

HTML5 (Parte 1)

Prof. Elio Toppano – Università di Udine
Tecnologie Web e Laboratorio (a.a. 2020-2021)

Cosa è l'HTML?

- **HTML** significa **H**yper**T**ext **M**arkup **L**anguage
- E' un linguaggio di marcatura (annotazione) di documenti ipertestuali
- Serve per descrivere (o specificare) la **struttura** interna di una pagina web e i collegamenti ipertestuali che collegano la pagina ad altre pagine dello stesso sito/portale web o di siti diversi. Nelle prime versioni veniva usato anche per descrivere la **presentazione** (oggi si usano le regole CSS nei fogli di stile).
- L'HTML definisce un insieme di **elementi** che costituiscono i “building blocks” di una pagina web (es. paragrafi di testo, titoli, liste, tabelle, immagini, moduli, collegamenti).

Evoluzione dell'HTML


Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1

WHATWG: Web Hypertext Application Technology Working Group fondato nel 2004 da Opera, Mozilla, Apple e Google

Strumenti di sviluppo

- Per costruire un documento HTML serve:
 - un ***editor di testo*** per esempio Visual Studio Code, Atoms ... ma anche semplicemente Notepad (PC) , o TextEdit (Mac). Un documento HTML è identificato da un nome seguito dal suffisso .html
 - un ***browser*** per esempio IE, Safari, Chrome, Mozilla Firefox
- L'evoluzione del linguaggio HTML è stata accompagnata da una parallela evoluzione dei browser. Verificare che il browser che si usa per la visualizzazione della pagina sia in grado di interpretare correttamente HTML5.

Strumenti di validazione

**Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URIValidate by File UploadValidate 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](#).




Interested in "developing" your developer skills? In W3C's hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards. [Find out more!](#)

[Donate](#) and help us build better tools for a better web.

[Home](#) [About...](#) [News](#) [Docs](#) [Help & FAQ](#) [Feedback](#) [Contribute](#)



COPYRIGHT © 1994-2013 W3C® (MIT, ERCIM, KEIO, BEIHANG), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



URL: <https://validator.w3.org>

Manuale di riferimento

Jon DUCKETT

[HTML & CSS](#)

[design and build websites](#)

John Wiley & Sons, Inc.,
Indianapolis, Indiana, 2011

ISBN: **978-1-118-00818-8**

<https://wtf.tw/ref/duckett.pdf>



COMPONENTI STRUTTURALI DI BASE

Struttura

- Ogni tipo di documento ha una sua **struttura** che ne caratterizza il **genere**
- La struttura descrive: i) le **parti** o componenti che costituiscono il documento; ii) il loro **significato** e iii) le **relazioni** (es. spaziali o temporali) esistenti tra di esse.
- Esempi: struttura tipica di un *articolo scientifico*, di una *tesi di laurea*, di un *quotidiano*, di un *modulo di autocertificazione Covid*, di una *pagina web*, ecc.

Header

[HOME](#)[OUR TEAM](#)[PROJECTS](#)[CONTACT](#)

Article heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Set sit amet ipsum mauris. Maecenas congue ligula as quam viverra nec consectetur ant hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur.

subsection

Donec ut libero sed accu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.

Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nist impendiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros.

Another subsection

Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et manis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, is fringille sem nunc vet mi.

Related

- [Oh I do like to be beside the seaside](#)
- [Oh I do like to be beside the sea](#)
- [Although in the North of England](#)
- [It never stops raining](#)
- [Oh well...](#)

HTML: elementi, tag

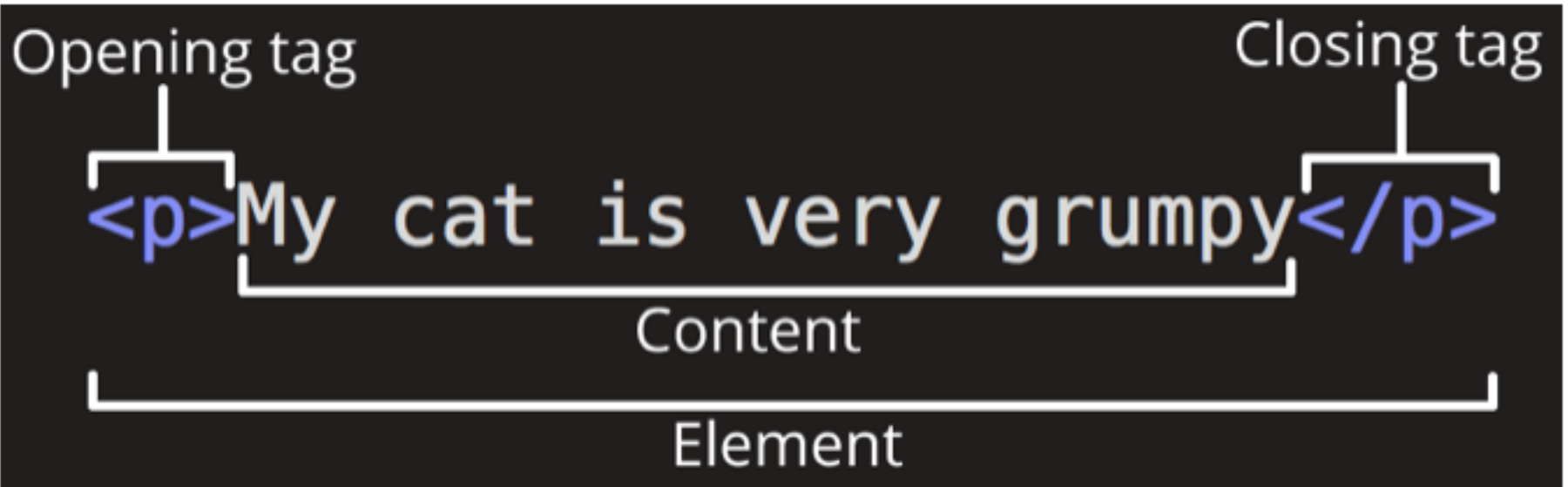
- **Elemento:** è un *contenitore* di informazione/dati. Viene identificato da un **tag di apertura** e un **tag di chiusura**. Alcuni elementi sono vuoti (non contengono dati).
- **Tag:** è una stringa di caratteri (anche un solo carattere) racchiusa tra parentesi angolari (<...>). *Tag di apertura:* la stringa specifica il nome dell'elemento (es. <nomeElemento>). *Tag di chiusura:* la stringa è costituita dal carattere di chiusura forward slash (/) concatenato col nome dell'elemento (es. </nomeElemento>). Per gli elementi vuoti lo slash di chiusura si inserisce direttamente nel tag di apertura prima del simbolo ">" (es. <nomeElemento />). Esempi di elementi: titolo (h), paragrafo (p) e nuova linea (br)

<h1> Questo è un titolo **</h1>**

<p> Questo è un paragrafo di testo. **</p>**

**
**

Anatomia di un elemento HTML

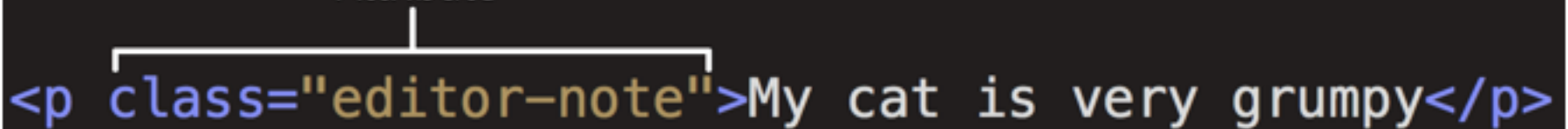


HTML: attributi e valori

- **Attributo:** descrive una *proprietà* dell'elemento. Ha un **nome** e un **valore**. Il valore è racchiuso tra (doppie o singole) virgolette (quotation mark) ed è assegnato al nome attraverso il segno “=”. Gli attributi vengono inseriti nei tag di apertura dopo il nome dell'elemento. Esempio: attributo ‘lingua’ del testo di un paragrafo con valore ‘inglese-USA’

`<p lang="en-us" > Paragraph in english. </p>`

Attribute



```
<p class="editor-note">My cat is very grumpy</p>
```

Annidamento di elementi (nesting)

- Un documento HTML contiene elementi annidati all'interno di altri elementi (elementi che contengono altri elementi).
- Esempio: un elemento 'paragrafo' che contiene un elemento 'a capo'

**<p lang="en-us" > Paragraph
 in english. </p>**

- Aprire e chiudere gli elementi in maniera corretta! (si vedano le slide su XML)

Note

1. Le regole di scrittura dei tag in HTML5 sono molto flessibili. Infatti, per gli elementi vuoti come per esempio `br`, la presenza del carattere slash di chiusura non è obbligatoria (si può scrivere `
` o `
`). Inoltre i tag e gli attributi possono essere scritti con caratteri in maiuscolo, minuscolo o addirittura in modo misto; per gli attributi booleani non è necessario specificare il valore relativo che è indicato automaticamente dalla presenza dell'attributo, mentre per gli altri il valore può essere scritto tra singoli apici, doppi apici, o senza apici. La flessibilità di HTML5 non dovrebbe però essere abusata. E' buona regola darsi delle regole e cercare di essere il più possibile rigorosi come se si lavorasse sui documenti XHTML; in questo modo si migliora la leggibilità, la scalabilità e la semantica dei documenti prodotti. Cercate di seguire le regole viste per xml!
2. E' buona pratica introdurre commenti per descrivere le varie parti del documento. In HTML5 i commenti sono definiti utilizzando la seguente notazione:

`<!-- commento -->`

ANATOMIA DI UN DOCUMENTO HTML

Dichiarazione di tipo

- Un documento HTML inizia con la *dichiarazione* del tipo di documento. Per i documenti HTML5 si scrive:

<!DOCTYPE html>

- La dichiarazione indica al browser di visualizzare la pagina in conformità alle regole e alla grammatica (il DTD) che il W3C ha codificato per la versione del linguaggio specificato, quelle stesse regole che il progettista si impegna a seguire nella marcatura del documento html.
- Si veda nella slide seguente altre definizioni adottabili

HTML 4.01 Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Frameset:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

HTML XHTML 1.0 Strict:

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

HTML XHTML 1.0 Transitional:

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

HTML XHTML 1.0 Frameset:

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

HTML XHTML 1.1 :

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

Esempio: XHTML 1.0

- La scrittura di un documento XHTML è soggetta a regole precise (ereditate da XML):
 - R1: ogni elemento deve avere un tag di apertura e di chiusura (unica eccezione gli elementi vuoti).
 - R2: i nomi degli attributi vanno scritti in minuscolo.
 - R3: tutti gli attributi richiedono un valore; i valori vanno inseriti tra virgolette doppie.
 - R4: gli elementi deprecati non vanno usati.
 - R5: ogni elemento che viene aperto all'interno di un altro elemento deve essere chiuso all'interno dello stesso elemento.
- Esistono due versioni principali di XHTML:
 - Strict XHTML 1.0:** gli autori devono seguire le regole di scrittura alla lettera
 - Transitional XHTML 1.0:** gli autori possono ancora usare elementi di presentazione (come `<center>` e ``)
- **Nota: XHTML 1.0 Frameset** permette agli autori di partizionare la finestra del browser in più “frames” ciascuno dei quali ospita una pagina HTML differente.

Struttura generale del documento

- Un documento HTML5 è costituito da: i) la dichiarazione del tipo di documento; ii) un elemento *html* (*radice* del documento) costituito nell'ordine da una testa (*head*) e un corpo (*body*) secondo lo schema seguente:

<!DOCTYPE html>

<html>

<head>

[...]

</head>

<body>

[...]

</body>

</html>

La testa (head) contiene informazioni sul documento (*metadati*) che non vengono visualizzate nella pagina web.

Il corpo (body) contiene la specifica della struttura dei contenuti (es. testi,immagini) della pagina che saranno visualizzati dal browser

Metadati (1)

- I *metadati* (dati su altri dati) sono informazioni che descrivono il documento nel suo complesso. Vengono introdotti nella testa del documento usando l'elemento `<meta />` assieme agli attributi *name* e *content*. Alcuni esempi:

```
< meta name="author" content=" .. qui va il nome dell'autore.. " />
```

```
< meta name="description" content=" .. qui va la descrizione del documento.. " />
```

```
<meta name="keywords" content=" .. qui vanno le parole chiave separate dalla virgola .." />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Metadati (2)

- Altri metadati importanti riguardano l'insieme di caratteri usato nel documento (charset), il tipo di contenuto e il titolo (title) del documento. Quest'ultimo verrà visualizzato nella barra del titolo della finestra del browser usato

<head>

<meta charset="utf-8" /> <!-- UTF8 è il character set di default di HTML5 -->

<meta http-equiv="content-type" content="text/html" />

<title> qui va il titolo del documento </title>

</head>

le prime due specifiche possono essere compattate:

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

Nota: usare virgolette diritte (quotation mark: " ") non curve per introdurre valori di attributi!

INSERIMENTO E FORMATTAZIONE DI BASE DEL TESTO

Testo

- Il testo viene visto come una mera sequenza di caratteri che hanno un qualche significato; è uno dei contenuti principali delle pagine web
- Il testo deve essere
 - **Leggibile** (chiarezza visiva)
 - **Scansionabile** (possibilità di identificare l'informazione pertinente in una pagina senza dover leggere parola per parola, a colpo d'occhio!)
 - **Comprensibile** (grado di intelligibilità della prosa; minimizzare la complessità sintattica, terminologica e semantica rispetto al target d'utenza previsto)

Elementi di base

<h1> ... </h1> , <h2> ... </h2> , .., <h6> ... </h6> : servono per la definizione dei **titoli** (heading); dal più importante (h1) visualizzato in grassetto con il font più grande al meno importante (h6) visualizzato in grassetto con font più piccolo, passando per i titoli intermedi visualizzati con font di dimensioni progressivamente inferiori (h2,h3,h4,h5). La scelta dei titoli è importante per migliorare la scansionabilità del testo, la indicizzazione della pagina da parte dei motori di ricerca e l'accessibilità.

Osservazione: in una pagina, usare al più 3 livelli diversi; usare h1 preferibilmente una volta sola; non creare “buchi” tra livelli (es. uso h1 e poi h3,h5); usare i livelli in maniera consistente (non usare h2 per il titolo di un paragrafo; h3 per il titolo di un sottoparagrafo e h2 per il titolo di un sotto-sotto paragrafo)

<hgroup> ... </hgroup> : consente di **raggruppare titoli** secondo una strutturazione a livelli multipli (es. titolo e sottotitolo)

<p> .. </p> : definisce un **paragrafo** il cui testo viene di norma visualizzato con l'aggiunta di una spaziatura superiore e inferiore (margini). E' un elemento di *tipo blocco*: ogni paragrafo inizia su una nuova riga. La distinzione tra elementi blocco (block) e in linea (inline) verrà ripresa più avanti!

**
** : serve per andare **a capo** (Usare con moderazione! Qualcuno usa un paragrafo vuoto: <p></p>)

<hr /> : introduce un **filetto orizzontale** (horizontal rule). Può essere usato per separare contenuti o specificare un cambiamento di tema

** ... ** oppure **<i> ...</i>** : servono per la definizione di testo in **grassetto** o in **italics** (usarli con moderazione!). Il grassetto viene spesso usato per parole chiave, nomi di prodotti, ... l'italics per le parole tecniche o per le parole straniere.

** ... ** oppure ** ...** : servono per dare **enfasi** (importanza) ad un segmento di testo. Nota: strong viene reso in grassetto, em in italics

Osservazioni su accessibilità:

- L'elemento i (e b) indica un aspetto *tipografico* (italics, bold). Un lettore di schermo non distingue tra caratteristiche tipografiche diverse!
- L'elemento em (o strong) indicano un *significato semantico* (enfasi, importanza). Un lettore di schermo utilizzerà una qualche tecnica di enfattizzazione (es. pausa, intensità vocale) per comunicare l'importanza del contenuto annotato con questi tag

^{...} o **_{...}** : servono per la definizione di **apici** o di **pedici**.

Esempio: H₂O, xⁿ. Esempio

<p> Formula chimica dell'acqua: H **₂** O </p>

<blockquote> ...</blockquote> oppure **<q> ... </q>** : servono per la definizione di una sezione il cui testo è una **citazione**. Blockquote visualizza il testo *su una nuova riga e rientrato*, q lo visualizza *in linea e tra doppi apici*. Questi due elementi possono usare l'attributo *cite* per introdurre l'URL della citazione.

<cite> ... </cite> : serve per introdurre **citazioni bibliografiche**. E' un componente *in linea* il cui contenuto viene visualizzato in italics. Viene usato per citare la risorsa (es. il titolo di un libro, la pagina web, titolo di un film) che contiene la citazione piuttosto che la citazione stessa.

<abbr title=" "> ... </abbr> : serve per introdurre **abbreviazioni e acronimi**.

Esempio:

.... **<abbr title="International Standard Organization">** ISO **</abbr>**

- **<dfn> ..</dfn>** : serve per introdurre **definizioni**. La definizione compare passando col cursore sopra il termine definito

<dfn title="Insieme di componenti software che"> Sistema Operativo </dfn>

- **<pre> ... </pre>** : serve per la definizione di **codice preformattato** (es. codice java, C++); vengono conservati gli "a capo" e gli spazi bianchi...
- **<code> ...</code>**: serve per specificare che il contenuto è un codice. Utilizza un font monospaziato; non conserva spazi bianchi o "a capo". E' un elemento in linea
- ** ..** : serve per **cancellare** testo (il testo cancellato appare tagliato)
- **<ins> ..</ins>** : serve per **inserire** del testo (il testo inserito appare sottolineato)
- ecc.

Si consulti il libro di riferimento per altri elementi!

Caratteri speciali (character entities)

- Alcuni caratteri (es. `<` , `>`) hanno un significato particolare nel linguaggio. Se devono essere usati all'interno di un testo vanno indicati con caratteri speciali (es. `<` o `>`;) La tabella seguente mostra alcuni esempi di caratteri speciali
- Nota: il carattere speciale ` ` (non-breaking space) viene usato per forzare l'inserimento del carattere spazio quando si ha la necessità di visualizzare parole separate non da uno, ma da più spazi. Evitare di usarlo per la formattazione; usare fogli di stile!

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

Note di stile

- Lo stile del testo va dichiarato con i fogli di stile (vedi lezioni su CSS3)
- Per ora (ma è sconsigliato!) usiamo l'attributo *style* (stile) all'interno di alcuni elementi (es. intestazioni e paragrafi) nel formato seguente:

`style="property:value;"`

Dove *property* (proprietà) è una caratteristica stilistica e *value* (valore) specifica il relativo valore

- Esempi di proprietà: color, background-color, font-family, font-size, text-align.

`<p style="color:red;font-family:verdana;font-size:50px;">` qui va il testo **`</p>`**

Esercizio

Scrivere il codice HTML per visualizzare il seguente frammento di testo
(inserire il significato della abbreviazione MDN (Mozilla Developers Network)):

According to the *MDN block quote page* :

The **HTML <blockquote> Element** (or *HTML Block Quotation Element*)
indicates that the enclosed text is an extended quotation.

The quote element - <q> - is " intended for short quotations that don't require paragraph breaks. "
MDN q page.

Possibile soluzione:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title> Esempio n.0 </title>
```

```
</head>
```

```
<body>
```

```
<p> According to the <cite> <abbr title="Mozilla Developers Network">MDN </abbr>block quote page </cite>:
```

```
<blockquote cite="URL ...">
```

```
    The <strong> HTML <code>&lt;blockquote&gt;</code>Element</strong> (or <em>HTML Block  
    Quotation Element</em> ) indicates that the enclosed text is an extended quotation.
```

```
</blockquote>
```

```
</p>
```

```
<p> The quote element - <code>&lt;q&gt;</code> - is <q> intended for short quotations that don't require  
paragraph breaks. </q> <cite> <abbr title="Mozilla Developers Network">MDN </abbr> q page. </cite> </p>
```

```
</body>
```

```
</html>
```


ATTRIBUTI SEMANTICI: ID E CLASS

Attributi: *id* e *class*

- Ogni elemento HTML può essere **identificato univocamente** da un identificatore usando l'attributo ***id*** con il relativo *valore* (`id="valore"`) . Il valore deve iniziare con una lettera o con "_" (non si possono usare numeri o altri caratteri) e **deve essere unico**. L'identificatore è importante per l'applicazione di specifiche regole stilistiche all'elemento identificato.
- Ogni elemento HTML può essere **classificato** (associato ad una classe assieme ad altri elementi!) Si usa l'attributo ***class*** con il relativo valore (`class=" nomeclasse"`). Un elemento può appartenere a più classi (i nomi vanno separati da spazio). La classe è importante per dare maggior significato all'elemento e per l'applicazione di specifiche regole stilistiche all'elemento di appartenenza.

Esempio

...

<p id="par1"> Lorem ipsum </p>

<p id="par2"> Lorem ipsum </p>

...

<p class="intro"> Lorem ipsum </p>

<p class="related_work"> Lorem ipsum </p>

<p class="method"> Lorem ipsum </p>

<p class="results"> Lorem ipsum </p>

<p class="conclusion"> Lorem ipsum </p>

<p class="biblio"> Lorem ipsum </p>

...

Nota: un identificatore (es. par1, par2) assegna ad un elemento (es. un paragrafo) un nome univoco usabile in altri elementi o nelle regole di presentazione. L'attribuzione di una classe ad un elemento può essere fatta sulla base del **contenuto** o della **funzione** (ruolo retorico) dell'elemento nel documento complessivo. Si veda la lezione sulla progettazione del contenuto!

ELEMENTI BLOCCO E IN LINEA

Elementi blocco (block)

- **Elemento blocco:** un elemento di tipo blocco (block) viene visualizzato in una nuova linea (va a capo), di seguito all'elemento che lo precede.
- Ogni elemento che segue un elemento blocco viene visualizzato su una nuova linea.
- Gli elementi blocco sono tipicamente elementi strutturali della pagina per esempio: titoli, paragrafi, blocchi di testo, liste, menù di navigazione, ecc.
- Il lucido che segue elenca i principali elementi di tipo blocco

<code><address></code>	Contact information.	<code><fieldset></code>	Field set label.	<code></code>	List item.
<code><article></code>	Article content.	<code><figcaption></code>	Figure caption.	<code><main></code>	Contains the central content unique to this document.
<code><aside></code>	Aside content.	<code><figure></code>	Groups media content with a caption (see <code><figcaption></code>).	<code><nav></code>	Contains navigation links.
<code><blockquote></code>	Long ("block") quotation.	<code><footer></code>	Section or page footer.	<code></code>	Ordered list.
<code><details></code>	Disclosure widget.	<code><form></code>	Input form.	<code><p></code>	Paragraph.
<code><dialog></code>	Dialog box.	<code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code>	Heading levels 1-6.	<code><pre></code>	Preformatted text.
<code><dd></code>	Describes a term in a description list.	<code><header></code>	Section or page header.	<code><section></code>	Section of a web page.
<code><div></code>	Document division.	<code><hgroup></code>	Groups header information.	<code><table></code>	Table.
<code><dl></code>	Description list.	<code><hr></code>	Horizontal rule (dividing line).	<code></code>	Unordered list.
<code><dt></code>	Description list term.				

Elemento in linea (inline)

- **Elemento in linea:** un elemento di tipo in linea (inline) appare tipicamente all'interno di un elemento blocco. Può contenere altri elementi in linea, ma non elementi di tipo blocco. Viene visualizzato di seguito al contenuto che lo precede senza andare a capo.
- Ogni elemento che segue un elemento in linea viene visualizzato a capo o di seguito (senza andare a capo) a seconda del tipo
- Il lucido che segue elenca i principali elementi in linea

<a>
<abbr>
<acronym>
<audio> (if it has visible controls)

<bdi>
<bdo>
<big>

<button>
<canvas>
<cite>
<code>
<data>
<datalist>

<dfn>

<embed>

<i>
<iframe>

<input>
<ins>
<kbd>
<label>
<map>
<mark>
<meter>
<noscript>
<object>
<output>
<picture>
<progress>
<q>
<ruby>
<s>
<samp>

<script>
<select>
<slot>
<small>

<sub>
<sup>
<svg>
<template>
<textarea>
<time>
<u>
<tt>
<var>
<video>
<wbr>

AGGREGAZIONE DI ELEMENTI

Elementi *div* e *span*

- L'elemento *div* (divisione) permette di raggruppare più elementi HTML5 in un unico segmento al quale viene attribuita un qualche tipo di coerenza semantica o pragmatica. E' un elemento di tipo blocco: il segmento viene visualizzato iniziando da una nuova riga
- L'elemento *span* (intervallo) permette di raggruppare più elementi di tipo in linea in un unico segmento (non va a capo). E' utile per raggruppare e annotare frammenti di testo al fine di presentarli stilisticamente in maniera diversa da altri
- Sia *div* che *span* possono possedere attributi *id* o *class*.

< div class="articolo" id="art120">

<p class="intro">

<p class="related_work">

<p class="method">

<p class="results">

<p class="conclusions">

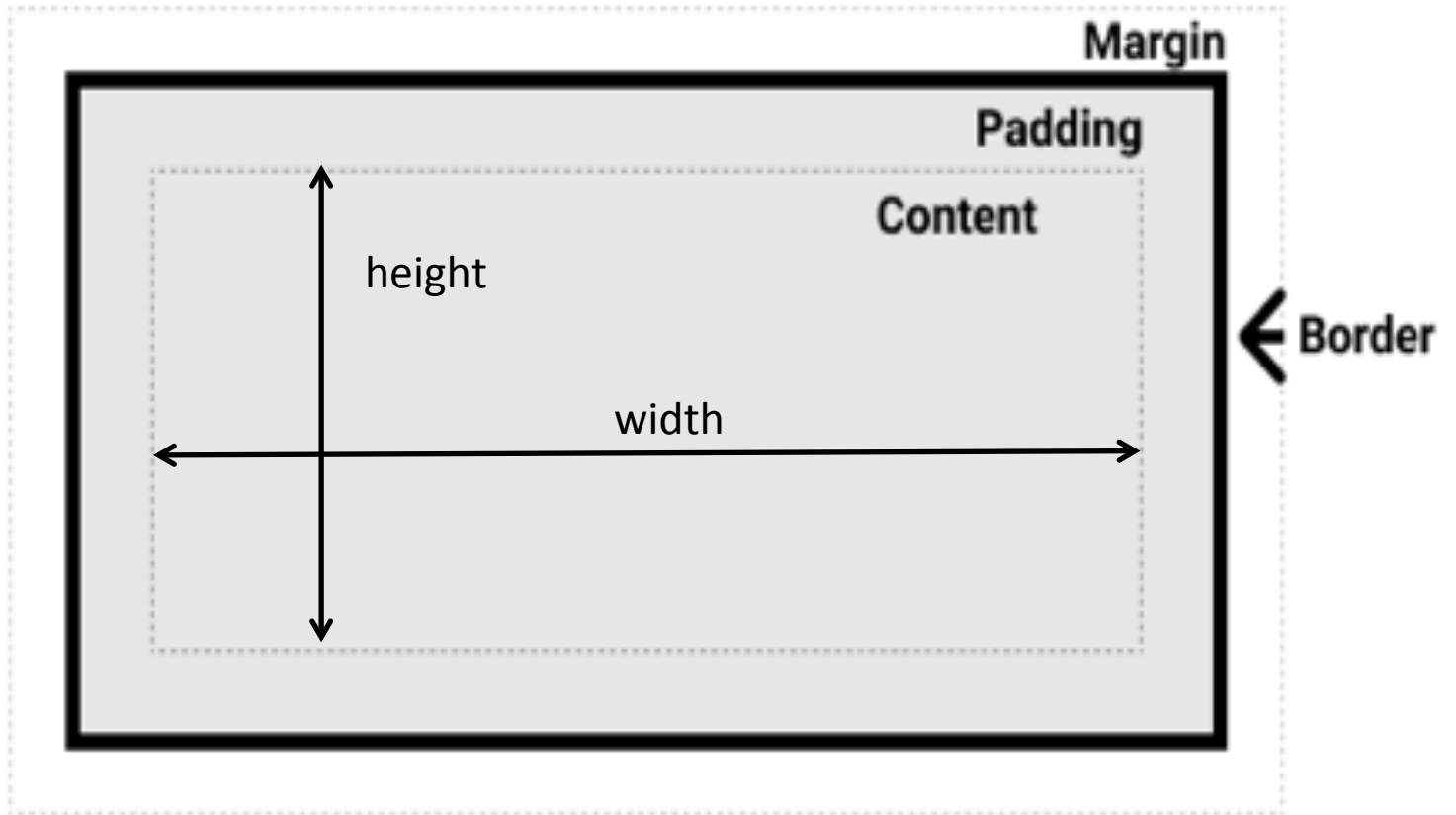
<p class="biblio">

< div ... >

<p class="profilo_utente" id="u11">

CSS: Box model

- Un elemento blocco (o inline) è una entità costituita da:
 - **contenuto**: è l'area in cui viene visualizzato il contenuto dell'elemento. Ha una larghezza e una altezza che possono essere specificati con CSS (questo vale però solo per gli elementi di tipo blocco!)
 - **padding**: è un spazio di contorno che racchiude il contenuto. Le sue proprietà (es. spessore, colore) possono essere specificate con CSS
 - **border**: è il bordo che contiene il contenuto e il padding. Le sue proprietà (es. spessore, colore, tipo di bordo) possono essere specificate con CSS
 - **margin**: è uno spazio di contorno esterno al bordo. Le sue proprietà (es. spessore, colore di sfondo) possono essere specificate con CSS



NUOVI ELEMENTI PER LA STRUTTURAZIONE DI UNA PAGINA HTML5

Osservazione

- E' consuetudine realizzare la struttura di un documento HTML avvalendosi degli elementi *div*
- Oggi, si sta iniziando a sconsigliarne l'uso per casi speciali (sezioni di intestazione, footer, sezioni di navigazione) perché sono considerati troppo generici e privi di semantica.
- L'HTML5 ha introdotto alcuni elementi nuovi che permettono di migliorare la leggibilità e la manutenibilità del codice in quanto rendono subito evidente lo scopo per cui sono utilizzati e rendono chiara la struttura del documento. Inoltre tali elementi migliorano l'accessibilità del documento alle persone che hanno problemi di disabilità

Header

[HOME](#)[OUR TEAM](#)[PROJECTS](#)[CONTACT](#)

Article heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Set sit amet ipsum mauris. Maecenas congue ligula as quam viverra nec consectetur ant hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur.

subsection

Donec ut libero sed accu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.

Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nist impendiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros.

Another subsection

Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et manis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, is fringille sem nunc vet mi.

Related

- [Oh I do like to be beside the seaside](#)
- [Oh I do like to be beside the sea](#)
- [Although in the North of England](#)
- [It never stops raining](#)
- [Oh well...](#)

<body>

<div id="page">

<div id="header">

<div id="nav">

<div id="content">

<div class="article">

<div class="article">

<div id="sidebar">

<div id="footer">

I nuovi elementi: *header* e *footer*

- **Testata** (<header>): è utile, in generale, per rappresentare un'area di intestazione per una pagina (o per una sezione) ossia un gruppo di elementi introduttivi o di ausilio alla navigazione (es. nome dell' azienda o del giornale online, logo, menu principale di navigazione, ricerca, ...)
- **Piè di pagina** (<footer>): è utile per indicare un'area a piè di pagina (o di sezione), dove è possibile raggruppare elementi come l'autore della pagina, informazioni di copyright, regole per l'accessibilità, contratti di licenza, ecc.

I nuovi elementi: *article* e *section*

- **Articolo** (<article>): è utile per indicare un'area di un documento che rappresenta un contenuto autosufficiente e quindi potenzialmente distribuibile e usabile al di fuori del contesto originale (es. un articolo di giornale, un post di un blog, una scheda prodotto, un modulo standard per l'acquisizione di dati demografici)
- **Sezione** (<section>): è utile per indicare un'area di un documento tematicamente (o funzionalmente) omogenea. La coerenza tematica dei contenuti da strutturare e la presenza di un titolo identificativo sono i due fattori cruciali per distinguere l'elemento section dall'elemento div che è un elemento per creare raggruppamenti generici
- Esempio: una tesi o un saggio sono elementi di tipo article mentre i vari capitoli della tesi o del saggio sono delle section

I nuovi elementi: *nav* e *aside*

- **Navigazione** (<nav>): è utile per raggruppare elementi di supporto alla navigazione (es. una lista non ordinata di elementi ancora *a*). Può far parte della testata (menù di navigazione principale) o rappresentare un menù secondario (fuori della testata)
- **A parte** (<aside>): è utile per indicare un' area di un documento che contiene informazioni correlate al contenuto principale, ma in modo tangenziale (ossia non strettamente appartenenti al contesto principale)

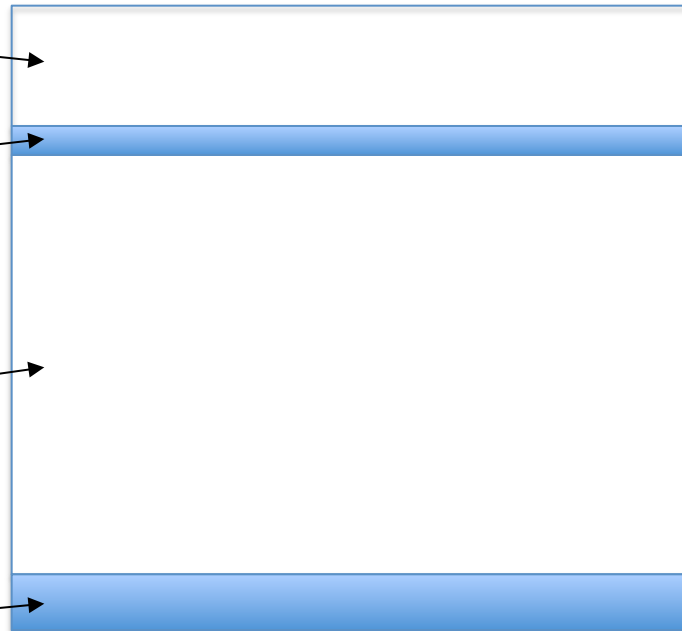
In HTML5 diventa (per esempio) :

`<header>`
 `[...]`
`</header>`

`<nav>`
 `[...]`
`</nav>`

`<article>`
 `[...]`
`</article>`

`<footer>`
 `[...]`
`</footer>`



Note: la sezione di navigazione può essere contenuta nell'intestazione; l'articolo può essere composto da sezioni ciascuna col proprio titolo

Per concludere

- Se il raggruppamento si giustifica solo con esigenze di impaginazione e di stile della presentazione si usi **div**
- Se i contenuti configurano una sezione con una funzione definita di navigazione, di composizione che forma una parte indipendente della pagina, oppure di informazioni correlate al contenuto principale si usi rispettivamente **nav, article e aside**
- Se i contenuti formano un insieme tematico coerente preferibilmente identificato da un titolo si usi **section**

Accessibilità: WAI_ARIA

- WAI-ARIA è l'acronimo di Web Accessibility Initiative – Accessible Rich Internet Applications
- ARIA mette a disposizione una serie di attributi (es. *role*, *aria-label*) per specificare in maniera più accurata il contenuto di un div. Vanno usati solo se strettamente necessario!

Si veda: <https://www.w3.org/WAI/standards-guidelines/aria/>

- Per l'accessibilità usare *span* per racchiudere testi scritti in una lingua diversa da quella del documento principale esplicitando all'interno di *span*, tramite l'attributo *lang*, la lingua usata ed eventualmente l'ordine di lettura tramite l'attributo *dir* che assume valori *ltr* (left to right) o *rtl* (right to left)

LISTE

Generazione di liste

- Le liste sono uno strumento utile per mostrare del testo sotto forma di *elenchi ordinati oppure non ordinati*. L'uso delle liste favorisce la *scansionabilità* del testo
- In HTML5 si possono definire tre tipi principali di liste:
 - liste ordinate
 - liste non ordinate
 - liste di definizione (o di descrizione)
- Si possono inoltre definire liste di liste (liste annidate o gerarchiche)

Liste ordinate

Si usano gli elementi *ol* (ordered list) e *li* (list item):

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

Gli elementi della lista vengono contrassegnati (per default) con numeri interi. Si può cambiare il contrassegno usando l'attributo *type* all'interno dell'elemento *ol* con possibili valori: 1 (default), A, a, I, o i. Esempio

```
<ol type="I"> ..... </ol>
```

```
<!-- Uso numeri romani maiuscoli-->
```

```
<ol type="a"> ..... </ol>
```

```
<!-- Uso lettere alfabeto minuscole-->
```

- Si può usare l'attributo *start* (in *ol*) per indicare il numero dal quale iniziare la numerazione dell'elenco. Per esempio l'istruzione `<ol start="5">` farà cominciare la lista con il numero 5.
- Si può usare l'attributo *value* (in *li*) per forzare un numero d'ordine specifico. Gli elementi successivi seguiranno quel numero
- Si può usare l'attributo booleano *reversed* che, se presente, permette di mostrare l'elenco in modo discendente. L'attributo *reversed* non è supportato da tutti i browser (es. IE e Safari 5.1 e precedenti)

Liste non ordinate

Si usano gli elementi *ul* (unordered list) e *li* (list item):

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

Gli elementi della lista vengono contrassegnati (per default) con dei piccoli cerchi neri (disc). Si può cambiare il contrassegno usando l'attributo *style* all'interno dell'elemento *ul* con proprietà *list-style-type* e valori *disc*, *circle*, *square*, o *none* . Esempio

```
<ul style="list-style-type:square;"> ..... </ul>  <!-- si usano quadrati -->
```

Liste di definizione

Si usano gli elementi *dl* (definition list), *dt* (definition term) e *dd* (definition description):

<dl>

`<dt> ... </dt>`

`<dd> ... </dd>`

...

`<dt> ... </dt>`

`<dd> ... </dd>`

</dl>

Le liste di definizione si possono usare, ad esempio, per introdurre le voci di un vocabolario o per introdurre una lista di domande con relative risposte ...

Liste annidate

```
<ul>
<li> nomeItem1
  <ul>
    <li> .. </li>
    <li> ..</li>
  </ul>
</li>
<li> nomeItem2
  <ul>
    <li> .. </li>
    <li> ..</li>
    <li> ..</li>
  </ul>
</li>
...
</ul>
```

Esempio di annidamento
di liste non ordinate (ul) in
una lista principale non
ordinata (ul)

TABELLE

Creazione di tabelle

- Una **tabella** è una struttura che permette di organizzare un insieme di dati in una griglia (matrice) formata da un certo numero di righe e di colonne.
- Le righe e le colonne hanno tipicamente dei **nomi** associati (le intestazioni).
- Ogni elemento della tabella si chiama **cella**.

Elementi: table, tr, td

- In HTML5 il contenuto di una tabella viene specificato riga per riga.
- Si usa l'elemento ***table*** per definire la tabella; l'elemento ***tr*** (table row) per definire ciascuna riga e l'elemento ***td*** (table data) per specificare (il contenuto) di una cella di una riga. Il contenuto di una cella può essere un elemento qualsiasi: un testo, una lista, un'immagine, una tabella.

Esempio-0

- Vogliamo rappresentare la seguente tabella (3 x 3) di numeri interi:

15 15 30

45 60 45

60 90 90

<table>

<tr> **<td>**15**</td>** **<td>**15**</td>** **<td>**30**</td>** **</tr>**

<tr> **<td>**45**</td>** **<td>**60**</td>** **<td>**45**</td>** **</tr>**

<tr> **<td>**60**</td>** **<td>**90**</td>** **<td>**90**</td>** **</tr>**

</table>

Inserimento delle intestazioni

- Si usa l'elemento ***th*** (table heading) per inserire le intestazioni (di riga o di colonna)
- Si può usare l'attributo ***scope*** (con valori: *col* o *row*) per specificare se l'intestazione è di colonna (*scope="col"*) o di riga (*scope="row"*). Richiesto per l'accessibilità!
- Per default l'intestazione viene visualizzata in bold e centrata nello spazio della cella.

Esempio-1

- Vogliamo rappresentare la seguente tabella (3x3) di dati con intestazioni:

| | Sabato | Domenica |
|--------------------|--------|----------|
| Biglietti venduti: | 120 | 135 |
| Incasso: | 1020 | 1148 |

```
<table>
  <tr> <th></th> <th scope="col">Sabato</th> <th scope="col">Domenica</th> </tr>
  <tr> <th scope="row">Biglietti venduti: </th> <td>120</td> <td>135</td> </tr>
  <tr> <th scope="row">Incasso: </th> <td>1020</td> <td>1148</td> </tr>
</table>
```

Celle che occupano più colonne

Come descrivere la tabella seguente?

| Name | Telephone | |
|------------|-----------|----------|
| Bill Gates | 55577854 | 55577855 |

Usiamo l'attributo **colspan** (in th) per definire la cella che occupa 2 colonne. **Nota:** in questo esempio e negli esempi seguenti viene usato l'attributo *border* in *table* (con valore: 1) per visualizzare il bordo delle celle. Vedremo più avanti come fare con regole CSS

```
<table border="1">  
  <tr> <th>Name</th> <th colspan="2">Telephone</th> </tr>  
  <tr> <td>Bill Gates</td> <td>55577854</td> <td> 55577855 </td> </tr>  
</table>
```

Celle che occupano più righe

Come descrivere la tabella seguente?

| | |
|-------------------|------------|
| Name: | Bill Gates |
| Telephone: | 55577854 |
| | 55577855 |

Usiamo l'attributo *rowspan* (in th) per definire la cella che occupa 2 righe

```
<table border ="1">  
  <tr> <th>Name: </th> <td >Bill Gates</td> </tr>  
  <tr> <th rowspan="2">Telephone</th> <td>55577854</td> </tr>  
  <tr> <td> 55577855 </td></tr>  
</table>
```


Inserimento di una caption

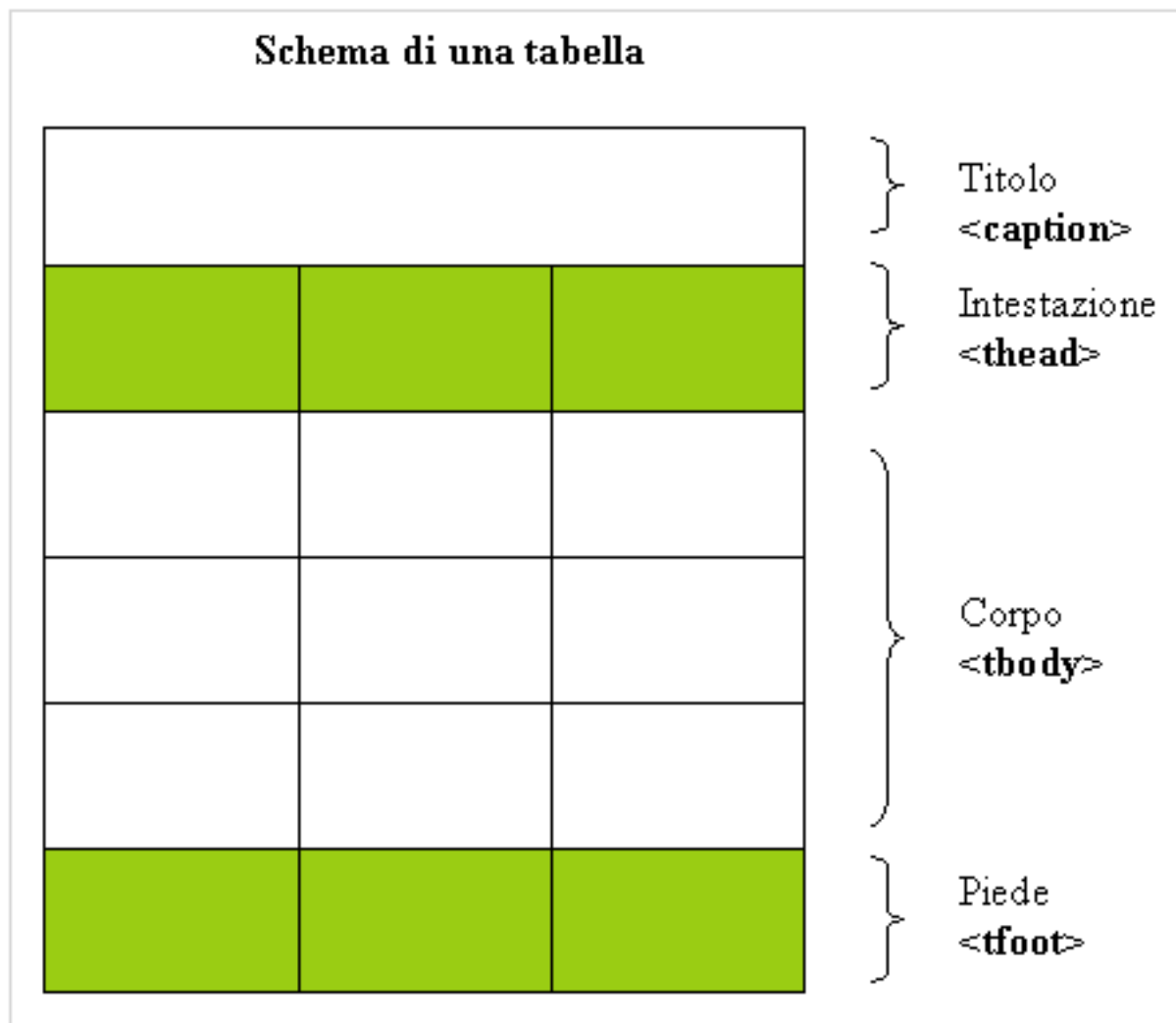
- Si può inserire un titolo per la tabella usando l'elemento *caption* immediatamente dopo l'elemento table. Importante per l'accessibilità
- Esempio precedente con caption:

```
<table border="1">  
  <caption> Telephone numbers </caption>  
    <tr> <th>Name: </th> <td >Bill Gates</td> </tr>  
    <tr> <th rowspan="2">Telephone</th><td>55577854</td> </tr>  
    <tr><td> 55577855 </td></tr>  
</table>
```

Tabelle con molte righe

- Per tabelle con molte righe di dati (ma non solo!) è opportuno usare gli elementi:
 - *thead*: contiene una riga (tr) con le intestazioni (th) delle colonne della tabella
 - *tbody*: contiene le righe (tr) con i dati (td)
 - *tfooter*: contiene una riga con dati (td) riassuntivi
- Questi elementi permettono di raggruppare gli elementi che compongono una tabella in sezioni omogenee in modo da dare più struttura alla tabella e utilizzare le regole CSS per presentare le varie componenti in maniera diversa. Si veda l'esempio seguente

Raggruppamento di righe



| Date | Income | Expenditure |
|--------------|--------|-------------|
| 1st January | 250 | 36 |
| 2nd January | 285 | 48 |
| 3rd January | 260 | 42 |
| 4th January | 290 | 38 |
| 5th January | 310 | 115 |
| 6th January | 168 | 14 |
| 7th January | 226 | 20 |
| 8th January | 253 | 37 |
| 9th January | 294 | 33 |
| 10th January | 216 | 46 |
| 11th January | 244 | 29 |
| 12th January | 297 | 32 |
| 13th January | 328 | 86 |
| 14th January | 215 | 38 |
| 15th January | 254 | 30 |
| 16th January | 256 | 27 |
| 17th January | 311 | 68 |
| 18th January | 212 | 39 |
| 19th January | 234 | 36 |
| 20th January | 221 | 43 |
| 21st January | 259 | 38 |
| 22nd January | 246 | 31 |
| 23rd January | 248 | 17 |
| 24th January | 229 | 45 |
| 25th January | 263 | 34 |
| 26th January | 258 | 41 |
| 27th January | 283 | 22 |
| 28th January | 256 | 30 |
| 29th January | 278 | 47 |
| 30th January | 251 | 15 |
| 31st January | 129 | 64 |
| | 7824 | 1241 |

```

<table>
  <thead>
    <tr>
      <th>Date</th> <th>Income</th> <th>Expenditure</th>
    </tr>
  </thead>

  <tbody> <tr>
    <th>1st January</th> <td>250</td> <td>36</td>
  </tr> <tr>
    <th>2nd January</th> <td>285</td> <td>48</td>
  </tr>
    <!-- additional rows as above -->
    <tr>
      <th>31st January</th> <td>129</td> <td>64</td>
    </tr>
  </tbody>

  <tfoot> <tr>
    <td></td> <td>7824</td> <td>1241</td>
  </tr>
</tfoot>
</table>

```

Come modificare per inserire righe orizzontali?

- Soluzione:

```
<table>
  <thead>
    <tr> <th>Date</th> <th>Income</th> <th>Expenditure</th></tr>
  </thead>

  <tbody>
    <tr> <th colspan="3"> <hr /> </th> </tr>
    <tr> <th>1st January</th> <td>250</td> <td>36</td></tr>
    <tr> <th>2nd January</th> <td>285</td> <td>48</td></tr>
    <!-- additional rows as above -->
    <tr><th>31st January</th> <td>129</td> <td>64</td></tr>
    <tr> <th colspan="3"> <hr /> </th> </tr>
  </tbody>

  <tfoot>
    <tr><td></td> <td>7824</td> <td>1241</td></tr>
  </tfoot>

</table>
```

Altri attributi delle tabelle

- Esistono alcuni attributi dell'elemento *table* che però è meglio non usare nei nuovi siti:
 - *width*: definisce la larghezza della tabella. Si può usare anche per le singole celle (es. per th)
 - *cellpadding*: permette di inserire dello spazio bianco attorno al dato inserito in una cella
 - *cellspacing*: permette di inserire dello spazio bianco tra le celle
 - *border*: permette di visualizzare il bordo della tabella e delle celle e definire lo spessore
 - *bgcolor*: permette di specificare un colore di sfondo per l'intera tabella (o per determinate celle)
- Vedremo più avanti come realizzare questi obiettivi con i fogli di stile

APPROFONDIMENTO SULLE TABELLE

Tabelle annidate

- Vediamo come inserire una tabella all'interno di un'altra tabella. Esempio

| Campo | Tabella | | | | |
|--------|---|-------|-------------|--------|--------------|
| Campo1 | <table><tr><th>Campo</th><th>Descrizione</th></tr><tr><td>Campo1</td><td>Descrizione1</td></tr></table> | Campo | Descrizione | Campo1 | Descrizione1 |
| Campo | Descrizione | | | | |
| Campo1 | Descrizione1 | | | | |

Nota: si usi thead, tbody e tfoot!


```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
</head>
```

```
<body>
```

```
<table border="1">
```

```
  <thead>
```

```
    <tr><th>Campo</th><th>Tabella</th></tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <tr>
```

```
      <td>Campo1</td>
```

```
      <td>
```

```
        <table border="1">
```

```
          <thead><tr><th>Campo</th><th>Descrizione</th></tr></thead>
```

```
          <tbody><tr><td>Campo1</td><td>Descrizione1</td></tr></tbody>
```

```
        </table>
```

```
      </td>
```

```
    </tr>
```

```
  </tbody>
```

```
</table>
```

```
</body>
```

```
</html>
```

Raggruppamento di colonne

Possiamo raggruppare le celle di una tabella per colonne usando l'elemento *colgroup* che deve essere inserito tra la caption e ogni possibile raggruppamento per riga

Con *colgroup* i gruppi vengono creati prendendo le colonne da sinistra verso destra. Il numero di colonne da considerare viene specificato tramite l'attributo *span*.

Possiamo usare l'elemento *col* per specificare singole colonne o sottogruppi di colonne all'interno di un gruppo. Tale elemento può essere figlio solo di un elemento *colgroup* che non abbia impostato l'attributo *span*

Esempio

| | | | | | |
|---------------|-------------|-------------|---------------|-------------|-------------|
| | | | col b1 | col b2 | |
| head | head | head | head | head | head |
| cell | cell | cell | cell | cell | cell |
| cell | cell | cell | cell | cell | cell |
| cell | cell | cell | cell | cell | cell |
| colgroup id=a | | | colgroup id=b | | |

...

```
<table border="1">
```

```
<colgroup id="a" span="3"></colgroup>
```

```
<colgroup id="b" >
```

```
  <col id="b1"></col>
```

```
  <col id="b2" span="2"></col>
```

```
</colgroup>
```

```
<thead>
```

```
<tr> <th> head </th><th> head </th><th> head </th><th> head </th><th> head </th>
```

```
<th> head </th> </tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr> <td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td></tr>
```

```
<tr> <td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td></tr>
```

```
<tr> <td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td><td>cell</td></tr>
```

```
</tbody>
```

```
<table>
```

Bibliografia

- Jon DUCKETT. [HTML & CSS: design and build websites](#). John Wiley & Sons, Inc., Indianapolis, Indiana, 2011. **Capitoli 1,2, 3 e 6**

Fine Parte 1