

# Cosa succede quando si clicca un un link

Le pagine web

## **Autori**

[Damiana Luzzi](mailto:damiana.luzzi@unibo.it) – [damiana.luzzi@unibo.it](mailto:damiana.luzzi@unibo.it)

Dipartimento di Storia Culture Civiltà e Dipartimento delle Arti, Università di Bologna, Bologna, Italia

[Silvio Peroni](mailto:silvio.peroni@unibo.it) – [silvio.peroni@unibo.it](mailto:silvio.peroni@unibo.it)

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia

## **Avviso sul copyright**

Questo lavoro è rilasciato con licenza [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/). Tu sei libero di condividere (riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato) e modificare (remixare, trasformare il materiale e basarti su di esso per le tue opere per qualsiasi fine, anche commerciale) questo lavoro alle seguenti condizioni: attribuzione, ovvero devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale. Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

## Sommario

In questo capitolo verrà introdotto il concetto di marcatura di documenti, e verrà presentato il linguaggio di markup principale usato per la creazione di pagine web, ovvero l'Hypertext Markup Language (HTML).

## Introduzione

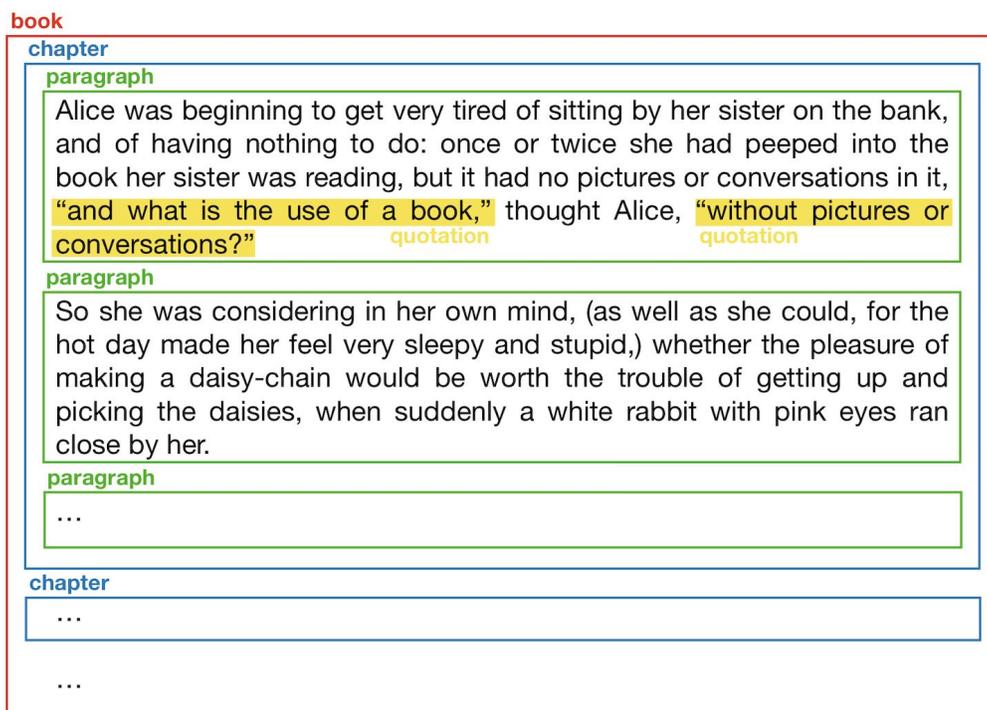
Negli scorsi capitoli abbiamo visto nel dettaglio cosa succede dal momento in cui un utente clicca su un link ipertestuale di una pagina web, a quando la pagina web indicata dal link stesso viene ricevuta e poi visualizzata all'interno del browser. In questo capitolo introdurremo il linguaggio per realizzare le pagine Web e tutti gli elementi che esse contengono: L'Hypertext Markup Language, o HTML. Prima, però, è necessario fare una piccola deviazione per introdurre le teorie che sono alla base di HTML, in modo da presentare cosa sia un linguaggio di markup e a cosa serve.

# I linguaggi di markup

Nel primo capitolo di questo corso, quando abbiamo presentato per la prima volta la struttura dati *albero*, abbiamo introdotto brevemente il concetto di [marcatore del testo](#), ovvero l'annotazione del testo così da definire esplicitamente i vari ruoli strutturali e semantici delle varie parti che lo compongono, come l'identificazione delle sezioni, capoversi, dialoghi, etc. Riprendiamo nuovamente l'esempio tratto dal primo capitolo di *Alice's Adventures in Wonderland* di Lewis Carroll [\[Carroll, 1866\]](#):

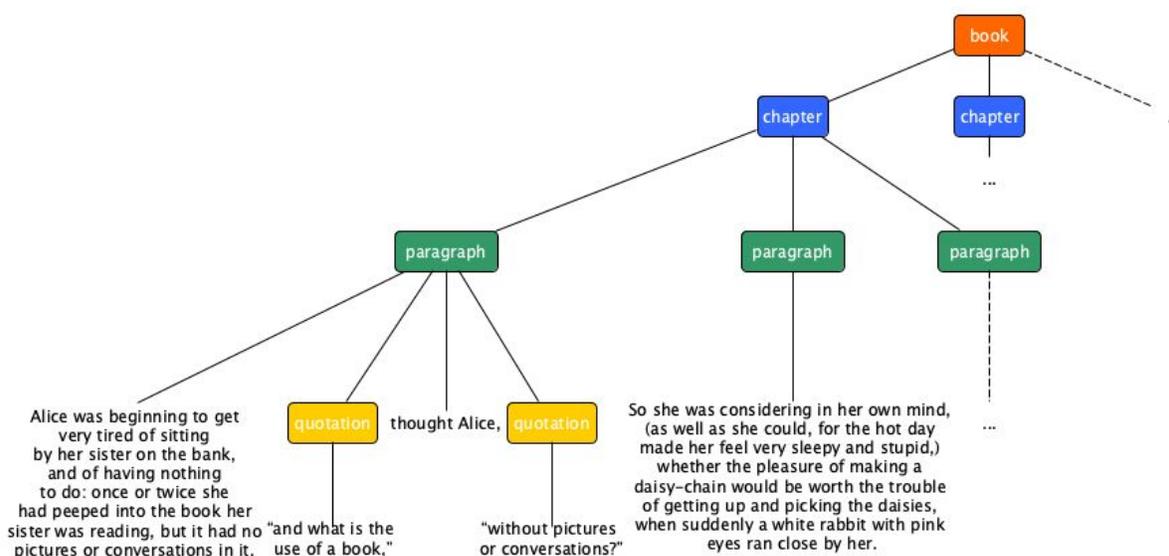
Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.



**Figura 1.** I primi due capoversi di *Alice's Adventure in Wonderland* marcati con strutture testuali di base: libro (*book*), capitolo (*chapter*), capoverso (*paragraph*), e dialogo (*quotation*).

Come abbiamo già anticipato, tutte le strutture che annotano il testo (dialoghi, capoversi, capitoli, etc.), mostrate in [Figura 1](#) per l'esempio di cui sopra, sono organizzate per contenimento, dove la struttura principale, chiamata *book*, è descritta da una sorta di scatola che contiene diverse scatole più piccole chiamate *chapter*, una per ogni capitolo. Ognuna di queste, a sua volta, contiene altre scatole chiamate *paragraph*, una per ogni capoverso, e così via. L'organizzazione a scatole appena presentata descrive un albero (mostrato in [Figura 2](#)), dove la più grande (ovvero *book*) altro non è che la radice dell'albero che contiene i vari *chapter*, e questi a loro volta contengono i *paragraph*, e così via.

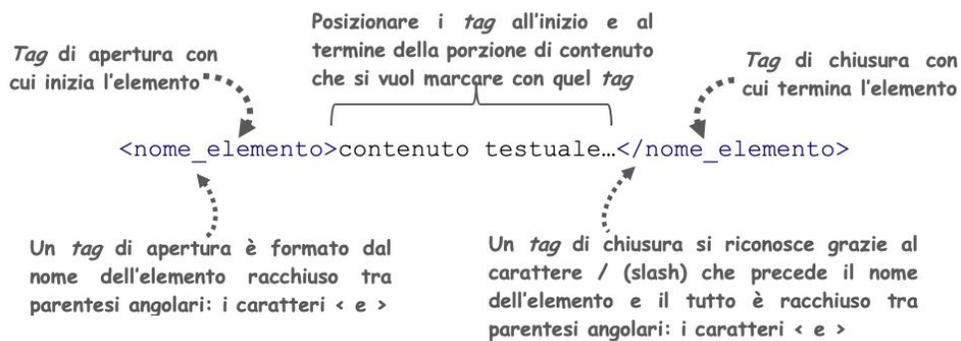


**Figura 2.** L'albero che descrive il contenimento delle varie strutture dell'incipit di *Alice's Adventure in Wonderland*.

In modo da definire formalmente la marcatura relativa a un certo testo, sono stati sviluppati diversi linguaggi di markup. Quelli che hanno maggiore rilevanza nel contesto del Web seguono una sintassi specifica introdotta per la prima volta nello [Standard Generalized Markup Language \(SGML\)](#), rilasciato nel 1986, e poi ripresa dal suo successore, l'[Extensible Markup Language \(XML\)](#), la cui prima versione è datata 1996. SGML e XML sono, di fatto, [metalinguaggi](#), ovvero definiscono le regole sintattiche che devono essere seguite per specificare la marcatura di un testo, ma non impongono alcun vocabolario particolare per il nome da assegnare ai vari marcatori – cosa che invece viene poi fatta quando si definisce un linguaggio di marcatura vero e proprio basato su queste regole sintattiche.

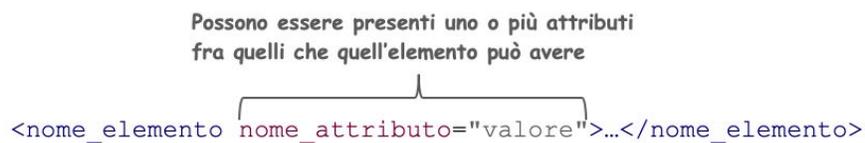
Conformemente alla sintassi di XML, con il termine *elemento* si denota il nome informativo che esprime la semantica della porzione del testo al quale l'elemento si riferisce. Ogni elemento è identificato da due etichette che delimitano il testo in esso contenuto, chiamati *tag*. In particolare, un tag è una sequenza di caratteri che corrisponde al nome dell'elemento collocato all'interno di parentesi angolari "<" e ">". Come mostrato in [Figura 3](#), per ogni elemento esistono

sempre due tag: il tag di apertura, che determina dove inizia l'elemento, e quello di chiusura, che determina dove finisce l'elemento.



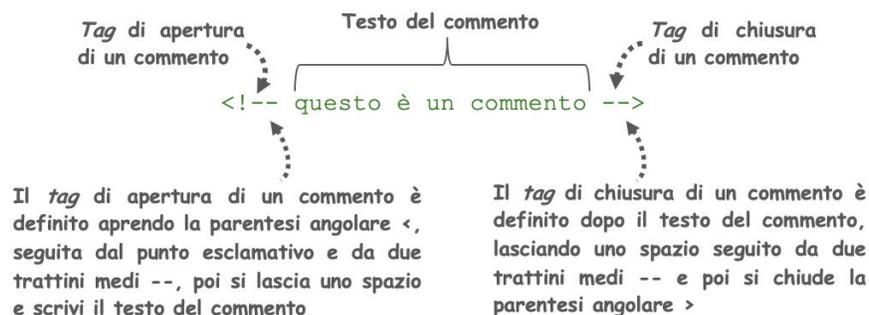
**Figura 3.** I componenti che permettono di definire un elemento di marcatura intorno ad un blocco di testo.

Inoltre, tutti gli elementi, nel loro tag di apertura, possono avere specificati degli attributi nella forma `nome_attributo="valore"`, come mostrato in [Figura 4](#). Un attributo contiene delle informazioni aggiuntive associate a quell'elemento che li specifica. Un elemento può avere nessuno, uno o più attributi.



**Figura 4.** Come specificare un attributo associato ad un elemento.

Infine, è possibile specificare dei commenti, ovvero delle informazioni opzionali che possono venire inserite in qualsiasi punto della documento marcato, ma non fanno parte effettiva del contenuto. I commenti sono solo usati per fornire del contenuto informativo su determinate parti del contenuto. Grazie ai commenti, specificati come mostrati in [Figura 5](#), si possono inserire informazioni, ad esempio sui marcatori utilizzati, sulla struttura e organizzazione del documento di markup.



**Figura 5.** La sintassi per specificare i commenti in un documento marcato.

Un esempio di documento marcato considerando le regole sintattiche appena introdotte e partendo da quello mostrato in [Figura 1](#) è il seguente:

```
<!-- Questo è l'elemento radice, che contiene tutti gli altri -->
<book language="english">
  <chapter>
    <paragraph>
      Alice was beginning to get very tired of sitting by her
      sister on the bank, and of having nothing to do: once or
      twice she had peeped into the book her sister was
reading,
      but it had no pictures or conversations in it,
      <quotation>"and what is the use of a book,"</quotation>
      thought Alice, <quotation>"without pictures or
      conversations?"</quotation>
    </paragraph>
    <paragraph>
      So she was considering in her own mind, (as well as she
      could, for the hot day made her feel very sleepy and
      stupid,) whether the pleasure of making a daisy-chain
      would be worth the trouble of getting up and picking the
      daisies, when suddenly a white rabbit with pink eyes ran
      close by her.
    </paragraph>
    <!-- Qui ci sono altri paragrafi -->
  </chapter>
  <chapter> <!-- Il contenuto del secondo capitolo --> </chapter>
  <!-- Qui ci sono altri capitoli -->
</book>
```

Nella scrittura della marcatura di un documento, come in quella precedente, è buona prassi utilizzare l'indentazione, ovvero l'uso di una sequenza di spazi, per far rientrare il testo quando ci sono elementi contenuti (o annidati) in altri elementi. L'utilizzo di questa tecnica è molto utile, siccome agevola la lettura della marcatura mentre la si sta scrivendo e, soprattutto, quando si devono apportare eventuali modifiche e aggiornamenti.

## Hypertext Markup Language (HTML): cenni storici

L'[Hypertext Markup Language \(HTML\)](#) è una delle tecnologie fondamentali del Web, essendo il linguaggio di markup usato per creare tutte le pagine web e le applicazioni presenti sul Web. Nella sua ultima versione, è un linguaggio che segue una sintassi simile a quella XML

presentata nella sezione precedente, e che mette a disposizione uno specifico vocabolario di elementi ed attributi per identificare i vari ruoli strutturali e semantici di una pagina web. In pratica, ogni qual volta viene fatta una richiesta per una pagina web, viene restituita una copia di un documento HTML contenente opportuni marcatori che il browser è in grado di interpretare e visualizzare a video in qualche modo.

La prima versione di HTML, basata su SGML, è stata creata da Tim Berners-Lee nel 1990 in concomitanza con la creazione del primo browser (WorldWideWeb, introdotto nel capitolo 4), seppur la prima menzione pubblica del linguaggio avviene solo nel 1991. Dal 1994, le specifiche delle varie versioni di HTML sono gestite direttamente dal [World Wide Web Consortium \(W3C\)](#), un'organizzazione internazionale creata da Tim Berners-Lee volta a veicolare e gestire l'evoluzione del Web e delle sue tecnologie.

Proprio in quegli anni, HTML inizia ad avere una fortissima diffusione in seguito ai primi utilizzi commerciali del Web. Negli anni successivi, durante la cosiddetta [prima guerra dei browser tra Netscape e Microsoft](#), le definizioni di nuove specifiche del linguaggio HTML si susseguono, rincorrendo le estensioni e le modifiche proposte dai produttori dei browser che cercano di accaparrarsi quote di mercato.

Negli anni a venire, vengono rilasciate diverse versioni di HTML, fino ad arrivare alla versione 4.01 del 1999. Questa versione, nonostante i miglioramenti apportati dalle precedenti, era piuttosto caotica: alcuni marcatori funzionavano soltanto su alcuni browser o su versioni specifiche di questi, mentre le indicazioni per gestire gli errori di sintassi erano molto generiche e per nulla chiare. In modo da ovviare a questo problema, il W3C decide di iniziare a lavorare ad una nuova versione di HTML basata interamente sulla sintassi XML, chiamata XHTML, pubblicata ufficialmente nel 2000.

La versione di XHTML rilasciata migliora l'interoperabilità con altri linguaggi basati su XML della stessa famiglia, come quello per creare [immagini vettoriali \(Scalable Vector Graphics, o SVG\)](#) e per descrivere formule matematiche ([Mathematical Markup Language, MathML](#)), oltre che a offrire una versione specifica per la telefonia mobile. Questi miglioramenti, seppur lodevoli, erano stati introdotti sacrificando la piena compatibilità con HTML.

Al termine del [W3C Workshop on Web Applications and Compound Documents](#) del giugno 2004, dopo una accesa discussione e successiva votazione, si decide (e solo per una manciata di voti) di non mantenere la retrocompatibilità con le versioni precedenti nello sviluppo della nuova versione di XHTML e, conseguentemente, con HTML stesso. Tuttavia, un gruppo di forti sostenitori di HTML, scontenti del risultato del workshop e supportati dalle aziende principali produttori di browser in circolazione (Apple, Mozilla Foundation, Opera Software, e Google), fonda il [Web Hypertext Application Technology Working Group \(WHATWG\)](#) per proseguire il lavoro di miglioramento di HTML al di fuori del W3C.

Il WHATWG inizia lo sviluppo di una nuova versione di HTML, più orientata allo sviluppo di applicazioni Web così da rendere le pagine ancora più interattive. La comunità Web, inclusi i comuni utenti fino alle aziende, preferiscono seguire la linea dettata WHATWG piuttosto che quella indicata dal W3C, in particolare considerando che i siti web interamente basati su XHTML erano in larga minoranza.

Il 27 ottobre 2006, Tim Berners-Lee ammette pubblicamente, nel post [Reinventing HTML](#) sul proprio blog, di aver sbagliato nel perseguire in modo così netto lo sviluppo della nuova versione di XHTML abbandonando completamente HTML, e annuncia la creazione di un nuovo gruppo di ricerca orientato ad HTML e a quello che sta facendo il WHATWG. Da quel momento in avanti il W3C e il WHATWG collaborano fino al 2011, ma continuano ad avere obiettivi inconciliabili. Da una parte, il W3C vuole cristallizzare e pubblicare la specifica della nuova versione di [HTML5](#) (quella in uso correntemente nel Web) in un documento ufficiale e definitivo, mentre il WHATWG non vuole la pubblicazione di una specifica versione ma piuttosto un *living standard* in evoluzione continua. La disputa si conclude con la pubblicazione di entrambi i documenti: il W3C con la [sua Recommendation](#), e il WHATWG con la [sua specifica in costante aggiornamento](#).

## Elementi HTML principali

Una pagina web è un documento HTML testuale che, attraverso il markup che specifica, permette la strutturazione del contenuto della pagina, la creazione di link ipertestuali, e di riferimenti a immagini e/o altri oggetti multimediali. Tutti questi oggetti marcati vengono interpretati dal browser e visualizzate su schermo in modo appropriato.

Come anticipato, l'operazione di marcatura, o di *tagging*, è un'operazione che permette di associare delle etichette, che definiscono determinati ruoli, alle varie parti del testo. Per portare un esempio, potremmo immaginare di essere in una cucina con un blocco di post-it in mano, dove ogni post-it ha una parola scritta sopra, ad esempio "frigorifero", "caffè", "zucchero", "sale", "mela", "banana", "uva", "corn-flakes", "tavolo", "sedia", "forno", etc. L'azione di etichettare ciascun oggetto della cucina con il rispettivo post-it che lo descrive – associare al frigorifero blu il post-it "frigorifero", alla mela verde il post-it "mela", e così via – è di fatto una marcatura a tutti gli effetti. Scrivere codice HTML equivale a svolgere lo stesso esercizio, ma invece di etichettare oggetti e cibo presente in cucina, si marca, con opportuni elementi identificati da un tag di chiusura e uno di apertura, il contenuto testuale di un documento. Una cosa importante da notare è che i vari post-it descritti sopra hanno etichette come "frigorifero" o "mela", non "frigorifero blu" o "mela verde". Questo perché i post-it servono a descrivere il contenuto dell'oggetto etichettato, ma non a delinearne le caratteristiche esteriori. In modo del tutto simile, gli elementi HTML permettono di descrivere il contenuto della pagina, ma non danno indicazioni dirette su come questo contenuto deve essere presentato a video – anche se c'è uno specifico linguaggio per fare questo chiamato CSS, che introdurremo alla fine di questo capitolo.

Come anticipato, il vocabolario degli elementi e degli attributi che si possono usare per marcare un documento è prestabilito da HTML ed è stato proposto per un uso abbastanza mnemonico. Ad esempio, si usa l'elemento `p` per identificare un paragrafo, `table` per identificare una tabella, `title` per identificare il titolo della pagina, ecc.

## Struttura di base una pagina HTML

Paradossalmente, il primo elemento con il quale inizia un documento HTML non è un elemento HTML, ma è la *Document Type Declaration*, che informa il browser su qual è la versione di HTML che dovrà interpretare. In particolare, `<!DOCTYPE html>` è la dichiarazione che indica al browser che il documento in questione utilizza le regole di marcatura di HTML5.

Il primo elemento, la radice dell'albero, di un documento HTML è `html`. Questo elemento contiene tutti gli altri elementi che descrivono la pagina HTML. Tra i tanti attributi che si possono specificare su questo elementi (ma anche su altri), esiste l'attributo `lang`, che viene usato per definire la lingua principale dei contenuti della pagina HTML. Il valore dell'attributo `lang` è il codice della lingua espresso da due caratteri [Phillips & Davis, 2006]. Alcuni esempi di codici espressi da due caratteri sono "it" (italiano), "en" (inglese), "fr" (francese), e "es" (spagnolo). Ad esempio, se si vuol indicare che il testo del documento HTML è in italiano, basta associare il valore "it" all'attributo `lang` contenuto nel tag di apertura dell'elemento `html`: `<html lang="it">`.

L'elemento `html` contiene due elementi per distinguere due sezioni importanti della struttura base di un documento HTML:

- `head`, le informazioni, tipicamente metadati (come `title`, il titolo del documento) particolarmente utili all'indicizzazione della pagina HTML da parte dei motori di ricerca;
- `body`, il contenuto vero e proprio del documento (descritto da elementi come `header`, `nav`, `h1`, `article`, `section`, `div`, `p`, `a`, `img`, `video`, `table`, `ul`, `ol`, etc.), che verrà visualizzato dal browser.

Per esempio, le strutture fondamentali che regolano l'organizzazione di un documento HTML, che contiene l'estratto di *Alice's Adventure in Wonderland* di Lewis Carroll mostrato precedentemente, sono le seguenti:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Alice's Adventure in Wonderland</title>
  </head>
  <body>
    <!-- il contenuto dell'estratto va qui -->
  </body>
```

```
</html>
```

## Il testo e la sua marcatura

Il contenuto testuale di un documento HTML può essere organizzato mediante l'uso di appropriati elementi HTML, che veicolano una precisa semantica. Nelle prossime sottosezioni presentiamo i più importanti.

### Titoli

I titoli (*heading*) vengono usati per organizzare gerarchicamente il contenuto di un documento HTML. I titoli sono definiti in ordine di importanza, dal più importante al meno importante: la lettera *h* (iniziale del termine inglese *heading*) è seguita da un numero progressivo da 1 (più importante) a 6 (meno importante), che vengono visualizzati dal browser con dimensioni diverse, così da sottolinearne l'importanza, come mostrato in [Figura 6](#).

```
<h1>Titolo di primo livello</h1>  
<h2>Titolo di secondo livello</h2>  
<h3>Titolo di terzo livello</h3>  
<h4>Titolo di quarto livello</h4>  
<h5>Titolo di quinto livello</h5>  
<h6>Titolo di sesto livello</h6>
```



**Figura 6.** Gli elementi usati per definire i titoli del contenuto di un documento HTML, e come sono visualizzati sul browser.

### Paragrafi

I paragrafi (in inglese *paragraph*) permettono la suddivisione di un testo. L'elemento *p* (iniziale del termine inglese *paragraph*) permette di definire paragrafi in HTML. Ad esempio, i due paragrafi dell'estratto di *Alice's Adventures in Wonderland* che abbiamo portato come esempio in precedenza possono essere definiti in HTML come segue, mentre la loro visualizzazione su browser è mostrata in [Figura 7](#).

```
<p>
```

```
Alice was beginning to get very tired of sitting by her  
sister on the bank, and of having nothing to do: once or  
twice she had peeped into the book her sister was reading,  
but it had no pictures or conversations in it,  
"and what is the use of a book,"
```

thought Alice, "without pictures or conversations?"

</p>

<p>

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

</p>

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

**Figura 7.** Come il browser visualizza i due paragrafi dell'estratto del libro.

## Enfasi

Enfatizzare porzioni di testo in HTML si specifica mediante i seguenti due elementi:

- `em` (abbreviazione del termine inglese *emphasized*) leggera enfasi, visualizzata di default in corsivo dal browser;
- `strong` (abbreviazione dei due termini inglesi *strongly emphasized*) enfasi forte, visualizzata di default in grassetto dal browser.

Ovviamente, associare l'enfasi ad una parola o ad una sequenza di parole veicola anche un forte significato semantico per il testo enfattizzato. Un esempio di come questi due elementi possono essere utilizzati è mostrato di seguito, sempre riprendendo l'estratto del libro di Lewis Carroll già presentato. Il modo in cui le enfasi sono mostrate sul browser è descritto in [Figura 8](#).

Alice was beginning to get `<em>very tired</em>` of sitting ...  
... when suddenly a `<strong>white rabbit</strong>` with pink eyes ...

Alice was beginning to get *very tired* of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, “and what is the use of a book,” thought Alice, “without pictures or conversations?”

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a **white rabbit** with pink eyes ran close by her.

**Figura 8.** Due enfasi, una normale ed una forte, specificate sulle parole “very tired” e “white rabbit”.

## Citazioni

HTML mette a disposizione tre differenti elementi per specificare le citazioni in un testo:

- `cite` descrive il titolo di un lavoro (libro, film, canzone, gioco, opera d’arte, rivista, ecc.) che viene citato nel testo e, nel browser, viene tipicamente visualizzato in corsivo;
- `q` indica una citazione in linea inserita all’interno del paragrafo. Differenti versioni di browser potrebbero visualizzare la citazione, marcata con l’elemento `q`, tra virgolette “...” o sergenti «...»;
- `blockquote` indica una citazione situata come blocco a se stante rispetto al paragrafo. Il browser di default indenta la porzione di testo racchiusa dall’elemento `blockquote`, ovvero dà un rientro rispetto al resto del testo.

Il seguente frammento mostra l’uso di tutti e tre gli elementi HTML.

```
<p>
  Riprendiamo nuovamente l’esempio tratto dal primo capitolo
  di <cite>Alice’s Adventures in Wonderland</cite> di
  Lewis Carroll:
</p>
<blockquote>
  <p>
    Alice was beginning to get <em>very tired</em> of sitting by
    her sister on the bank, and of having nothing to do: once or
    twice she had peeped into the book her sister was reading,
    <q>and what is the use of a book,</q>
    thought Alice, <q>without pictures or
    conversations?</q>
  </p>
  <p>
    So she was considering in her own mind, (as well as she
```

```
could, for the hot day made her feel very sleepy and
stupid,) whether the pleasure of making a daisy-chain
would be worth the trouble of getting up and picking the
daisies, when suddenly a <strong>white rabbit</strong>
with pink eyes ran close by her.
</p>
</blockquote>
```

Riprendiamo nuovamente l'esempio tratto dal primo capitolo di *Alice's Adventure in Wonderland* di Lewis Carroll:

Alice was beginning to get *very tired* of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a **white rabbit** with pink eyes ran close by her.

**Figura 9.** Tutti e tre gli elementi HTML che si riferiscono alle citazioni mostrati come vengono visualizzati dal browser.

## Liste

Una lista permette di organizzare dei frammenti di testo in forma di elenco ordinato secondo una numerazione oppure non ordinato. In HTML esistono due elementi che permettono di creare una lista:

- `ol` per la lista ordinata (acronimo dei termini inglese *ordered list*);
- `ul` per la lista non ordinata (acronimo dei termini inglesi *unordered list*).

Le singole voci della lista, sia ordinata che non ordinata, vengono indicate usando l'elemento `li` (acronimo dei termini inglesi *list item*). In particolare, all'interno dell'opportuno elemento lista che vogliamo usare, devono essere presenti tanti elementi `li` quanti sono le voci della lista che vogliamo considerare. Ovviamente, gli elementi `li` possono contenere non solo testo, ma anche altri elementi HTML.

Il frammento qua sotto mostra due liste, la prima ordinata che rappresenta l'indice dei capitoli del libro *Alice's Adventures in Wonderland*, mentre la seconda non ordinata elenca i vari personaggi che compaiono nel libro. La loro visualizzazione su browser è mostrata in [Figura 10](#).

```
<ol>
```

```

<li>Nella conigliera</li>
<li>Lo stagno di lagrime</li>
<li>Corsa scompigliata. Racconto con la coda</li>
<li>...<!-- altri capitoli --></li>
</ol>

<ul>
<li>Alice</li>
<li>Coniglio</li>
<li>Gatto del Cheshire</li>
<li>...<!-- altri personaggi --></li>
</ul>

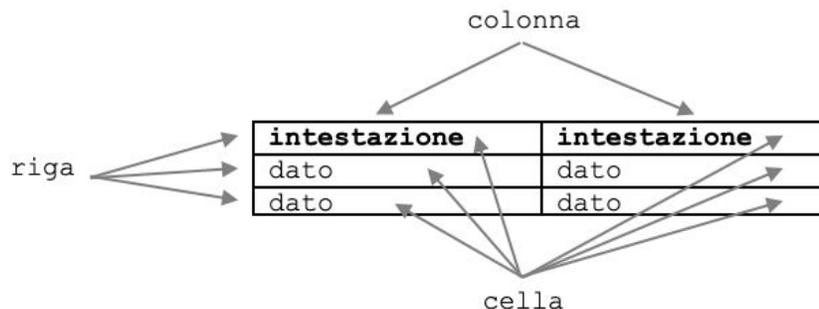
```

1. Nella conigliera
  2. Lo stagno di lagrime
  3. Corsa scompigliata. Racconto con la coda
  4. ...
- Alice
  - Coniglio
  - Gatto del Cheshire
  - ...

**Figura 10.** La visualizzazione delle due liste introdotte nel testo sul browser.

## Tabelle

In HTML, una tabella (elemento `table`) è composta da una sequenza di righe (elemento `tr`, per *table row*), ognuna delle quali contiene una o più celle che possono essere di due tipi: celle contenenti dati (`td`, per *table data*) o celle che definiscono intestazioni per le righe o le colonne identificate dalla tabella (`th`, per *table heading*). Seppur la tabella sia organizzata in righe, ovviamente tutte le celle nella medesima posizione di ogni riga identificano, in maniera astratta, una particolare colonna della tabella, come mostrato in [Figura 11](#).



**Figura 11.** Una tabella e la sua scomposizione in righe, celle, e colonne.

L'esempio di seguito mostra una tabella di quattro righe, in cui la prima ha la funzione di intestazione, contenente alcuni personaggi del libro *Alice's Adventures in Wonderland* e il capitolo dove appaiono per la prima volta. La visualizzazione della tabella è mostrata in [Figura 12](#).

```
<table>
  <tr><th>Personaggio</th><th>Capitolo</th></tr>
  <tr><td>Alice</td><td>Capitolo 1</td></tr>
  <tr><td>Coniglio</td><td>Capitolo 1</td></tr>
  <tr><td>Gatto del Cheshire</td><td>Capitolo 6</td></tr>
</table>
```

Personaggio	Capitolo
Alice	Capitolo 1
Coniglio	Capitolo 1
Gatto del Cheshire	Capitolo 6

**Figura 12.** Una visualizzazione della tabella in HTML sul browser. I bordi non sono mostrati di default e, nel caso si vogliano vedere, bisogna specificarlo esplicitamente mediante l'utilizzo dell'attributo `border` dell'elemento `table` (ad esempio `<table border="1">`).

## Collegamenti ipertestuali

I collegamenti ipertestuali (o link) sono il meccanismo utilizzato per creare l'ipertesto che è sotteso dalle varie pagine Web esistenti. Ogni risorsa sul Web, sia essa un documento HTML, un'immagine, un video, un audio, etc., senza l'esistenza dei link sarebbe separata dalle altre risorse. L'intero Web, definendo di fatto un ipertesto, si basa pesantemente sui link, che possono essere cliccati per "navigare" verso un'altra pagina Web, un'altra sezione della stessa pagina Web, un'altra immagine, un altro video, etc.

Il link definibile in HTML si compone di due parti principali:

- l'**etichetta** del link, ovvero quella porzione del contenuto cliccabile, solitamente evidenziata in blu con una sottolineatura, che permette l'identificazione visuale del link sul browser;
- la **destinazione**, ovvero l'URL della particolare risorsa che si vuole "raggiungere" una volta che si è cliccato sull'etichetta.

In HTML, l'elemento per creare un link è `a` (abbreviazione del termine inglese *anchor*, ancora in italiano) accompagnato:

- dall'attributo `href` (abbreviazione dei due termini inglesi *hypertext reference*) che ne definisce la destinazione;
- dal testo (e/o da altri elementi HTML) che contiene, che ne definiscono l'etichetta.

Ad esempio, il link alla versione di Wikisource del libro *Alice's Adventures in Wonderland*, introdotta negli esempi precedenti, può essere definito come segue (e viene visualizzato sul browser come mostrato in [Figura 13](#)):

```
Riprendiamo nuovamente l'esempio tratto dal primo capitolo di <a href="https://en.wikisource.org/wiki/Alice%27s_Adventures_in_Wonderland_(1866)"><cite>Alice's Adventure in Wonderland</cite> di Lewis Carroll</a>: ...
```

Riprendiamo nuovamente l'esempio tratto dal primo capitolo di [Alice's Adventure in Wonderland di Lewis Carroll](#):

Alice was beginning to get *very tired* of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

**Figura 13.** La visualizzazione di un link ipertestuale sul browser. Se la pagina relativa al link è stata già visitata, solitamente il link assume il colore viola.

Un link si dice *interno* alla pagina quando la sua destinazione è un'area dello stesso documento HTML che lo definisce. In questo caso, per definire l'area precisa a cui si vuole arrivare a seguito di un click su un link interno, bisogna usare l'attributo `id`, con valore una stringa alfabetica a piacere (ad esempio `id="capitolo_1"`), sull'elemento HTML su cui si vuole arrivare a seguito del link. Fatto questo, l'URL della destinazione include soltanto la parte dedicata al frammento, ovvero il carattere `#` seguito dalla stringa associata al valore dell'attributo `id` dell'elemento a cui si vuole arrivare (ad esempio `<a href="#capitolo_1">vai al primo capitolo</a>`).

## Multimedia

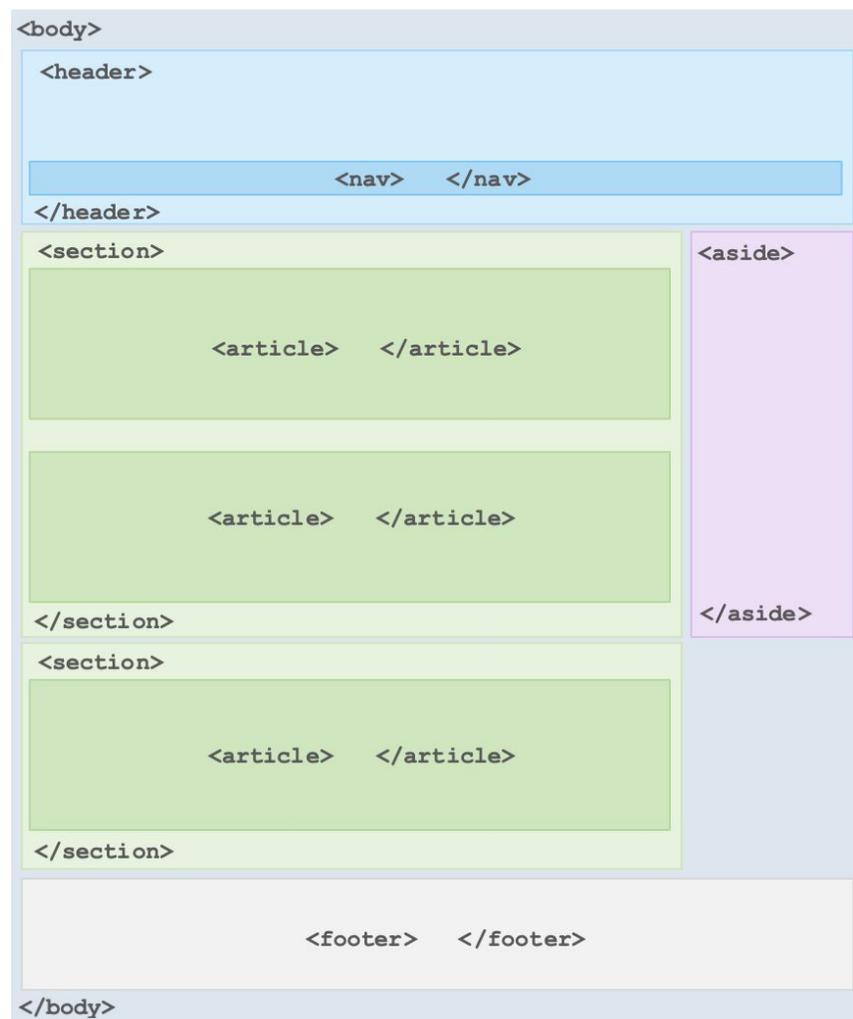
Le immagini – logo, fotografie, disegni, grafici, ecc. – sono parte importante di un documento HTML. L'elemento HTML per specificare un'immagine è `img`, un elemento che non contiene alcun testo ma che deve specificare i seguenti due attributi:

- `src` (abbreviazione del termine inglese *source*), il cui valore indica l'URL dove reperire l'immagine;
- `alt` (abbreviazione dei due termini inglesi *alternative text*) il cui valore indica un testo alternativo mostrato al posto dell'immagine se quest'ultima, per qualsiasi ragione, non è visualizzabile.

In modo analogo, l'elemento `video` usato insieme all'elemento `source` permettono di inserire un video in un documento HTML. L'elemento `source` deve avere specificato l'attributo `src` che, come per le immagini, si riferisce all'URL dove reperire il video.

## Il contenuto di una pagina web

HTML permette di specificare le varie aree logiche di una pagina Web a seconda dei contenuti che queste contengono. Per esempio, un documento può essere suddiviso in diverse sezioni, come la testata, il piè di pagina, una barra laterale di navigazione, etc. In [Figura 14](#) viene mostrato un possibile modello per l'organizzazione delle varie parti di un documento HTML, in modo da rendere meglio l'idea di quali sono le varie aree logiche che si possono utilizzare, e come si potrebbero posizionare.



**Figura 14.** Un possibile modello per organizzare il contenuto di una pagina HTML.

Gli elementi introdotti in [Figura 14](#) sono i seguenti:

- l'elemento `header` contrassegna l'area della testata della pagina e raggruppa gli elementi introduttivi e di ausilio alla navigazione, ad esempio il titolo del sito Web, il logo, il menu di navigazione, il campo di ricerca, etc.:

- l'elemento `footer` individua l'area del "piè di pagina" dove si inseriscono informazioni sul copyright della pagina, i contatti, l'anno di pubblicazione della pagina, etc.;
- l'elemento `section` rappresenta una sezione generica per organizzare, correlare e contraddistinguere, da un punto di vista tematico, i contenuti. La sua funzione è simile a quella dei capitoli di un libro o delle sezioni di una tesi di laurea. Al suo interno puoi avere, ad esempio: articoli, titoli, paragrafi, liste, tabelle, immagini, video, audio, etc.;
- l'elemento `article` indica un blocco di contenuto indipendente dal contesto nel quale si trova, un'entità a sé stante e collocabile anche in un'altra posizione. Il significato semantico dell'elemento `article` è inteso in senso ampio: un insieme di informazioni che abbiano un significato compiuto e che non necessitano di altri elementi per essere comprese. Il contenuto del blocco marcato con `article` può essere l'articolo di una rivista o di un quotidiano, il post di un blog, o, più in generale, un qualsiasi blocco di contenuto che sia indipendente e abbia un significato proprio anche se estratto dal contesto della pagina.
- l'elemento `nav` individua l'area con i link (o voci) che guidano l'utente nella navigazione del sito Web;
- l'elemento `aside` individua l'area i cui contenuti sono in parte in relazione, arricchiscono o approfondiscono, i contenuti principali, ad esempio i link di navigazione secondari, i banner pubblicitari, le categorie di un blog, etc. Solitamente tale area è posizionata a lato, per questo è chiamata anche "barra laterale", ma nulla vieta di posizionarla in altre aree.

## Separazione tra markup e presentazione: i CSS

Nel 1996 viene rilasciata per la prima volta la specifica del [Cascading Style Sheet \(CSS\)](#), ovvero un ulteriore linguaggio che permette di definire il livello presentazionale degli elementi di HTML. I CSS definiscono l'aspetto grafico dei documenti HTML, ad esempio permettendo di specificare:

- il colore e la dimensione del testo contenuto nei titoli e nei paragrafi;
- la posizione di sezioni, articoli, immagini o video;
- l'animazione dinamica quando si posiziona il cursore del mouse su una voce del menu e/o sui link;
- etc.

CSS è stato creato per permettere la netta separazione tra la struttura del contenuto, definito dagli elementi del documento HTML, e come questo contenuto deve apparire o essere visualizzato sul browser, definito dal CSS. Ogni documento CSS è descritto usando uno specifico linguaggio con una sintassi completamente diversa da HTML. In particolare, CSS permette di definire un insieme di proprietà presentazionali (colore, font, posizione, ecc.) su determinati elementi HTML. Ad esempio, per assegnare il colore grigio e l'allineamento

giustificato a tutti gli elementi `p` (*paragraph*) della pagina HTML basta definire la seguente regola:

```
p {  
    text-align: justify;  
    color: grey;  
}
```

Nell'esempio, `p` è il *selettore* che si riferisce al relativo elemento HTML `p`. Il blocco delle regole o dichiarazione di stile associate a quel selettore sono incluse tra parentesi graffe `{ }`. All'interno delle parentesi si possono indicare una o più coppie `proprietà:valore` (separate da dei punti e virgola) che permettono di definire regole di visualizzazione degli elementi selezionati.

## Ringraziamenti

Gli autori ringraziano Eugenia Galli per le preziose correzioni proposte al testo.

## Bibliografia

Carroll, L. (1866). Alice's Adventures in Wonderland. Macmillan and Co. Available at [https://en.wikisource.org/wiki/Alice%27s\\_Adventures\\_in\\_Wonderland\\_\(1866\)](https://en.wikisource.org/wiki/Alice%27s_Adventures_in_Wonderland_(1866)) (last visited 8 December 2018)

Phillips, A., & Davis, M. (2006). Matching of Language Tags (Request For Comment No. RFC4647). Internet Engineering Task Force. <https://doi.org/10.17487/rfc4647>