank lines
    - But multiple `<br/>`'s do
  - End tags not officially required in HTML 5, but you should use them so that what you see corresponds directly to how the browser represents the page internally
    - This makes it easier to visualize how to use Javascript/JQuery to manipulate the page

## Attribute align

`align = left`

`align = center`

`align = right`

`align = justify`

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume

## Use of <p> tag

Preferred Style	Technically Legal
<pre>&lt;body&gt; &lt;p&gt; Paragraph 1 &lt;/p&gt; &lt;p&gt; Paragraph 2 &lt;/p&gt; &lt;p&gt; Paragraph 3 &lt;/p&gt; &lt;/body&gt;</pre>	<pre>&lt;body&gt; Paragraph 1 &lt;p&gt; Paragraph 2 &lt;p&gt; Paragraph 3 &lt;/body&gt;</pre>

# <pre> preformatted paragraphs

- Characteristics
  - Line wrapping disabled
  - Whitespace no longer ignored
  - Fixed-width font used
- Problem: special characters

<pre>

```
If (a<b) {  
    dothis();  
} else {  
    dothat();  
}
```

</pre>

Desired Character	Text Required
<	&lt;
>	&gt;
&	&amp;
"	&quot;
Non-breaking space	&nbsp;

# <img> images

- In HTML the <img> tag has no end tag

****

Attribute	Value	Description
<u>align</u>	top bottom middle left right	<b>Not supported in HTML5.</b> Specifies the alignment of an image according to surrounding elements
<u>alt</u>	<i>text</i>	Specifies an alternate text for an image
<u>border</u>	<i>pixels</i>	<b>Not supported in HTML5.</b> Specifies the width of the border around an image
<u>height</u>	<i>pixels</i>	Specifies the height of an image
<u>src</u>	<i>URL</i>	Specifies the URL of an image
<u>usemap</u>	<i>#mapname</i>	Specifies an image as a client-side image-map
<u>width</u>	<i>pixels</i>	Specifies the width of an image

Here's the new  
and improved  
page.

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

Head First Lounge

file:///chapter2/completelounge/lounge.html

# Welcome to the New and Improved Head First Lounge



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

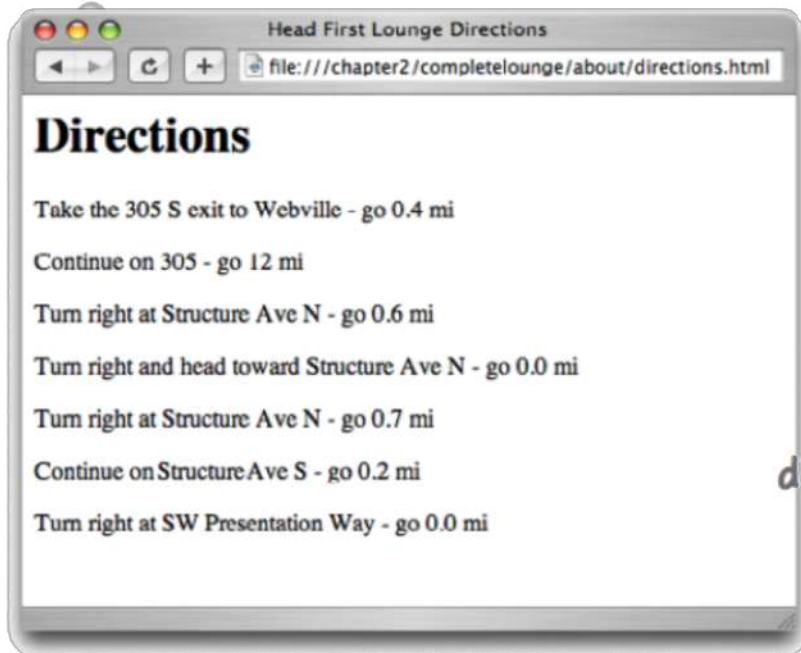
Join us any evening for refreshing [elixirs](#), conversation and maybe a game or two of *Dance Dance Revolution*. Wireless access is always provided; BYOWS (Bring Your Own Web Server).

## Directions

You'll find us right in the center of downtown Webville. If you need help finding us, check out our [detailed directions](#). Come join us!

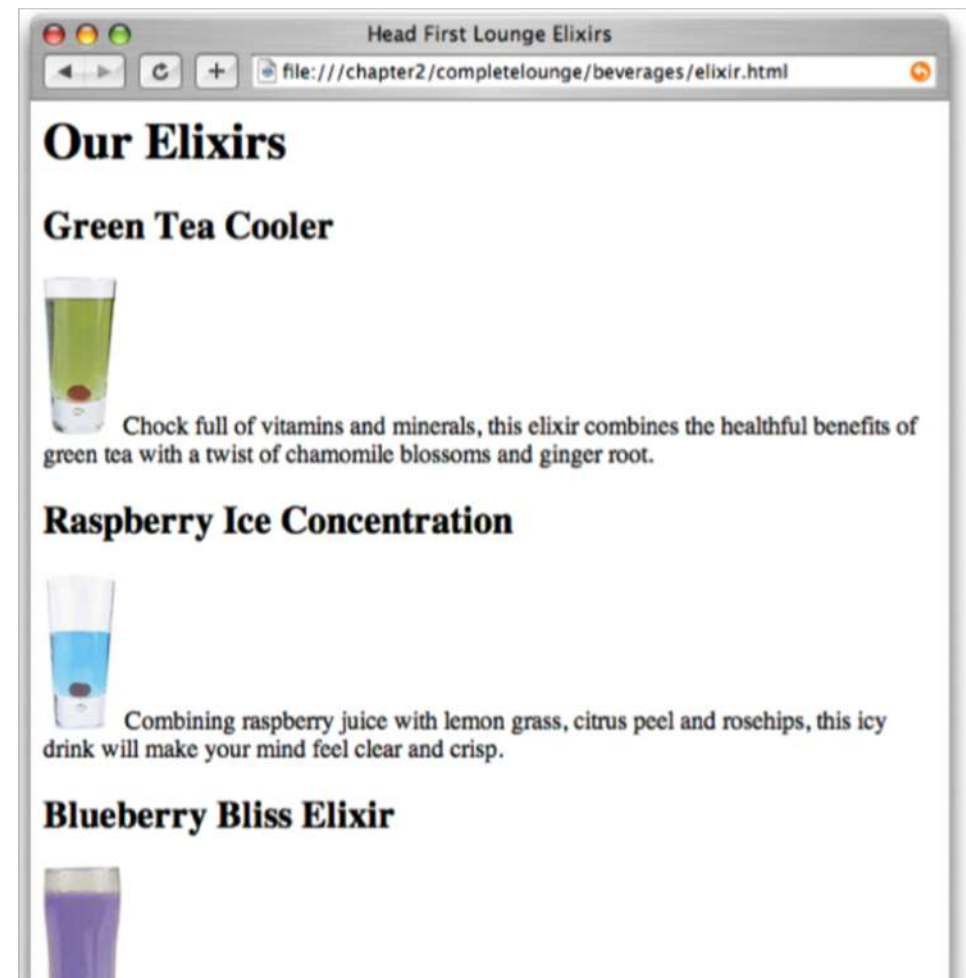


# directions.html and elixir.html



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

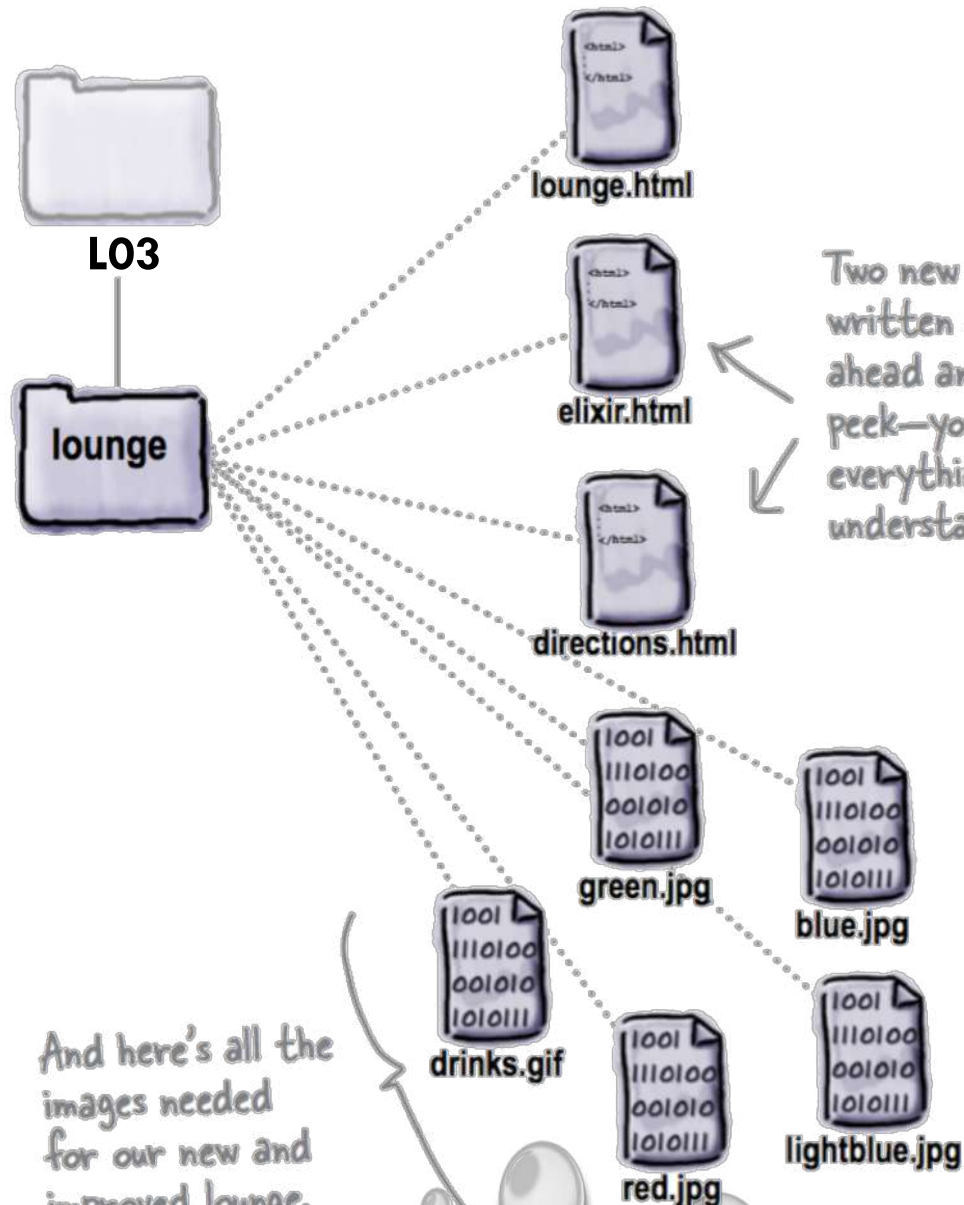
←  
directions.html



- Download “L03 HTML code”
- Copy it under web content folder of your static project...

You'll find the lounge directory here in your source files.

All the lounge files are in this folder.



Two new files, already written for you. Go ahead and take a peek—you already know everything you need to understand them.

And here's all the images needed for our new and improved lounge.

```
<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 [elixirs](elixir.html),  
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 [detailed directions](directions.html).  
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.

# Testing

- Save the file “lounge.html” and open it in your browser.
  1. Click on the [elixir](#) link and the new elixir page will display.
  2. Click on the browser’s back button and “lounge.html” should be displayed again.
  3. Click on the [directions](#) link and the new directions page will display.
- The `<a>` element is used to define a reference to another page

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

↑  
The href attribute is  
how you specify the  
destination of the link.

For this link, the browser will display the text “elixirs” that, when clicked, will take the user to the “elixir.html” page.

## Testing (2)

Wow, no difference. Well, we didn't really expect any because all the doctype does is let the browser know for sure you're using HTML5.





# Hypertext Links

- Usage

- Links can contain images and other inline elements (i.e., `<a href="..."> ... </a>`)

- Link to absolute url

- Use a complete URL beginning with **http://**

Javascript is discussed in

`<a href="http://host/path/chapter2.html">chapter 2</a>.`

- Link to relative url

- Use a filename or relative path to filename

- If no slash, interpreted with respect to location of current file

Javascript is discussed in `<a href="chapter2.html">chapter 2</a>.`



# Hypertext Links (Continued)

- Link to section (fragment)
  - Use a section name (see below) preceded by #  
Images are discussed in  
`<a href="#section2">section 2</a>.`
- Link to section in URL
  - Use absolute or relative URL, then #, then section name  
Images are discussed in  
`<a href="chapter1.html#section2">`  
Section 2 of chapter 1</a>.
- Naming a section
  - Use the id attribute and do not include the pound sign  
`<h2 id="section2">images</h2>`

# Link in un documento (ancora)

```
<p>  
  <a href="#section1">  
    Introduzione</a><br>  
  <a href="#section2">  
    Concetti di base</a><br>  
  <a href="#section2.1">  
    Definizione del problema</a><br>  
    ...  
</p>  
  
<h2><a name="section1">  
  Introduzione</a></h2>  
  ...sezione 1...  
<h2><a name="section2">  
  Concetti di base</a></h2>  
  ...sezione 2...  
<h3><a name="section2.1">  
  Definizione del problema</a></h3>  
  ...sezione 2.1...
```

## Sommario

[Introduzione](#)  
[Concetti di base](#)  
[Definizione del problema](#)  
...

### Introduzione

...sezione 1...

### Concetti di base

...sezione 2...

### Definizione del problema

...sezione 2.1...

# Link impliciti

- Si può esprimere un'ancora di destinazione in forma "implicita", cioè senza utilizzare il tag `<a>`
  - È sufficiente assegnare l'attributo **ID** a un qualunque elemento della pagina

```
<p>  
  <a href="#section1">  
    Introduzione</a><br>  
  <a href="#section2">  
    Concetti di base</a><br>  
  <a href="#section2.1">  
    Definizione del problema</a><br>  
  ...  
</p>  
  
<h2 id="section1">  
  Introduzione</h2>  
  ...sezione 1...  
  
<h2 id="section2">  
  Concetti di base</h2>  
  ...sezione 2...  
  
<h3 id="section2.1">  
  Definizione del problema</h3>  
  ...sezione 2.1...
```



## Sommario

[Introduzione](#)  
[Concetti di base](#)  
[Definizione del problema](#)  
...

### Introduzione

...sezione 1...

### Concetti di base

...sezione 2...

### Definizione del problema

...sezione 2.1...

# Interpreting Relative URLs

- Urls starting with **http://** (absolute urls)
  - Interpreted independently of the URL of current page
- Urls starting with **.** (or anything other than the other options)
  - Interpreted with respect to the “directory” of current file
    - Really with respect to the rightmost/ in the URL, since Urls don’t necessarily refer to real folders or files
- Urls starting with **..**
  - Interpreted with respect to the **parent** directory of current file
    - Again, based on the URL browser sees, not necessarily real location of resource
- Urls starting with **/**
  - Interpreted with respect to the hostname of the URL

# Interpreting Relative URLs: Example

- Assume URL of current page is

*`http://somehost.com/a/b/c/d/e/f.html`*

Link	Result
<code>&lt;a href="http://otherhost.com/a/b.html"&gt;...&lt;/a&gt;</code>	<code>http://otherhost.com/a/b.html</code>
<code>&lt;a href="./g.html"&gt;...&lt;/a&gt;</code> <u>or</u> <code>&lt;a href="g.html"&gt;...&lt;/a&gt;</code>	<code>http://somehost.com/a/b/c/d/e/g.html</code>
<code>&lt;a href="../h.html"&gt;...&lt;/a&gt;</code>	<code>http://somehost.com/a/b/c/d/h.html</code>
<code>&lt;a href="/i.html"&gt;...&lt;/a&gt;</code>	<code>http://somehost.com/i.html</code>

- **Note:** In addition to `<a href...>`, these URLs can apply to `img`, `form`, `script`, `style`, and a few other elements

# The target attribute

- The **target** attribute specifies where to open the linked document
  - `<a href="http://www.w3schools.com/" target="_blank">Visit W3Schools!</a>`

Target Value	Description
<code>_blank</code>	Opens the linked document in a new window or tab
<code>_self</code>	Opens the linked document in the same frame as it was clicked (this is default)
<code>_parent</code>	Opens the linked document in the parent frame
<code>_top</code>	Opens the linked document in the full body of the window
<code>framename</code>	Opens the linked document in a named frame



# Formatting elements

- Formatting elements were designed to display special types of text:

- Bold text

`<b>`This text is bold.`</b>`

- Important text

`<strong>`This text is strong.`</strong>`

- Italic text

`<i>`This text is italic.`</i>`

- Emphasized text

`<em>`This text is emphasized.`</em>`

- Marked text

`<h2>`HTML `<mark>`Marked`</mark>` Formatting`</h2>`

- Small text

`<h2>`HTML `<small>`Small`</small>` Formatting`</h2>`

- Deleted text

`<p>`My favorite color is `<del>`blue`</del>` red.`</p>`

- Inserted text

`<p>`My favorite `<ins>`color`</ins>` is red.`</p>`

- Subscripts

`<p>`This is `<sub>`subscripted`</sub>` text.`</p>`

- Superscripts

`<p>`This is `<sup>`superscripted`</sup>` text.`</p>`

**This text is bold.**

**This text is strong.**

*This text is italic.*

*This text is emphasized.*

**HTML **Marked** Formatting**

**HTML Small Formatting**

My favorite color is ~~blue~~ red.

My favorite color is red.

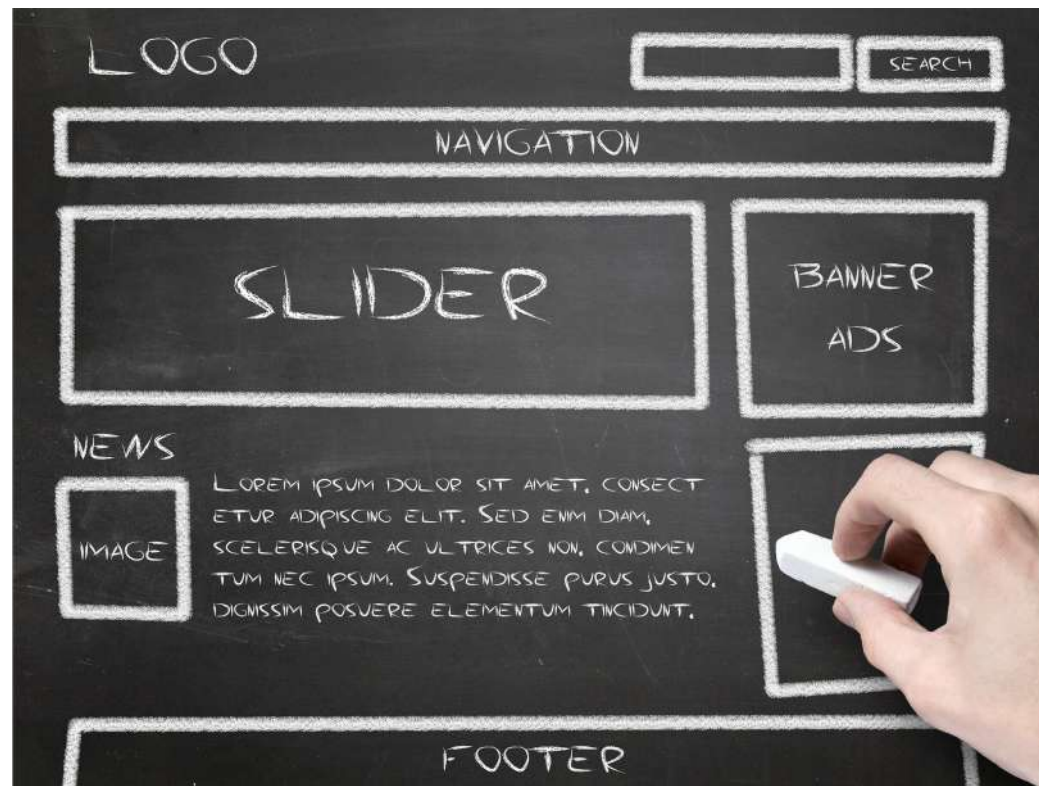
This is subscripted text.

This is superscripted text.



# Website structure

- *It's a good idea to organize your website files early on in the process of building your site, so you don't have to change a bunch of paths later when the website grows...*



# Exercise: Website construction

- **Example: a journal website (L03 code →journal)**
  1. First, you're going to create a rough sketch of the journal that is the basis for your page design
  2. Next, you'll use the basic building blocks of HTML (<h1>, <h2>, <h3>, <p>, and so on) to translate your sketch into an outline (or blueprint) for the HTML page
  3. Write HTML
  4. Finally, with the basic page done, you'll add some enhancements and meet some new HTML elements along the way

## Segway'n USA

Documenting my trip around the US on my very own Segway!

August 20, 2012



Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ and Truth or Consequences, NM.

July 14, 2012

I saw some Burma Shave style signs on the side of the road today: "Passing cars, When you can't see, May get you, A glimpse, Of eternity." I definitely won't be passing any cars.

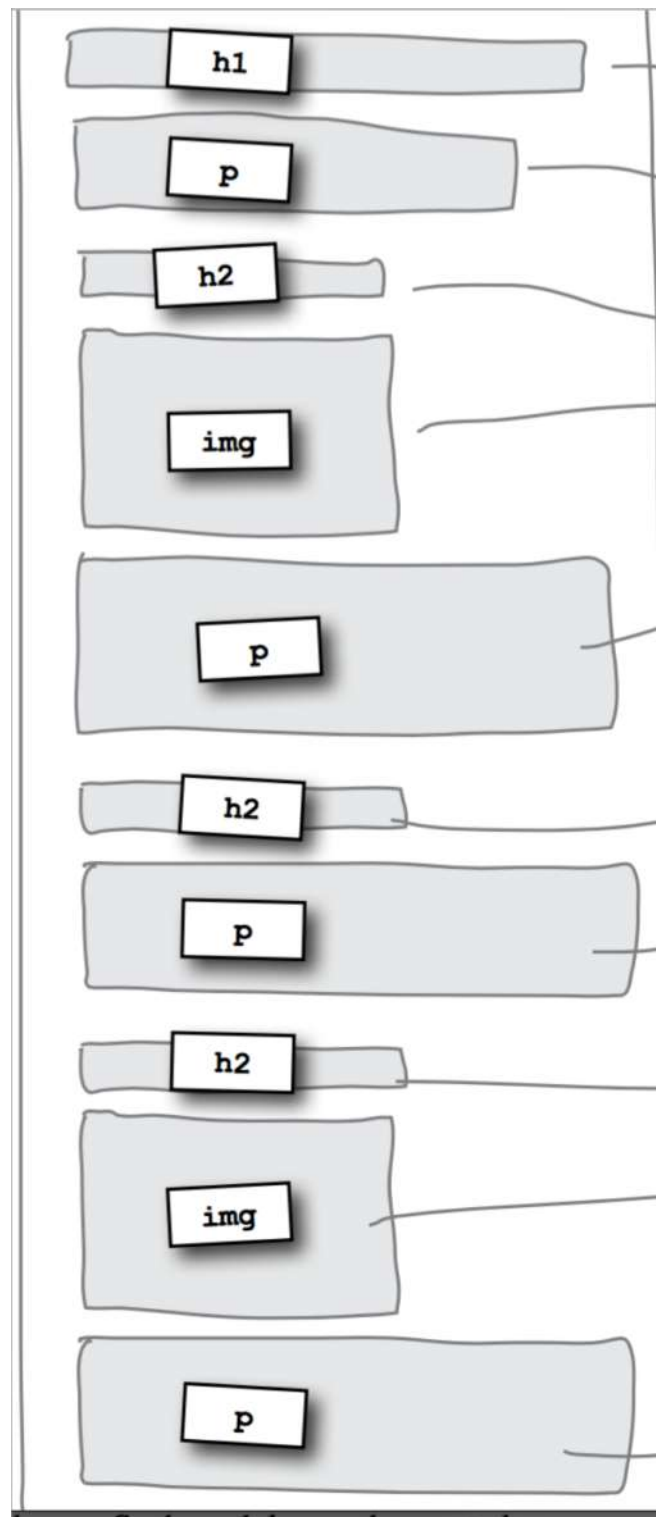
June 2, 2012



My first day of the trip! I can't believe finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cell phone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."

## Sketch

# Blocks, Write



# Result

My Trip Around the USA on a Segway

file:///chapter3/journal/journal.html

Google

## Segway'n USA

Documenting my trip around the US on my very own Segway!

### August 20, 2012



Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ and Truth or Consequences, NM

### July 14, 2012

I saw some Burma Shave style signs on the side of the road today: Passing cars, When you can't see, May get you, A glimpse, Of eternity. I definitely won't be passing any cars.

### June 2, 2012



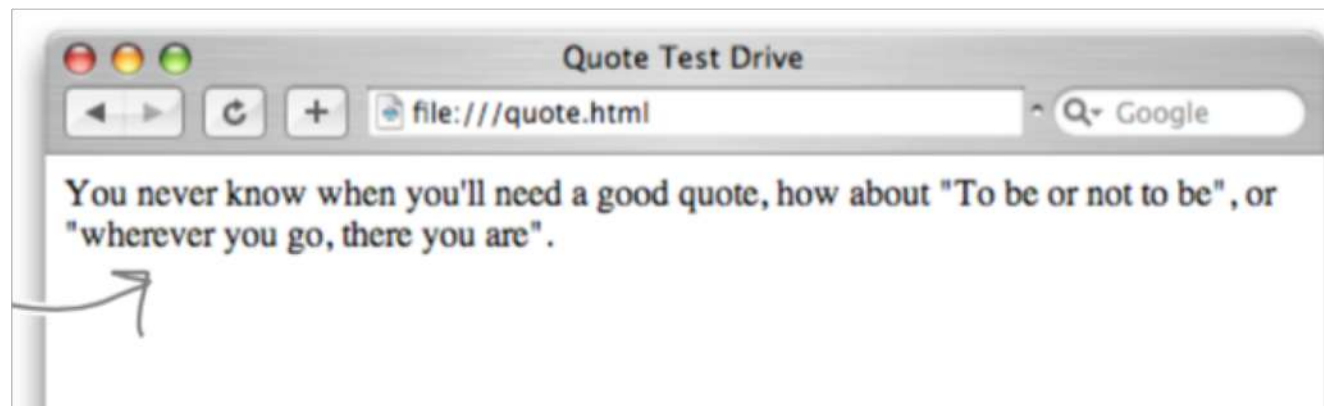
My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cellphone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."

# <q> quotations

```
<html>
  <head>
    <title>Quote Test Drive</title>
  </head>
  <body>
    <p>
      You never know when you'll need a good quote, how
      about <q>To be or not to be</q>, or <q>Wherever you go, there you are</q>
    </p>
  </body>
</html>
```

We've got two quotes in this HTML...

We surround each quote with a <q> opening tag and a </q> closing tag. Notice that we don't put our own double-quote characters around the quotes.





# <blockquote> blockquote

`<h2>July 14, 2012</h2>`

`<p>`

I saw some Burma Shave style signs on the side of the road today:

`</p>`

`<blockquote>`

Passing cars,  
When you can't see,  
May get you,  
A glimpse,  
Of eternity.

`</blockquote>`

`<p>`

I definitely won't be passing any cars.

`</p>`

To insert the `<blockquote>` element, we need to end this paragraph first.

Next we put the Burma Shave text in the `<blockquote>` element.

We also put each line of text on a separate line so it reads more like a Burma Shave slogan.

And finally, we need to add a `<p>` tag to start this paragraph after the `<blockquote>`.

**July 14, 2012**

I saw some Burma Shave style signs on the side of the road today:

Passing cars, When you can't see, May get you, A glimpse, Of eternity.

I definitely won't be passing any cars.

**June 2, 2012**



# <ol> ordered list (numbered)

```
<h2>August 20, 2012</h2>
```

```

```

```
<p>
```

Well I've made it 1200 miles already, and I passed through some interesting places on the way:

```
</p>
```

```
<ol>
```

```
<li>Walla Walla, WA</li>
```

```
<li>Magic City, ID</li>
```

```
<li>Bountiful, UT</li>
```

```
<li>Last Chance, CO</li>
```

```
<li>Why, AZ</li>
```

```
<li>Truth or Consequences, NM</li>
```

```
</ol>
```

← And here we close the <ol> element.

```
<h2>July 14, 2012</h2>
```

```
<p>
```

I saw some Burma Shave style signs on the side of the road today:

```
</p>
```

← We want this to be an ordered list, because Tony visited the cities in a specific order. So we use an <ol> opening tag.

← All the list items sit in the middle of the <ol> element and become its content.

## Ordered list (2)

Well I made it 1200 miles already, and I passed through some interesting places on the way:

1. Walla Walla, WA
2. Magic City, ID
3. Bountiful, UT
4. Last Chance, CO
5. Why, AZ
6. Truth or Consequences, NM

There's a linebreak before the list starts, so `<ol>` must be a block element.

But there's also a linebreak after each item, so `<li>` must be a block element too!

## Ordered list (3)

- ol element

```
<ol>
```

```
<li>...</li>
```

```
<li>...</li>
```

```
...
```

```
</ol>
```

- Attributes: type (1, A, a, l, i)  
start, compact

- List items: li

- Attributes: (when inside ol)  
value, type

A sample list:

1. List Item One
2. List Item Two
3. List Item Three

```
<p>A sample list:</p>
```

```
<ol>
```

```
<li>List Item One</li>
```

```
<li>List Item Two</li>
```

```
<li>List Item Three</li>
```

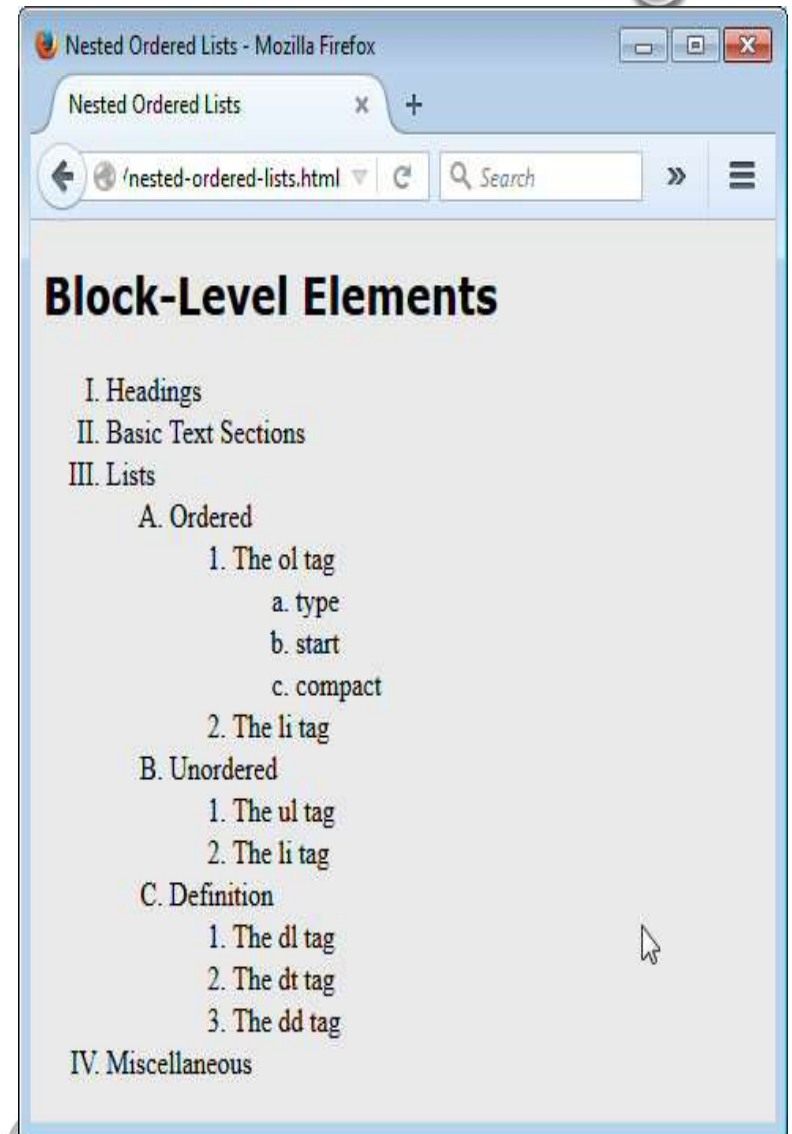
```
</ol>
```

```

<ol type="I">
  <li>headings</li>
  <li>basic text sections</li>
  <li>lists
    <ol type="a">
      <li>ordered
        <ol type="1">
          <li>the ol tag
            <ol type="a">
              <li>type</li>
              <li>start</li>
              <li>compact</li>
            </ol></li>
          <li>the li tag</li>
        </ol></li>
      <li>unordered ... </li>
      <li>definition ... </li>
    </ol></li>
  <li>miscellaneous</li>
</ol>

```

# Nested Ordered Lists



# <ul> unordered (Bulleted) Lists

- ul element

**<ul>**

**<li>...</li>**

**<li>...</li>**

**</ul>**

- Attributes: type, compact
  - type (disc, circle, or square)
- List items: li (type)
  - type (disc, circle, or square)

**<p>A sample list:</p>**

**<ul>**

**<li>List Item One</li>**

**<li>List Item Two</li>**

**<li>List Item Three</li>**

**</ul>**

A sample list:

- List Item One
- List Item Two
- List Item Three

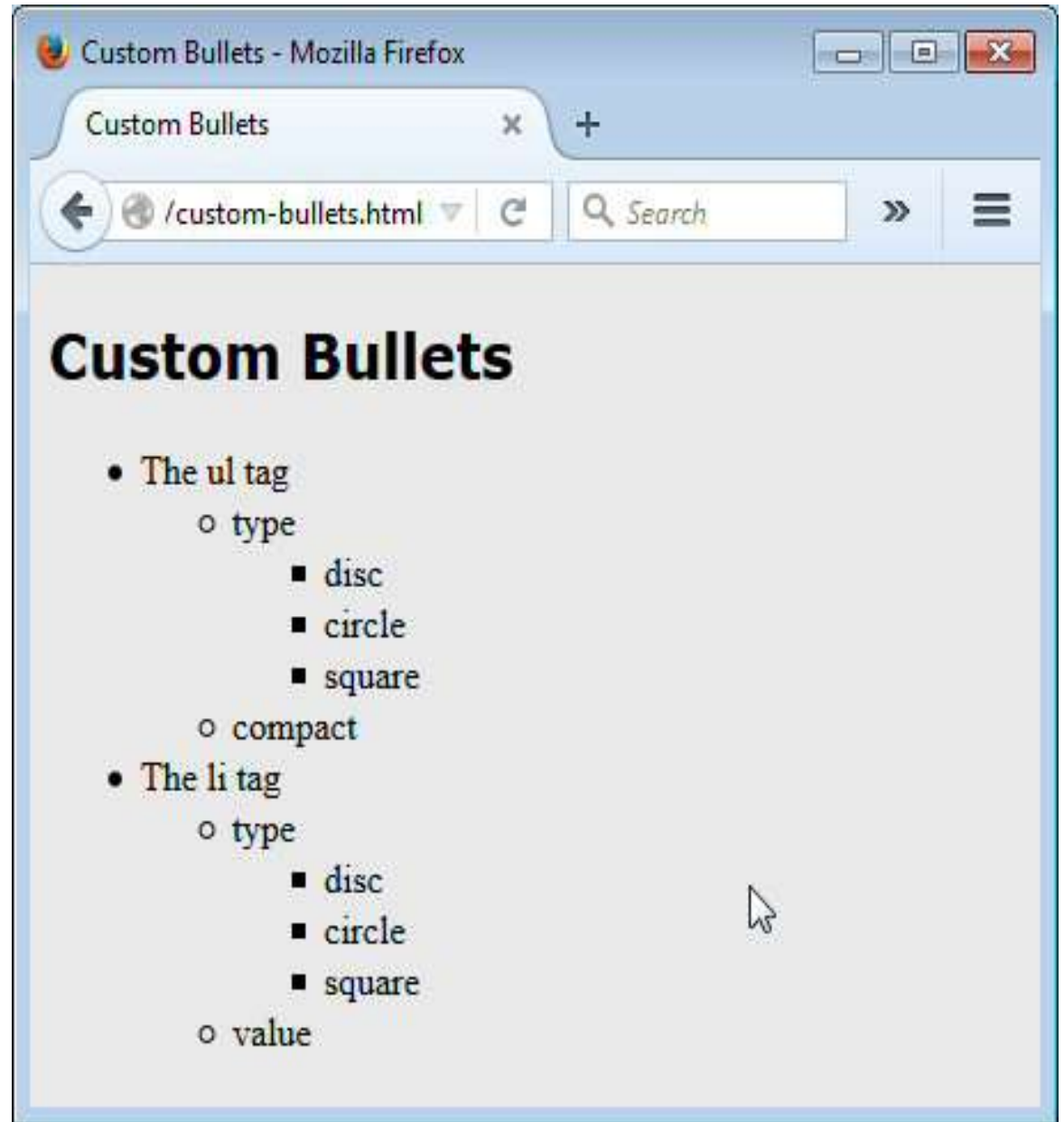


```

<ul type="disc">
  <li>The ul tag
    <ul type="circle">
      <li>type
        <ul type="square">
          <li>disc</li>
          <li>circle</li>
          <li>square</li>
        </ul></li>
        <li>compact</li>
      </ul></li>
    <li>The li tag
      <ul type="circle">
        <li>type
          <ul type="square">
            <li>disc</li>
            <li>circle</li>
            <li>square</li>
          </ul></li>
          <li>value</li>
        </ul></li>
      </ul>
    </ul>
  </ul>

```

## <ul> custom bullets





# Exercise



- Here's another list from Tony's journal:
  - cell phone, iPod, digital camera, and a protein bar. You'll find it in his June 2 entry. This is an *unordered* list of items.
- Make these changes in your "journal.html" file, too

Homework: L03 exercise





# A Brief History of HTML

## HTML 1.0-2.0 1989-1991

- Pages weren't pretty, but at least they were **hypertext enabled**. No one cared much about presentation, and just about everyone on the Web had their very own "home page"

## HTML3 - 1995

- "Browser Wars." Netscape and Microsoft were duking it out for control of the world. At the center of the fallout was the web developer. During the wars, an arms race emerged as each browser company kept adding their own **proprietary extensions** in order to stay ahead
- You had to often write two separate web pages: one for the Netscape browser and one for Internet Explorer. **Not good!**

# A Brief History of HTML (2)

## HTML4 1998

- The end of the Browser Wars and, to our rescue, the *World Wide Web Consortium* (nickname: W3C). Their plan: *to bring order to the universe by creating the **ONE HTML** “standard” to rule them all*
- Separate HTML’s **structure and presentation** into two languages—a language for structure (*HTML*) and a language for presentation (*CSS*)—and convince the browser makers it was in their best interest to adopt these standards

## HTML 4.01 1999

- The good life: Almost all browsers (at least the ones anyone would care about) were going to display our content just fine

# A Brief History of HTML (2)

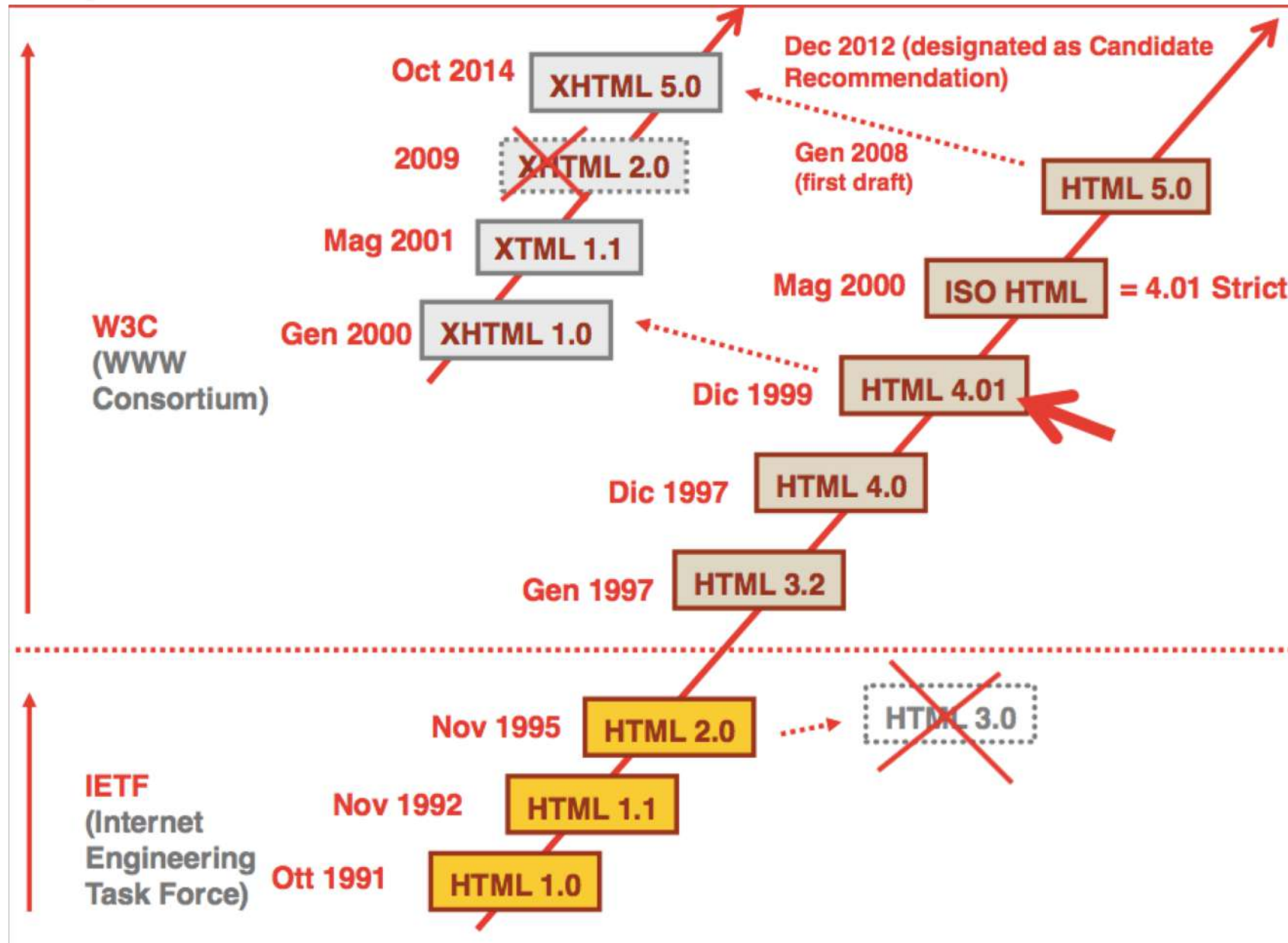
## XHTML 1.0 2001

- XHTML was developed to make HTML more extensible and increase interoperability with other data formats. *People ended up hating XHTML!*

## HTML5 since 2009

- The marriage between XML and HTML didn't end well and was replaced by new version of HTML named HTML5. With its support for most of the HTML 4.01 standard, and new features that reflect the way the Web has grown. *HTML5 is what developers were looking for!*
- And, with features like support for blog-like elements, new video and graphic capabilities, and a whole new set of capabilities aimed at building web applications, HTML5 was destined to become the standard
- To be honest, the divorce of HTML and XML took a lot of people by surprise, leading to confusion about what HTML5 actually is for a while

# History of HTML



# Web Page Validator

- A **validator** is a software program that can check your web pages against the web standards
- When using a validator to check HTML, XHTML or CSS documents, the validator returns a list of errors found, according to your chosen standard
- Make sure you make it a habit to validate all your web pages before publishing

# https://validator.w3.org



## Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by **URI**

Validate by **File Upload**

Validate by **Direct Input**

### Validate by URI

Validate a document online:

Address:

► [More Options](#)

Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).



The W3C validators are hosted on server technology donated by HP, and supported by community donations. [Donate](#) and help us build better tools for a better web.

5498

Flattr

# Validate Lounge

1. **Info** The Content-Type was `text/html`. Using the HTML parser.
2. **Info** Using the schema for HTML with SVG 1.1, MathML 3.0, RDFa 1.1, and ITS 2.0 support.
3. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).  
[From line 6, column 63; to line 6, column 84](#)  

```
e</h1> <p>↵Join
```



## Fixing that error

- You just need to add an **alt** attribute to your `<img>` elements in HTML5. Go ahead and open “lounge.html”, make the change, save, and then let’s try to validate again

```
<body>
  <h1>Welcome to the New and Improved Head First Lounge</h1>
  
  <p>
```

You know the alt attribute;  
add it into the <img> element

- Info** The Content-Type was `text/html`. Using the HTML parser.
- Info** Using the schema for HTML with SVG 1.1, MathML 3.0, RDFa 1.1, and ITS 2.0 support.

Document checking completed. No errors or warnings to show.

# References

- HTML 4.01 Specification: <http://www.w3.org/TR/html401/>
- HTML5: <http://www.w3.org/TR/html5/>
- Guida al linguaggio HTML: <http://www.w3schools.com/>