

HTML 5

HTML

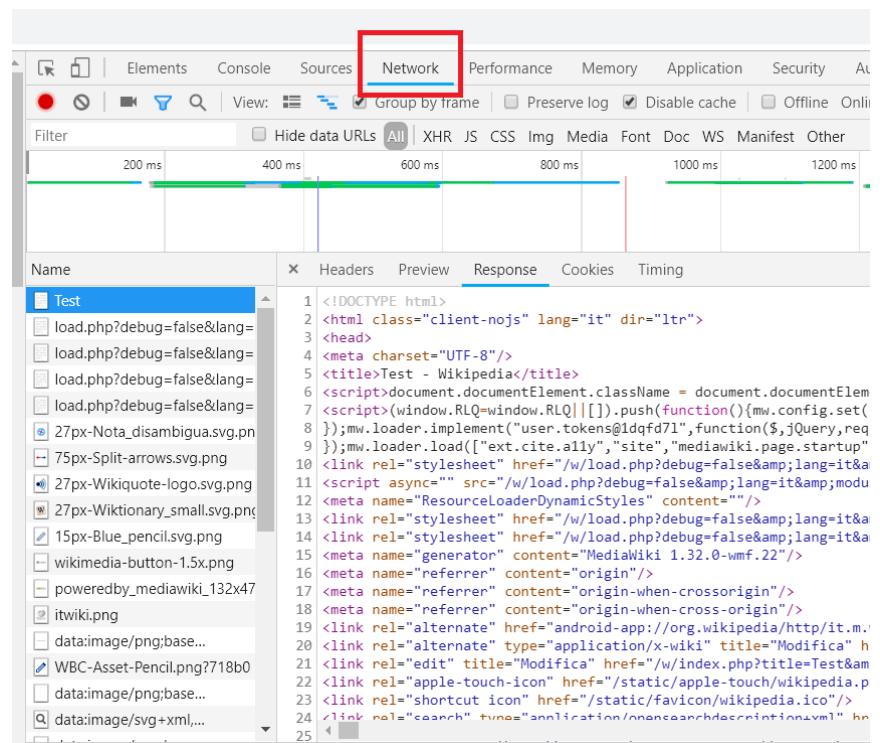
- Acronimo di hypertext markup language
- Inventato da Tim Berners-Lee negli anni '80 per condividere documenti su internet
- Permette di condividere documenti su server attraverso un browser
- è un linguaggio a tag tramite cui il browser interpreta e renderizza il contenuto
- separa il contenuto dalla presentazione
- usa i tag per indicare il tipo di contenuto
- La sua sintassi è definita dal World Wide Web Consortium (W3C)
 - Il World Wide Web Consortium (W3C) sviluppa standard e linee guida per aiutare tutti a costruire un web basato sui principi di accessibilità, internazionalizzazione, privacy e sicurezza.

Seppure molto potente, **HTML non è un linguaggio complesso** e può offrire soddisfazioni immediate.

estensione .html

Estensione .html

È la tipica estensione dell'html, se un file con estensione html viene aperto nel browser diviene leggibile



The screenshot shows the Network tab of the Google Chrome Developer Tools. A red box highlights the 'Network' tab at the top. Below it, there's a timeline with markers at 200 ms, 400 ms, 600 ms, 800 ms, 1000 ms, and 1200 ms. The main area displays a list of network requests. One request, 'Test', is selected and expanded, showing its details. The 'Response' tab is active, displaying the HTML code of the page. The code starts with the standard DOCTYPE declaration and includes meta tags for charset and directionality, followed by a title, script tags for configuration, and various links and stylesheets.

```
1 <!DOCTYPE html>
2 <html class="client-nojs" lang="it" dir="ltr">
3 <head>
4 <meta charset="UTF-8"/>
5 <title>Test - Wikipedia</title>
6 <script>document.documentElement.className = document.documentElement.className + 'client-nojs';</script>
7 <script>(window.RLQ=window.RLQ||[]).push(function(){mw.config.set(
8   'wgToken', mw.util.getParamValue('token') || ''
9 );mw.loader.implement("user.tokens$1dqfd7l",function($,jQuery,req,
10   res){mw.loader.load(["ext.cite.al1y","site","mediawiki.page.startup"]);
11   var $el=$.parseHTML(res);
12   $el.find('img').each(function(){
13     if(this['src']) $(this).attr('src',this['src'].replace(/\?debug=true/,''));
14   });
15   $el.find('link[rel="stylesheet"]').each(function(){
16     var href=$(this).attr('href');
17     if(href) $(this).attr('href',href.replace(/\?debug=true/,''));
18   });
19   $el.find('script').each(function(){
20     var src=$(this).attr('src');
21     if(src) $(this).attr('src',src.replace(/\?debug=true/,''));
22   });
23   $el.find('meta[content]').each(function(){
24     var content=$(this).attr('content');
25     if(content) $(this).attr('content',content.replace(/\?debug=true/,''));
26   });
27   $el.find('link[rel="apple-touch-icon"]').each(function(){
28     var href=$(this).attr('href');
29     if(href) $(this).attr('href',href.replace(/\?debug=true/,''));
30   });
31   $el.find('link[rel="shortcut icon"]').each(function(){
32     var href=$(this).attr('href');
33     if(href) $(this).attr('href',href.replace(/\?debug=true/,''));
34   });
35   $el.find('link[rel="search"]').each(function(){
36     var type=$(this).attr('type');
37     if(type) $(this).attr('type',type.replace(/\?debug=true/,''));
38   });
39   $el.find('hr').each(function(){
40     $(this).attr('style','border-top: 1px solid black; margin-top: 10px; margin-bottom: 10px;');
41   });
42   $el.find('table').each(function(){
43     $(this).attr('style','border-collapse: collapse; width: 100%;');
44   });
45   $el.find('tr').each(function(){
46     $(this).attr('style','border-bottom: 1px solid black; padding: 5px 0; margin: 0;');});});});</script>
46 <script>mw.loader.load(['ext.cite.al1y','site','mediawiki.page.startup']);</script>
47 <link rel="stylesheet" href="/w/load.php?debug=false&lang=it&modules=ResourceLoaderDynamicStyles" content=""/>
48 <link rel="stylesheet" href="/w/load.php?debug=false&lang=it&modules=mediawiki.styles&skin=vector&only=css&version=1.32.0-wmf.22"/>
49 <meta name="generator" content="MediaWiki 1.32.0-wmf.22"/>
50 <meta name="referrer" content="origin"/>
51 <meta name="referrer" content="origin-when-crossorigin"/>
52 <meta name="referrer" content="origin-when-cross-origin"/>
53 <link rel="alternate" href="https://it.wikipedia.org/w/index.php?title=Modifica&oldid=71800000" title="Modifica" hreflang="it"/>
54 <link rel="edit" title="Modifica" href="/w/index.php?title=Test&oldid=71800000" hreflang="it"/>
55 <link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png" hreflang="it"/>
56 <link rel="shortcut icon" href="/static/favicon/wikipedia.ico" hreflang="it"/>
57 <link rel="search" type="application/opensearchdescription+xml" hreflang="it"/>
```

Editor di testo

Editor	Piattaforma	Licenza
Atom.io	Windows, Linux, Mac	Gratuita
Visual Studio Code	Windows, Linux, Mac	Gratis
Notepad++	Windows	Gratuita

La scelta dell'editor (IDE = Integrated Development Environment) è piuttosto importante, è il principale compagno di strada in questo percorso e tipicamente diventa un'estensione naturale del pensiero. Gli editor che risultano tra i più utilizzati sono:

Syntax Highlighting

Serve a **evidenziare la sintassi del linguaggio e rendere più leggibile il codice sorgente**. Ogni tag, attributo, numero o parola chiave assume uno specifico colore e diventa subito riconoscibile.

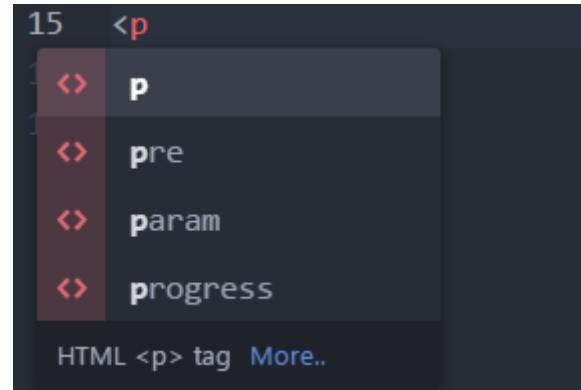
Grazie a questa caratteristica emergono più facilmente eventuali errori di battitura e diventa anche più chiara la struttura della pagina

```
1  <!doctype html>
2  <html lang="it">
3  <head><title>Ciao Mondo!</title></head>
4  <body>
5    <h1>Ciao Mondo!</h1>
6    <p>Questa è la nostra prima pagina HTML!</p>
```

Autocompletamento

Mentre scriviamo il codice l'editor propone di completare automaticamente la parola o il comando che stiamo digitando. Ci sono diversi livelli di autocompletamento anche in base ai linguaggi.

Visual Studio Code ha gli emmet snippets integrati, non sono necessarie estensioni aggiuntive



da dove copiare il testo

<https://lipsum.com/>

<https://it.lipsum.com/>

serve per prendere del testo da copiare

Lorem Ipsum

"Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consecetur, adipisci velit..."

"There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain..."

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel elit eget erat placerat bibendum et eget diam. Suspendisse potenti. Vivamus volutpat pharetra nulla. Aenean commodo felis eget risus rutrum convallis. Vivamus et urna pharetra lorem laoreet auctor. Cras vel cursus leo, at consectetur libero. Vestibulum nunc enim, ornare ut malesuada id, egestas sed lectus. Ut tincidunt hendrerit sapien eget luctus. Morbi ac lorem nisl. Aliquam non elementum ligula, quis vehicula libero. Mauris varius gravida quam, vitae lobortis tortor facilisis vitae. Nunc pretium libero turpis, eu congue mauris pharetra quis. Mauris at justo id arcu bibendum tempus at vitae lacus. Donec viverra, ligula sit amet imperdiet varius, risus dolor vestibulum ex, id posuere tortor nisl nec leo.

 Morbi porttitor imperdiet pharetra. Maecenas quis ex ultrices, consectetur erat in, laoreet justo. Nunc commodo nibh eu augue tincidunt commodo. Sed tempor porttitor tortor eget ultricies. Praesent ligula nulla, consectetur a mollis nec, efficitur elementum libero. Pellentesque eget nulla quis quam scelerisque commodo sit amet non nunc. Morbi sed feugiat ipsum. Duis aliquam ex vel commodo venenatis. Nunc interdum, risus eu fermentum auctor, tellus velit bibendum urna, quis molestie ipsum metus eu augue. Praesent quis lorem ex. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae: Nulla placerat aliquet luctus. Aenean nec velit ut neque accumsan semper. In a tellus sit amet libero varius finibus. Mauris ipsum nisi, volutpat vel euismod sit amet, feugiat ut turpis. Quisque scelerisque mattis metus sed hendrerit.

 Vivamus quis magna dictum, hendrerit justo id, tincidunt est. Vestibulum vitae vehicula lorem, ut porta velit. Nulla ac lectus convallis, egestas nunc at, hendrerit turpis. Mauris non arcu pellentesque, porta purus quis, efficitur ante. Phasellus vitae velit congue dolor suscipit lacinia. Donec pretium erat vitae volutpat sollicitudin. Sed id tellus sit amet tortor placerat fringilla. Etiam dapibus, quam et interdum varius, dui erat pellentesque nunc, eu ultricies arcu neque quis dolor. Ut facilisis vitae diam quis varius. Vestibulum purus odio, pretium vitae condimentum ac, ultrices in nibh.

 Sed feugiat id ante eget condimentum. Etiam mollis justo a dui mollis condimentum. Phasellus ornare dapibus commodo. Suspendisse potenti. Donec sagittis porta vestibulum. Vestibulum venenatis eros id tristique ultricies. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Quisque in lobortis nisi, molestie semper nisl. Maecenas molestie mauris est, a finibus metus vehicula et. Integer pulvinar pellentesque tempus. Pellentesque pellentesque justo at elementum lobortis. Praesent tincidunt, nulla eu ultrices lobortis, ex lectus molestie nisi, sit amet bibendum nulla tellus nec magna. Aenean laoreet tempus facilisis. Ut vitae interdum est, in accumsan mauris. Fusce venenatis efficitur nisl, sed mollis urna venenatis eu. Vivamus id massa erat.

Browser

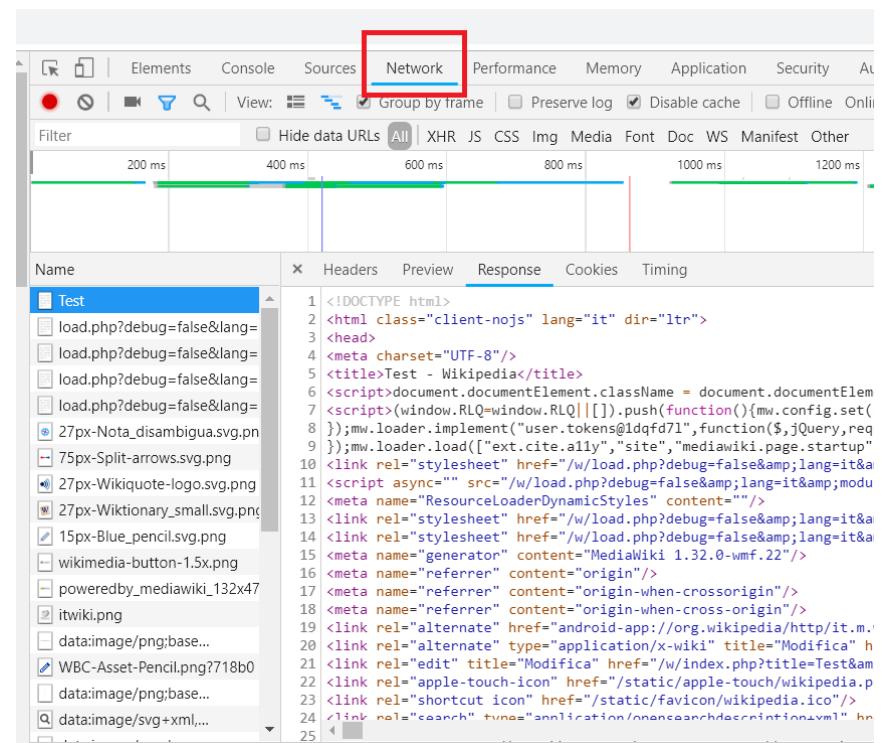
Il programma che **interpreta** il markup delle nostre pagine **HTML** e mostra a video tutto ciò che indichiamo si chiama **Browser**

Tra i compiti svolti dal browser, sono due quelli che ci interessano di più: il caricamento e la visualizzazione (rendering) della pagina HTML.

La fase di caricamento della pagina è il primo punto cruciale quando si parla di perfomance: perché un sito o una applicazione risultino gradevoli da utilizzare bisognerà fare in modo di minimizzare quanto più possibile i **tempi di caricamento**.

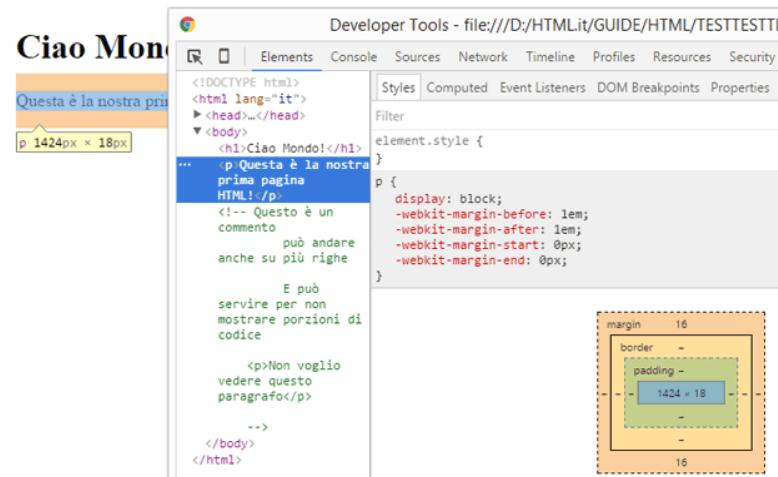
Rendering della pagina

La seconda fase cruciale è quella della visualizzazione ovvero del cosiddetto *rendering* della pagina, la fase in cui il browser interpreta i documenti HTML e dispone sullo schermo gli elementi (testi, immagini, filmati) a seconda delle indicazioni ivi contenute



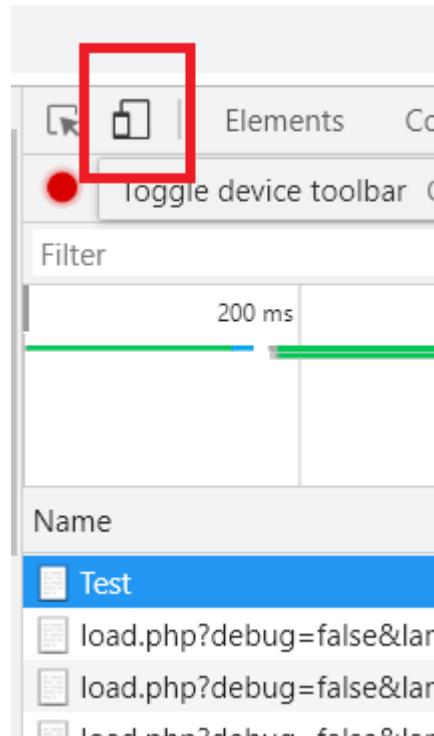
Strumenti di sviluppo del browser

Il browser è fondamentale per testare il nostro lavoro sotto il profilo visuale e non solo. I browser moderni come Google Chrome, Mozilla Firefox, Microsoft Edge e Apple Safari (per non fare torto a nessuno), forniscono infatti i cosiddetti “**strumenti di sviluppo**” grazie ai quali possiamo tenere sotto controllo diversi aspetti delle pagine HTML



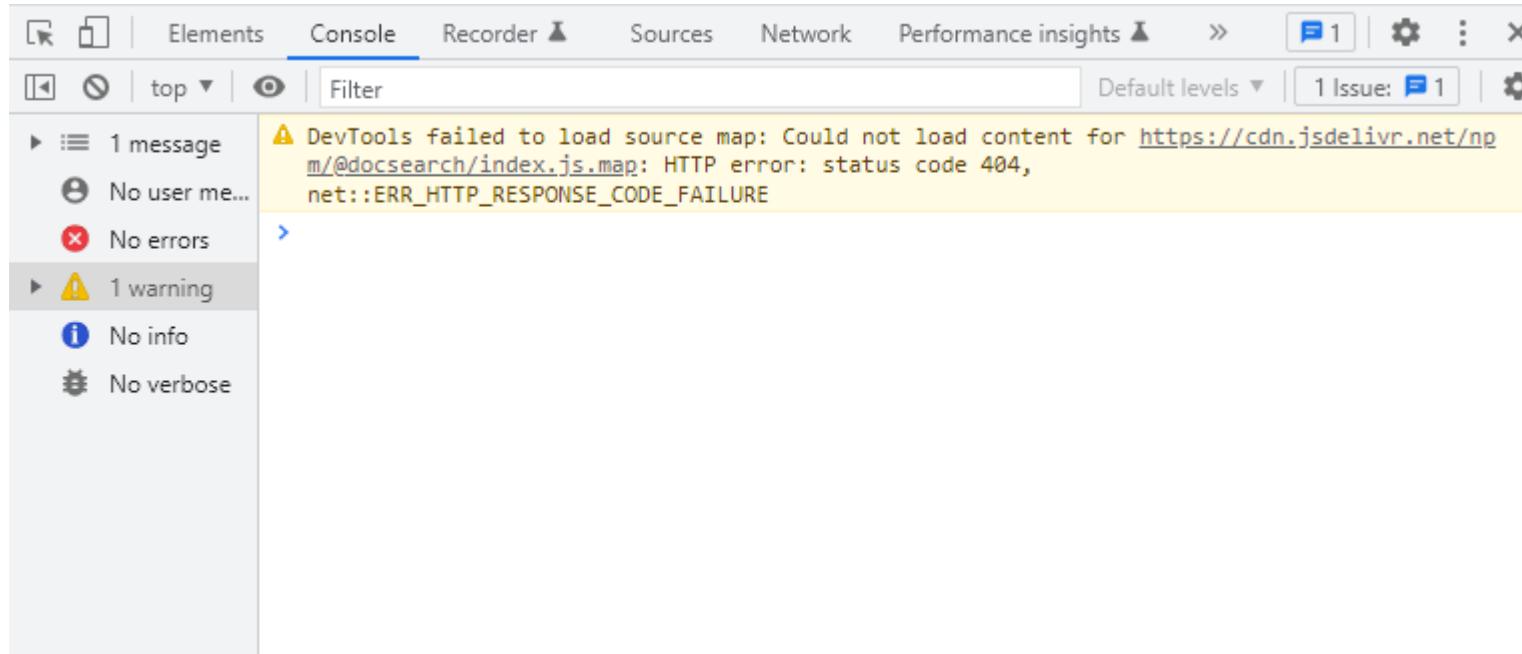
Mobile View

È una modalità che ci consente di effettuare **una prima e rapida valutazione della resa grafica e della user experience che otterremo sui dispositivi mobili.**



Console

Molto utile per interventi rapidi: per tracciare variabili, testare piccole porzioni di script o ricordare al volo le proprietà e i metodi di un oggetto.



I browser non sono tutti uguali

Anche se praticamente tutti i browser moderni garantiscono una pressoché totale aderenza agli standard del W3C, **rimangono ancora alcune piccole differenze nell'implementazione delle direttive**, quindi nel modo in cui i singoli browser mostrano i contenuti.

In sostanza quando lavoriamo al nostro HTML, possiamo trovarci a fare i conti con:

- la compatibilità dei browser** (specie quelli più datati possono dare problemi);
- le capacità dei dispositivi** (in termini di **performance** e banda ad esempio);
- le tipologie di schermo** cui sono destinate le nostre pagine HTML.

Elementi e tag in HTML

In una pagina HTML tutti gli elementi sono connotati da **tag** (letteralmente “etichette”). Si tratta di marcatori che evidenziano non tanto l’aspetto, quanto il senso, il ruolo, o l’organizzazione che vogliamo assegnare ai contenuti

Come è fatto un tag

Un tag è una keyword del linguaggio racchiusa tra parentesi angolari (<>). Esempi di tag sono <html>, <body>, <h1>...

I tag HTML non sono “case sensitive” ciò significa che scrivere <head> o <HEAD> è esattamente la stessa cosa. In ogni caso la consuetudine è quella di scrivere i tag in minuscolo (era una raccomandazione in HTML4 e una richiesta in XHTML).

In HTML5 è consentito scrivere un tag usando anche il maiuscolo

Alcuni TAG sono autochiudenti (non c'è bisogno di mettere la chiusura) ad es.:

 TAG AUTOCHIUDENTE
<div></div> TAG CON CHIUSURA

Elementi contenitori e autonomi, il DOM

Molti elementi in HTML servono per descrivere porzioni di pagina, aree, o contenuti. Ad esempio <body> descrive il contenuto di tutta la pagina, <h1> racchiude un titolo e <p> denota un paragrafo nel testo.

Pertanto **un elemento HTML è quasi sempre un contenitore** e il suo contenuto è delimitato da:

tag di apertura (es. <p>);
tag di chiusura (es. </p>).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p></p>
</body>
</html>
```

Indentare il codice

Spesso per sottolineare l'annidamento di un elemento in un altro si usa “**indentare il codice**”, ovvero discostare il contenuto dall'inizio della riga lasciando spazi (o tab). In pratica apertura e chiusura del tag si trovano allo stesso livello, mentre il contenuto viene spostato verso destra di un tab.

comando rapido: Shift + Alt + F

<body>

- <header class="principale ">
 - <section class="boxed ">
 - <section class="col-3 ">
 - <p class="logo ">Logo</p>
 - </section>
 - </section>

.....

```
<body>
  <header class="principale">
    <section class="boxed">
      <section class="col-3">
        <p class="logo">Logo</p>
      </section>
    </section>
  </header>
</body>
```

I tag <head> e <body>

la struttura minimale della pagina HTML prevede di usare <html>, <head> e <body>

<!DOCTYPE html>

Serve semplicemente a dire che **il file è una pagina HTML** e in particolare che si tratta di un documento secondo lo standard HTML5.

- <**html**> È il tag che racchiude tutta la pagina e ci permette ad esempio di definire quale sia la lingua della pagina
- <**head**> Questo tag contiene una serie di informazioni utili (metadata) per la gestione della pagina. Questi dati riguardano solitamente il titolo della pagina, il set di caratteri, stili, scripts e altre informazioni.
- <**body**> Contiene gli elementi della pagina, tutto il contenuto e tutti i relativi tag che saranno poi resi a video vengono inseriti qui dentro, come abbiamo fatto nel nostro caso con il titolo e il paragrafo.

```
<!doctype html>
<html lang="it">
<head>
<meta charset="utf-8">
<title>Ciao Mondo!</title>
</head>
<body>
  <h1>Ciao Mondo!</h1>
  <p>Questa è la nostra prima
pagina HTML!</p>
</body>
</html>
```

Charset

Per visualizzare correttamente una pagina HTML, un browser Web deve conoscere il set di caratteri utilizzato nella pagina.

Questo è specificato nel tag
<meta>

Prima: Ascii=128 caratteri

Poi: ISO-8859-1: 256 caratteri

Poi Ansi: Windows-1252:256 + 32 caratteri

Ora: UTF-8: copre la maggior parte dei caratteri mondiali

For HTML4:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

For HTML5:

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
```

I meta tag

Sono una serie di parole chiave e descrizioni, che aiutano i motori di ricerca a classificare il sito. Sono racchiusi nel tag <head>

```
<title>Titolo della pagina</title>
```

```
<meta name="description" content="Description della pagina">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta name="keywords" content="html,aaa,bbb">  
(keywords oramai in disuso)
```

```
<meta name="author" content="Mauro">
```

<meta> non sono visualizzati nella pagina web ma vengono usati dal browser, dai motori di ricerca e da altri web services. Di solito riguardano la descrizione della pagina, l'autore, il viewport, il charset....

Favicon

nel tag head

dimensione 16x16 o 32x32

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



Metadata content

Fanno parte di questa categoria tutti gli elementi **utili alla definizione del documento nel suo insieme**: a livello di presentazione o di funzionamento

Base

(specifica l'URL di base e/o l'obiettivo di tutti gli URL relativi in una pagina. Solo un base in un documento)

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

Title

(Titolo del documento. Visualizzato nella title bar del browser e nel tab della pagina. Molto importante per SEO perché è il titolo che appare nella pagina di ricerca Google.)

```
<title>HTML Reference</title>
```

Link

(definisce la relazione tra il documento e una risorsa esterna)

```
<head>
  <link rel="stylesheet" href="theme.css">
</head>
```

noscript (alternative a chi ha script disabilitati)

```
<script>
document.write("Hello World!")
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

Commenti in HTML

I **commenti** in HTML, come in altri linguaggi, permettono di aggiungere al markup una serie di annotazioni che non incidono sul rendering della pagina. I browser comunque leggono queste annotazioni e grazie ad esse possiamo agire anche attivamente sulla resa grafica del documento.

Per aggiungere un commento la sintassi è piuttosto semplice, si racchiude il testo tra <!-- e -->:

<!-- questo è un commento -->

[CTRL+K+C] scorciatoia da tastiera
[CTRL+K+U] scorciatoia da tastiera
(x decommentare)

```
<!doctype html>
<html lang="it">
<head><title>Ciao Mondo!</title></head>
<body>
    <h1>Ciao Mondo!</h1>
    <p>Questa è la nostra prima pagina HTML!</p>
    <!-- Questo è un commento
        può andare anche su più righe
        E può servire per non mostrare porzioni di codice
    <p>Non voglio vedere questo paragrafo</p>
    -->
</body>
</html>
```

Elementi di Blocco, In linea e Liste

I tag HTML possono rappresentare oggetti (come ad esempio le immagini) o **servire a suddividere la pagina in aree** (come i ‘div’ o le ‘section’). Ci sono diverse tipologie di tag e conoscerle diventa determinante per usare il tag giusto al posto giusto e per applicare in seguito le regole CSS.

Block (elementi blocco)	Elementi che costituiscono un blocco attorno a sé . Non fanno parte di un testo e di conseguenza mandano a capo, come i paragrafi, i div o le section . In genere sono elementi contenitore nell’albero che abbiamo tracciato in alto.
Inline (elementi “in linea”)	Elementi che non vanno a capo e possono essere integrati nel testo, come i link, le immagini, oppure gli span .
Liste	Liste numerate o non numerate. Sono simili agli elementi blocco con una struttura particolare, ideali per definire elenchi o menu

Gli attributi

I tag HTML possono essere corredati di uno o più attributi, che servono per meglio specificare la funzione o la tipologia dell'elemento, per memorizzare dati o per arricchire di significato il contenuto.

Un tag con attributi si scrive in questo modo:

```
<tag attributo1="valore1"  
attributo2="valore2">
```

Esempio:

```

```

In sostanza gli attributi:

- sono **coppie chiave-valore** separate dal carattere = (uguale);
- i valori sono tipicamente racchiusi tra virgolette "", ma è possibile anche utilizzare gli apici ' ';
- si scrivono lasciando almeno uno spazio dopo il nome dell'elemento nel tag di apertura (o nell'unico tag nel caso di elementi non contenitori).

global attributes HTML

title, lang, id, class, style, draggable

Lo standard HTML prevede un insieme di attributi che **possono essere applicati a tutti i tag**.

Possiamo suddividere questi attributi in due principali sottogruppi:

- **attributi di base (core);**
- **attributi per la gestione eventi.**

title, lang, id, class, style, draggable, contenteditable ecc..

https://developer.mozilla.org/it/docs/Web/HTML/Global_attributes

Eccone altri che consentono la selezione e l'applicazione di regole di stile per la rappresentazione grafica:

Attributi di base (core)

title	Informazione che aiuta a descrivere il contenuto dell'elemento. Il testo viene spesso rappresentato in un tooltip che viene mostrato al passaggio del mouse.
lang	Indica la lingua del contenuto. Es. <p lang="it">ciao</p>

id	Serve ad associare un identificatore univoco ad un elemento. Una volta assegnato un identificatore, non si dovrà assegnarlo a nessun altro elemento della pagina.
class	Serve a stabilire che l'elemento appartiene ad una serie di "classi". Torneremo sull'argomento, per ora ci basti sapere che possiamo applicare a tutti gli elementi di una classe proprietà simili. Possiamo inserire quante classi vogliamo, tutte separate da uno spazio. Es. <p class="saluto testo-chiaro">ciao</p>, ha due classi: "saluto" e "testo-chiaro".
style	Serve ad assegnare delle proprietà grafiche (Stili CSS) all'elemento, questo tipo di assegnamento degli stili viene detto stilizzazione "inline". Anche di questo parleremo in seguito
draggable	Può assumere i valori true o false e serve a stabilire se un elemento sia trascinabile per una operazione di drag-n-drop.

Attributi per la gestione degli eventi (normalmente utilizzati con eventi javascript)

Attributo	Descrizione
onclick	Rileva il click (o il tap) effettuato sull'elemento
onload	Associa un comportamento alla conclusione del caricamento dell'elemento
onscroll	Attiva un comportamento correlato allo scrolling della pagina
ondrag	Si attiva quando iniziamo a trascinare un elemento che abbiamo indicato come draggable="true"

Attributi

Rispetto alle definizioni degli attributi HTML5 consente una libertà maggiore rispetto a XHTML, segnando di fatto un ritorno alla filosofia di HTML 4. In sintesi: non è più obbligatorio racchiudere i valori degli attributi tra virgolette.

- **Attributi ‘vuoti’:** non è necessario definire un valore per l’attributo, basta il nome, il valore si ricava implicitamente dalla stringa vuota. Un caso da manuale:

Secondo le regole XHTML:

```
<input checked="checked" />
```

In HTML5:

```
<input checked>
```

- Attributi senza virgolette: **è perfettamente lecito in HTML5 definire un attributo senza racchiudere il valore tra virgolette.** Esempio:

```
<div class= testata>
```

- **Attributi con apostrofo (singolo apice):** il valore di un attributo può essere racchiuso tra due apostrofi

```
<div class='testata'>
```

- **Attributi con virgolette (doppio apice):** per concludere, è possibile usare la sintassi che prevede l’uso delle virgolette per racchiudere il valore di un attributo. Il codice:

```
<div class="testata">
```

HTML Colors

Tutti i browser moderni supportano i seguenti 140 nomi di colore

COLORNAME

AliceBlue - [#F0F8FF](#)

AntiqueWhite - [#FAEBD7](#)

Aqua - [#00FFFF](#)

...

WhiteSmoke - [#F5F5F5](#)

Yellow - [#FFFF00](#)

YellowGreen - [#9ACD32](#)

https://www.w3schools.com/colors/colors_names.asp

RGB

#000000 – Black, #000080 – Navy

#006400 DarkGreen

RRGGBB = red, green, blue

Rgb=rgb(255,255,255) = bianco

https://www.w3schools.com/colors/colors_rgb.asp

Colors 2021 (w3schools.com)

Gli attributi data- (data-attributes)

Lo standard HTML5 ha introdotto i cosiddetti “data-attributes”, ovvero degli attributi inseriti nei tag allo scopo di ritenere **alcune informazioni utili alla gestione del layout** o a supporto dell’interfaccia utente.

si legge in javascript da proprietà .dataset

```
<ul id="elenco">
<li data-indirizzo="Piazza Repubblica" data-
telefono="3863535555">Luca</li>
<li data-indirizzo="Via verdi" data-
telefono="3164192342">Paolo</li>
<li data-indirizzo="Largo la loggia" data-
telefono="3213344314">Umberto</li>
</ul>
```

I paragrafi <p>

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

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

Elemento di sezione <div>

Definisce una sezione in un HTML

```
<div>
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
```

Andare a capo,

Il tag **
** sta per *break line* e serve per andare a capo nel bel mezzo di un testo. Come è facile notare infatti, anche se andiamo a capo all'interno dell'editor, il testo nel browser non va a capo se non alla fine dello spazio disponibile per il suo contenitore

Per retro-compatibilità con xhtml si può usare anche

Questo è un testo che va a capo
spesso all'interno dell'editor
tuttavia i browser ignoreranno
questo fatto e considereranno <**br**>
il testo come una singola
riga continua, che va a capo
solo quando incontra il limite
del proprio contenitore

Headings, i titoli h1, h2, h3, etc.

I titoli (**headings**) sono i tag che ci aiutano a definire il tema della pagina. In genere sono rappresentati in grassetto e con una dimensione del testo ingrandita. Anche questi sono elementi di tipo **block** e sia le dimensioni sia il margine applicato per default dal browser sono proporzionali all'importanza del titolo.

<**h1h1**>

<**h2h2**>

<**h3h3**>

<**h3h3**>

<**h2h2**>

<**h3h3**>

<**h3**>Lampadari, tipologie e differenze</**h3**>

Grassetti e corsivi

Fin dalle prime versioni di HTML, perché un testo apparisse in grassetto, è stato sufficiente inserirlo in un tag ****, così come per mettere in corsivo una frase si è utilizzato il tag **<i>**.

Tag	Descrizione	Resa di base
	Attribuisce al testo una forte importanza, serietà o urgenza (ora <i>strong</i> sta per <i>importance</i>). Aiuta a tematizzare la pagina e può essere utilizzato anche per strategie SEO, può essere utilizzato anche all'interno di headings per indicare la parte più importante di un titolo (es. <h1>Capitolo 1. La casa</h1>). HA VALORE SEMANTICO	Grassetto
	Offre una differenza stilistica rispetto al resto del contenuto, senza attribuire un'importanza specifica al testo (nota: che sia bold o no non importa). NON HA VALORE SEMANTICO	Grassetto
	Simile a strong, serve a rappresentare un testo o una frase che si pronuncia in modo differente dal resto al testo. (da <i>emphasis</i> diventa <i>stress emphasis</i>) . HA VALORE SEMANTICO	Corsivo
<i>	Serve a rappresentare testo che esprima un tono, uno stato d'animo o qualcosa che si discosti dal resto del contenuto, senza aggiungere ulteriori significati o importanza.	Corsivo

Acronimi e abbreviazioni

In passato per le sigle o gli acronimi era possibile utilizzare il tag *acronym* che è stato poi deprecato e soppiantato dal tag **<abbr>** prima previsto solo per le abbreviazioni.

abbr

HTML

HyperText Markup Language

Questo tag viene utilizzato sia per abbreviazioni come “*Sig.*”, “*Ven.*”, sia per le sigle come *CSS* o *FIFA*. In ogni caso lo utilizziamo quando scriviamo la versione breve di una frase o una parola.

Passando con il mouse sopra il testo abbreviato, apparirà la sua versione estesa in un tooltip.

```
<p>In questa guida approfondiamo l'<abbr title="HyperText Markup Language"e ne esploriamo i tag e le potenzialità espressive.</p>
```

Apici e pedici

Inserire un apice o un pedice è qualcosa che ha a che fare con la tipografia o con l'aspetto? Non siamo nemmeno troppo a metà strada, pensiamo all'utilizzo dei pedici ad esempio, per inserire la numerazione nelle variabili in matematica, è qualcosa che è strettamente correlato con il contenuto.

sup e sub

I tag che utilizziamo sono **<sup>** per gli apici e **<sub>** per i pedici. Sono entrambi **elementi inline** e la rappresentazione di default sui browser è quella che ci aspettiamo, carattere rimpicciolito e posto più in alto o più in basso a seconda che si tratti di un apice o di un pedice.

E = mc²

<p> Questo è un _{pedice}</sub></p>

E = mc² ← SUP

Questo è un pedice

Codice e testo pre-formattato

Tag	Descrizione
<code>	Describe brani di codice sorgente
<kbd>	Indica il testo inserito dagli utenti. Il contenuto è visualizzato con il font monospace di default del browser.
<samp>	Rappresenta l'output di un programma o di una riga di comando. Contenuto visualizzato con il font monospace di default del browser.

```
<code>codice</code>
<kbd>keyboard digit</kbd>
<samp>sample</samp>
```

code kbd sample

Altri elementi per la formattazione del testo

Tag	Descrizione
	Describe un contenuto che vogliamo togliere dal documento , tipicamente traccia una barra sul testo e sostituisce <strike> , già deprecato . Una alternativa a del è <s> che indica un testo di nessuna importanza all'interno del documento.
<ins>	Può essere utile per definire degli aggiornamenti al documento : grazie all'attributo datetime possiamo infatti assegnare una data specifica a una porzione del testo.
<mark>	Rende il testo “evidenziato”, rappresentato non a caso con un colore di sfondo giallino “evidenziatore”
<time>	Grazie all'attributo datetime , consente di specificare in modo formale la data e l'ora di un elemento che riporta una data in un formato qualsiasi (es. <code><time datetime="2016-05-31">il trentun maggio</time></code>). Non fa nulla di particolare..
<small>	Indica un testo che ha una importanza minore rispetto al resto del documento e viene spesso rappresentato con un carattere più piccolo.

del ins mark time small

tag <pre>

Il tag <pre> definisce il testo preformatto. Il testo in un elemento <pre> viene visualizzato in un font a larghezza fissa (in genere Courier) e conserva spazi e interruzioni di riga.

```
<pre>
    Nel mezzo del cammin di nostra vita
    mi ritrovai per una selva oscura,
    ché la diritta via era smarrita.
    Ahi quanto a dir qual era è cosa dura
    esta selva selvaggia e aspra e forte
    che nel pensier rinova la paura!
    Tant' è amara che poco è più morte;
    ma per trattar del ben ch'i' vi trovai,
    dirò de l'altre cose ch'i' v'ho scorte.
</pre>
```

Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura,
ché la diritta via era smarrita.
Ahi quanto a dir qual era è cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!
Tant' è amara che poco è più morte;
ma per trattar del ben ch'i' vi trovai,
dirò de l'altre cose ch'i' v'ho scorte.

tag address

Il tag <address> definisce le informazioni di contatto dell'autore / proprietario di un documento o di un articolo.

Se l'elemento <address> si trova all'interno dell'elemento <body>, rappresenta le **informazioni di contatto per il documento**.

Se l'elemento <address> si trova all'interno di un elemento <article>, rappresenta le **informazioni di contatto per quell'articolo**.

```
<address>
Azienda
via della repubblica 450,
00100 Milano (MI)
</address>
```

Azienda via della repubblica 450, 00100 Milano (MI)

è un tag semantico e i crawler lo interpretano come tale

htmlentities

simboli che hanno un significato speciale per le pagine web
rappresentare ad esempio

<>, spazio, &

potrebbe essere un problema in quanto il browser interpreta < > come tag

<http://www.asciiitable.com/>

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	®	®

Elenchi puntati e numerati

Gli elenchi giocano un ruolo molto importante nei testi, ma anche nella definizione di elementi strutturali delle pagine HTML.

e tre tipologie di elenco disponibili in HTML:

Elenchi ordinati

Elenchi non ordinati

Elenchi di definizioni

UL, gli elenchi non ordinati (o elenchi puntati)

L'elenco non ordinato (**unordered list**) è forse il più usato e si descrive utilizzando il tag ****.

Al suo interno possiamo inserire gli elementi della lista (**list item**) utilizzando il tag ****. E

Ecco un semplice esempio

Type: deprecato in html5, sostituito da
style="list-style-type:disc"

```
<ul>
  <li>primo elemento</li>
  <li>secondo elemento</li>
  <li>terzo elemento</li>
</ul>
```

Type (deprecato)	Description
disc	Default. A filled circle
circle	An unfilled circle
square	A filled square

Annidare le liste non numerate

```
<ul>
  <li>primo della 1a lista</li>
  <li>secondo della 1a lista
    <ul>
      <li>primo della 2a lista</li>
      <li>secondo della 2a lista
        <ul>
          <li>primo della 3a lista</li>
        </ul>
      </li>
      <li>terzo della 2a lista</li>
    </ul>
  </li>
</ul>
```

[W3Schools CSS list-style-type demonstration](https://www.w3schools.com/css/css_list-style_type.asp)

OL, gli elenchi ordinati (o elenchi numerati)

Gli elenchi ordinati (ordered list) sono contraddistinti dall'enumerazione della lista. Il tag da utilizzare per aprire un elenco ordinato è **** e anche in questo caso gli elementi sono individuati dal tag ****:

```
<ol>
    <li>primo</li>
    <li>secondo</li>
    <li>terzo</li>
</ol>
```

[W3Schools CSS list-style-type demonstration](#)

Attributo type, per descrivere il tipo di elenco

Type: deprecato in html5, sostituito da **style="list-style-type:lower-alpha"** in CSS

Valore di type	Descrizione	List-style-type value
type="1"	numeri interi "arabi" (valore di default)	
type="a"	numeri alfabeto minuscolo	Lower-alpha
type="A"	numeri alfabeto maiuscolo	Upper-alpha
type="i"	numeri numeri romani minuscoli	Lower-roman
type="I"	numeri numeri romani maiuscoli	Upper-roman

OL: Start e value, riprendere o gestire la numerazione **DEPRECATI**

Può capitare che per esigenze diverse vogliamo far partire la numerazione da un certo valore (default start = 1), in questo caso ci serviamo di due attributi:

Type: deprecato in html5, sostituito da style="list-style-type:lower-alpha"

Attributo	Si applica al tag	Descrizione
Start DEPRECATO	<i>ol</i>	indica il valore iniziale da cui far partire la numerazione della lista. Alternativa CSS: counter-reset
Value DEPRECATO	<i>li</i> (solo se è all'interno di un <i>ol</i>)	indica il valore applicato alla numerazione di un certo elemento e modifica la numerazione anche degli elementi successivi nella lista.

```
<ol start="C" style="list-style-type:decimal-leading-zero" >
  <li>primo elemento</li>
  <li>secondo elemento</li>
  <li value="8">terzo elemento</li>
</ol>
```

```
ul.a {list-style-type: circle;}
ul.b {list-style-type: square;}
ol.c {list-style-type: upper-roman;}
ol.d {list-style-type: lower-alpha;}
```

Esempio ol li con counter-reset

- 1.Buy milk_
 - 2.Feed the dog
 - 3.Drink coffee
-
- c.Buy milk
 - d.Feed the dog
 - e.Drink coffee

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        ol#primo {
            list-style: none;
            counter-reset: li 0;
        }
        ol#primo li:before {
            content: counter(li) ".";
            counter-increment: li;
        }
        ol#secondo {
            list-style: none;
            counter-reset: li 2;
        }
        ol#secondo li:before {
            content: counter(li, lower-alpha) ".";
            counter-increment: li;
        }
    </style>
</head>
<body>
    <ol id="primo">
        <li>Buy milk_</li>
        <li>Feed the dog</li>
        <li>Drink coffee</li>
    </ol>
    <ol id="secondo">
        <li>Buy milk</li>
        <li>Feed the dog</li>
        <li>Drink coffee</li>
    </ol>
</body>
</html>
```

Tabella: struttura di base

Le tabelle sono componenti importanti in HTML: nate agli inizi del Web per impaginare dati aggregati, si sono poi trasformate in uno strumento indispensabile per gestire i layout grafici, per tornare, già nell'epoca dei CSS, ad essere elementi utilissimi per rappresentare informazioni.

Come sempre in questa guida ci rifacciamo allo standard attuale, per cui l'**elemento <table>** serve alla rappresentazione di dati, anche “in più dimensioni”, sotto forma di tabelle.

In HTML una tabella è formata da celle all'interno di righe e colonne.

```
<table>
<tr>
  <td>Colonna 1</td>
  <td>Colonna 2</td>
</tr>
<tr>
  <td>Dato 1,1</td>
  <td>Dato 1,2</td>
</tr>
<tr>
  <td>Dato 2,1</td>
  <td>Dato 2,2</td>
</tr>
<tr>
  <td>Dato 3,1</td>
  <td>Dato 3,2</td>
</tr>
</table>
```

Tabella: struttura di base

Tag	Descrizione
<table>	È il contenitore di tutta la tabella e la definisce
<tr>	<i>“table row”</i> Contiene una riga della tabella
<td>	<i>“table data”</i> Una cella che contiene i valori all'interno di una riga

caption, thead, tbody, tfoot

Per scrivere una tabella che fornisca una rappresentazione più chiara dei dati introduciamo un template leggermente più ricco

The diagram illustrates the structure of an HTML table and its corresponding data representation. The table structure is shown on the left, and the resulting data grid is on the right. Arrows point from specific HTML tags to their corresponding components in the grid.

titolo colonna 1	titolo colonna 2
1.1	1.2
2.1	2.2
Tot 1	Tot 2

```
<table>
  <caption>titolo della tabella</caption>
  <thead>
    <th>titolo colonna 1</th>
    <th>titolo colonna 2</th>
  </thead>
  <tbody>
    <tr>
      <td>1.1</td>
      <td>1.2</td>
    </tr>
    <tr>
      <td>2.1</td>
      <td>2.2</td>
    </tr>
    <tfoot>
      <tr>
        <td>Tot 1</td>
        <td>Tot 2</td>
      </tr>
    </tfoot>
  </tbody>
</table>
```

Annotations in the grid:

- A red arrow points from the `<caption>` tag to the header cell "titolo della tabella".
- A green arrow points from the `<thead>` tag to the first two columns of the data grid.
- An orange arrow points from the `<tfoot>` tag to the last two columns of the data grid.

`caption`, `thead`, `tbody`, `tfoot`

Tag	Descrizione
<code><caption></code>	È una didascalia che ci permette di dare una contestualizzazione ai dati e rendere più chiaro il loro significato.
<code><thead></code>	Serve per raggruppare le righe che rappresentano l'intestazione della tabella.
<code><th></code>	<i>“table header”</i> Indica una cella che contiene un'intestazione (ad esempio il titolo di una colonna o di una riga) e serve a dare una definizione dei dati cui si riferisce.
<code><tbody></code>	Raggruppa le righe che contengono il corpo della tabella, spesso con i dati veri e propri.
<code><tfoot></code>	Racchiude le righe che utilizziamo come footer della tabella, in cui si possono inserire dei dati di riepilogo, somme, medie, etc.

colgroup e col, raggruppare gli stili delle colonne

Colgroup serve a raggruppare le colonne di una tabella. Il tag colgroup deve essere inserito tra il caption (se presente) e qualsiasi altro raggruppamento per riga (thead, tbody, tfoot).

Tramite il tag **col**, si definiscono le **colonne che si vogliono includere nel gruppo** grazie all'attributo span nel quale viene indicato il numero di colonne facente parte di quel gruppo.

Ad ogni col appartiene un gruppo di colonne.

Le colonne vengono considerate da sinistra verso destra.

titolo della tabella	
titolo colonna 1	titolo colonna 2
1.1	1.2
2.1	2.2
Tot 1	Tot 2

```
<colgroup>
<col span="1" style="background-
color: #FF0000" />
</colgroup>
```

colgroup e col, raggruppare gli stili delle colonne

```
<h2>Colgroup</h2>
<p>
  Add the a colgroup with a col element that spans over two columns to
  define a style for the two columns:
</p>

<table style="width: 100%">
  <colgroup>
    <col span="4" style="background-color: #d6eeee" />
    <col span="2" style="background-color: #d6ffff" />
    <col span="1" style="background-color: #d68888" />
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
    <th>FRI</th>
    <th>SAT</th>
    <th>SUN</th>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
    <td>5</td>
    <td>6</td>
    <td>7</td>
  </tr>
  <tr>
    <td>8</td>
    <td>9</td>
    <td>10</td>
    <td>11</td>
    <td>12</td>
    <td>13</td>
    <td>14</td>
  </tr>
  <tr>
    <td>15</td>
    <td>16</td>
    <td>17</td>
    <td>18</td>
    <td>19</td>
    <td>20</td>
    <td>21</td>
  </tr>
  <tr>
    <td>22</td>
    <td>23</td>
    <td>24</td>
    <td>25</td>
    <td>26</td>
    <td>27</td>
    <td>28</td>
  </tr>
</table>
```

Colgroup

Add the a colgroup with a col element that spans over two columns to define a style for the two columns:

MON	TUE	WED	THU	FRI	SAT	SUN
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Col span=4

Col span=2

Col span=1

Annidare tabelle

Chiudiamo mostrando come è possibile annidare le tabelle le une dentro le altre

```
<table>
  <thead>
    <tr>
      <th>Campo</th>
      <th>Tabella</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Campo1</td>
      <td>
        <table>
          <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>
```

Campo	Tabella	
Campo1	Campo	Descrizione
	Campo1	Descrizione1

colspan (attr del td e th)

Per ottenere una **cella che occupi più colonne** utilizziamo l'attributo **colspan** che si applica a **td** e a **th** come nel nostro caso.

```
<table>
  <thead>
    <tr><th>Lunedì</th><th>Mercoledì</th><th>Venerdì</th></tr>
  </thead>
  <tbody>
    <tr><td colspan="3">Cyclette</td></tr>
    <tr><td>Esercizi spalle</td><td>Pettorali</td><td>Tricipiti</td></tr>
    <tr><td>Squat</td><td>Estensioni gambe</td><td>Polpacci</td></tr>
  </tbody>
</table>
```

Lunedì	Mercoledì	Venerdì
Cyclette		
Esercizi spalle	Pettorali	Tricipiti
Squat	Estensioni gambe	Polpacci

rowspan

Per ottenere una cella che si estenda su più di una riga utilizziamo l'attributo **rowspan**

```
<table>
  <thead>
    <tr><th>head</th><th>head</th><th>head</th></tr>
  </thead>
  <tbody>
    <tr><td rowspan="2">double row cell</td><td>cell</td><td>cell</td></tr>
    <tr><td>cell</td><td>cell</td></tr>
    <tr><td>cell</td><td>cell</td><td>cell</td></tr>
  </tbody>
</table>
```

head	head	head
double row cell	cell	cell
	cell	cell
cell	cell	cell

I link in HTML, l'ipertesto

I link sono “il ponte” che consente di passare da un testo all’altro. In quanto tali, i link sono formati da due componenti:

- il **contenuto**
- la **risorsa**

< a href="https://www.html.it/">HTML.it.

Il contenuto	La risorsa
È la parte visibile del link, e proprio per questo l’utente deve essere sempre in grado di capire quali sono i collegamenti da cliccare all’interno della pagina	Si tratta di un’altra pagina (sullo stesso server o su un server diverso), oppure è un collegamento interno a un punto della pagina stessa

I percorsi assoluti e relativi

I progetti HTML e i siti web sono dunque organizzati in strutture ordinate: non a caso si parla di **albero di un sito**.

- **Percorso assoluto** significa indicare per esteso l'indirizzo di un certo documento, elencando tutte le directory e sottodirectory che dobbiamo attraversare per raggiungerlo

```
<a href="https://www.html.it/css/index.html">fogli di stile</a>
```

- I **percorsi relativi** fanno riferimento alla posizione degli altri file rispetto al documento in cui ci si trova in quel momento

```
<a href="cartella/paginaDaLinkare.html">A  
NCORA</a>
```

è consigliabile non lasciare spazi vuoti nei nomi dei file, **MEGLIO USARE IL –**

/ = fare riferimento alla root

./image/ o image/ = stessa cartella

/image = dalla root

../image = dalla cartella precedente

I link interni o ancore

Possiamo sfruttare il meccanismo dei link anche per creare un indice interno al documento, utilizzando le àncore. Ciascuna àncora può avere infatti un nome univoco.

```
<div id="top"></div>  
...  
...  
...  
...  
<a href="#top">Vai su</a>
```

```
<a href="#primo">vai al primo  
paragrafo</a>  
<a name="primo">Stiamo per  
esaminare la struttura...  
Eccetera...</a>
```

```
<div id="top">Stiamo per  
esaminare la struttura...  
Eccetera...</a>
```

Gli attributi dei link

• title

L'attributo **title** è molto importante, e serve per specificare un testo esplicativo per l'elemento a cui l'attributo è riferito

• hreflang

Con “**hreflang**” si indica la lingua del documento: si tratta di un attributo che migliora l'accessibilità del sito

```
<a href="pagina.html" title="titolo"  
target="_blank|_self|_parent|_top|"  
hreflang="en" accesskey="h">link</a>
```

• target

È anche possibile specificare in quale finestra la pagina linkata deve essere aperta: di **default** infatti la pagina viene aperta all'interno del documento stesso, ma è possibile specificare che la pagina sia aperta in una nuova finestra:

Target

_blank (apre il collegamento in una nuova finestra o scheda)

_parent (apre il collegamento nell'eventuale collegamento principale)

_self (apre il collegamento nella stessa finestra. Default)

_top (apre il collegamento a tutta finestra)

framename (permette di aprire il collegamento in uno specifico riquadro – frame)

• accesskey

Le **accesskey** sono delle scorciatoie “da tastiera” da mettere nel tag a (accesskey=«h») PER ATTIVARE alt+h

```
<a href="#miaancora"  
accesskey="h">vai a ancora</a>
```

Inserire le immagini

img

Il tag **** rappresenta il principale elemento per inserire un'immagine in una pagina HTML


```

<https://dummyimage.com/>

<https://picsum.photos/200/300>

## ✓ img

È il nome del tag, abbreviazione di image (immagine)

## ✓ src

Sta per source (origine), è l'indirizzo (URL) del file che vogliamo mostrare

## ✓ alt

È il testo alternativo, ovvero il testo che appare se, per qualche motivo, il client non riesce a mostrare l'immagine.

Possiamo anche omettere questo attributo, ma risulta utile per l'accessibilità e per i motori di ricerca

# Includere un file Audio

---

Il tag <audio> è stato introdotto con HTML5

```
<audio controls>
 <source src="musica.mp3"
 type="audio/mp3">
 <source src="musica2.ogg"
 type="audio/ogg">
Il browser non supporta il tag audio
</audio>
```



Oppure

```
<audio src="mp3.mp3" controls>
</audio>
```

# Includere un file Video

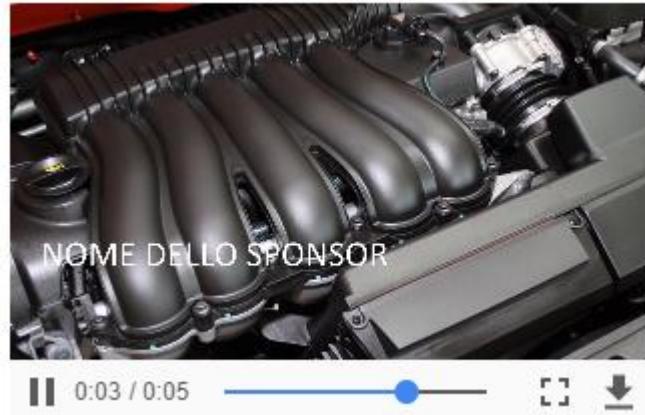
---

Il tag <video> è stato introdotto con HTML5.

```
<video width="400" controls
autoplay>
<source src="video.mp4">
</video>
```

Oppure

```
<video src="mp4.mp4" controls>
</video>
```



# Form

---

Utilizzati per raccogliere i dati dell'utente.

L'invio dei dati è solitamente organizzato in due parti:

- una **pagina principale** che contiene i vari campi dei form, che consentono all'utente di effettuare delle scelte, scrivere del testo, inserire un'immagine
- Una **pagina secondaria** che viene richiamata dalla principale e che effettua "il lavoro" vero e proprio di processare e raccogliere i dati. Di norma si tratta di una pagina di programmazione che si trova sul server.

## Name e action

```
<form name="datiUtenti"
action="paginaRisposta.php"
method="GET"
enctype="multipart/form-data">
...
</form>
```

## Method

due metodi di invio: **GETe POST.**

## Enctype

non è necessario utilizzarlo se non quando si passano file allegati

# Raggruppare i moduli

Tramite il tag **<fieldset>** possiamo creare delle macro-aree all'interno dei form, e grazie al tag **<legend>**, possiamo indicare il nome di ciascuna macro-area

Datianagrafici

---

--	--

```
<form action="#">
<fieldset>
 <legend>Datianagrafici</legend>
 <label></label><input>
 <label></label><input>
</fieldset>

<fieldset>
 <legend>Residenza</legend>
 <label></label><input>
 <label></label><input>
</fieldset>

.....
</form>
```

# label

---

Identifica l'etichetta di un campo

l'attributo **for** serve ad indicare l'id del suo control. Deve quindi essere uguale all'id dell'elemento al quale il label si riferisce.

```
<label for="male">Male</label>
 <input type="radio" name="gender" id="male" value="male">

```

# input

---

Il tag più utilizzato è l'**<input>**, che è senza chiusura.

Specifica un campo di input nel quale l'utente può inserire dati.

Si presenta in diversi modi a seconda del valore dell'attributo type.

[per tutti gli attributi:](#)

[https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp)

```
<input type="text"> DEFAULT
<input type="button">
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="time">
<input type="url">
<input type="week">
```

# I bottoni (sumbit, reset, button, image)

## <button>

---

Tag per creare submit o bottoni generici.

L'attributo type indica di che tipo di button si tratta: button, reset, submit.

Grazie all'attributo **“disabled”** è infine possibile disabilitare i bottoni.

Es:

```
<input type="submit"
value="invia" disabled>
```

```
<form
action=http://www.html.ittarget=
"_blank">
<input type="text">

<button type="button">
bottone generico
</button>
<button type="reset">
cancella
</button>
<button type="submit">
invia
</button>
</form>
```

# Inserire testo (campo testo, textarea, password)

---

Per consentire all'utente di inserire del testo è possibile **utilizzare un 'campo testo'**

```
<input name="mioTesto" type="text"
value="qui il tuo testo" size="40"
maxlength="200" />
```

## **INPUT TYPE PASSWORD**

```
<input type="password"
maxlength="8" size="18"
value="mia_password"
name="mioTesto" />
```

## **TEXTAREA**

Campo di input di testo multilinea.  
Le dimensioni sono specificate dagli attributi **rows** e **cols**.

```
<textarea name="testo" rows="5"
cols="40">
 qui puoi scrivere il tuo testo
</textarea>
```

[https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp)

# Checkbox

---

Con le **checkbox** possiamo consentire all'utente di operare delle scelte multiple. Ad esempio:

- html
- css
- JavaScript

## Checked

Valori spuntati di default

- checked="checked"

## disabled

rendere una casella non accessibile

- disabled="disabled"

```
<form action="">
 <fieldset>
 <legend>Linguaggi conosciuti</legend>

 <input type="checkbox" name="html" value="html"/> html

 <input type="checkbox" name="css" value="css"/> css

 <input type="checkbox" name="javascript" value="javascript"/> JavaScript
 </fieldset>
</form>
```

# Radio button

**radio button** (“bottoni circolari”) invece consentono di effettuare una scelta esclusiva.

Generalmente usato in gruppi di radio button che descrivono una serie di opzioni. Ai radio button appartenenti allo stesso gruppo deve essere dato lo **stesso name, ma value differenti**.

HTML  CSS  JavaScript 

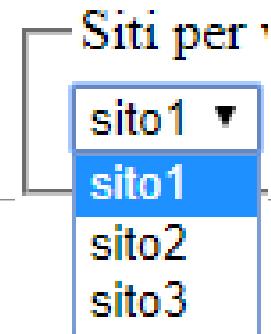
```
<input type="radio" name="r1" value="2" checked="checked">2
<input type="radio" name="r1" value="3">3
<input type="radio" name="r1">4
```

```
<form action="">
 <fieldset>
 <legend>Linguaggi conosciuti</legend>
 HTML
 <input type="radio" name="linguaggio" value="html"/>
 CSS
 <input type="radio" name="linguaggio" value="css"/>
 JavaScript
 <input type="radio" name="linguaggio" value="javascript"/>
 </fieldset>
</form>
```

# select

Grazie al tag **<select>** è possibile costruire dei menu di opzioni. In questo caso ciascuna voce deve essere compresa all'interno del tag **<option>**

Con l'attributo **“selected”** si può indicare una scelta predefinita:  
`selected="selected"`



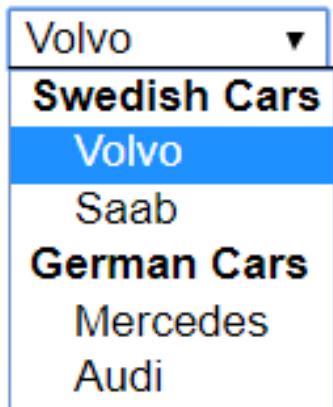
```
<form>
<fieldset>
<legend>Siti per
webmaster</legend>

<select name="siti">
<option
value="http://www.sito1.it"
selected="selected">sito1</option>
<option
value="http://www.sito2.it">sito
2</option>
<option
value="http://www.sito3.it">sito
3</option>
</select>
</fieldset>
</form>
```

# optgroup

---

Utilizzato per raggruppare le option di una select



```
<select>
 <optgroup label="Swedish Cars">
 <option value="volvo">Volvo</option>
 <option value="saab">Saab</option>
 </optgroup>
 <optgroup label="German Cars">
 <option
 value="mercedes">Mercedes</option>
 <option value="audi">Audi</option>
 </optgroup>
</select>
```

# Campi file e hidden

## CAMPI HIDDEN

Servono per passare dei parametri “di servizio”, senza far percepire la loro presenza all’utente

```
<input type="hidden"
name="urlDiProvenienza"
value="www.sito1.it">
```

## CAMPO FILE

Il campo “file”, consente invece di inviare uno o più file sul server

```
<input type="file"
name="mioallegato" >
```

## Alcuni attributi dell'input type file

- accept="audio/\*,video/\*»  
**accept**=specifica i formati di file accettati. Lista separata da virgolette.
- Multiple: se presente, accetta più file.

**Scegli file** d111\_2.jpg

# Lista Tag Semanticci

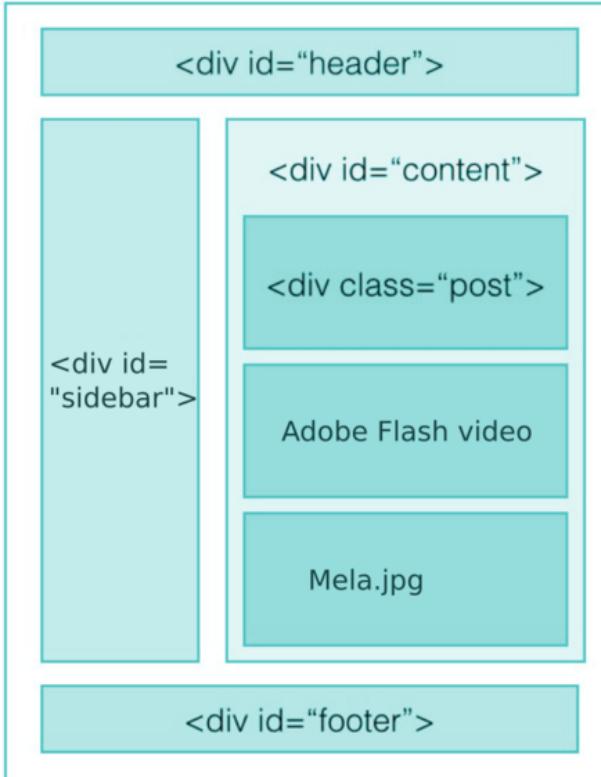
Etichetta	Descrizione
<article>	Definisce un articolo
<aside>	Definisce il contenuto a parte il contenuto della pagina
<details>	Definisce dettagli aggiuntivi che l'utente può visualizzare o nascondere
<figcaption>	Definisce una didascalia per un <figure> elemento
<figure>	Specifica il contenuto self-contained, come illustrazioni, diagrammi, foto, elenchi di codici, etc.
<footer>	Definisce un piè di pagina di un documento o di una sezione
<header>	Specifica un'intestazione per un documento o sezione
<main>	Specifica il contenuto principale di un documento
<mark>	Definisce segnato text / evidenziato
<nav>	Definisce link di navigazione
<section>	Definisce una sezione in un documento
<summary>	Definisce un'intestazione visibile per un <details> elemento
<time>	Definisce una data / ora

# Un nuovo content model per HTML5

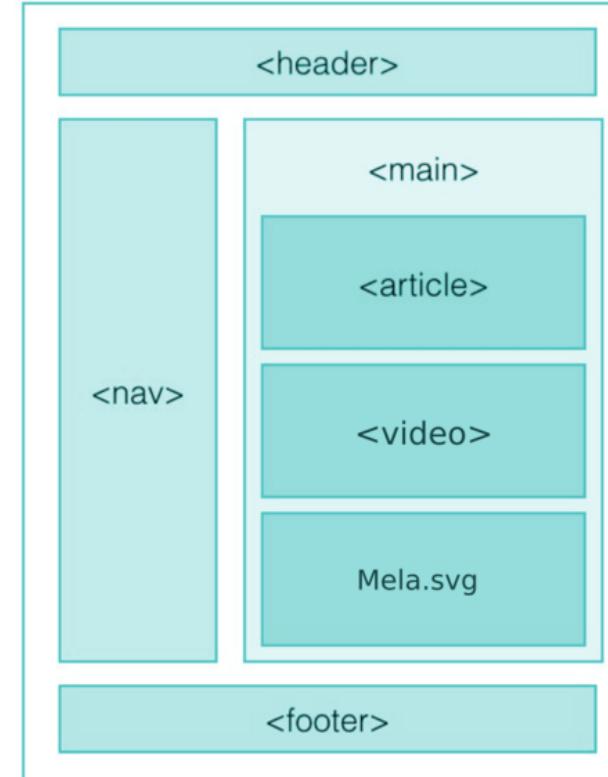
**NON PIU' SOLO DIV MA TAG**

**SEMANTICI**

**HTML4**



**HTML5**



# Un nuovo content model per HTML5

---

## Non più solo div

si possono utilizzare i tag  
semantici per offrire una struttura  
alla nostra pagina html

<header>  
<nav>  
<article>  
<footer>

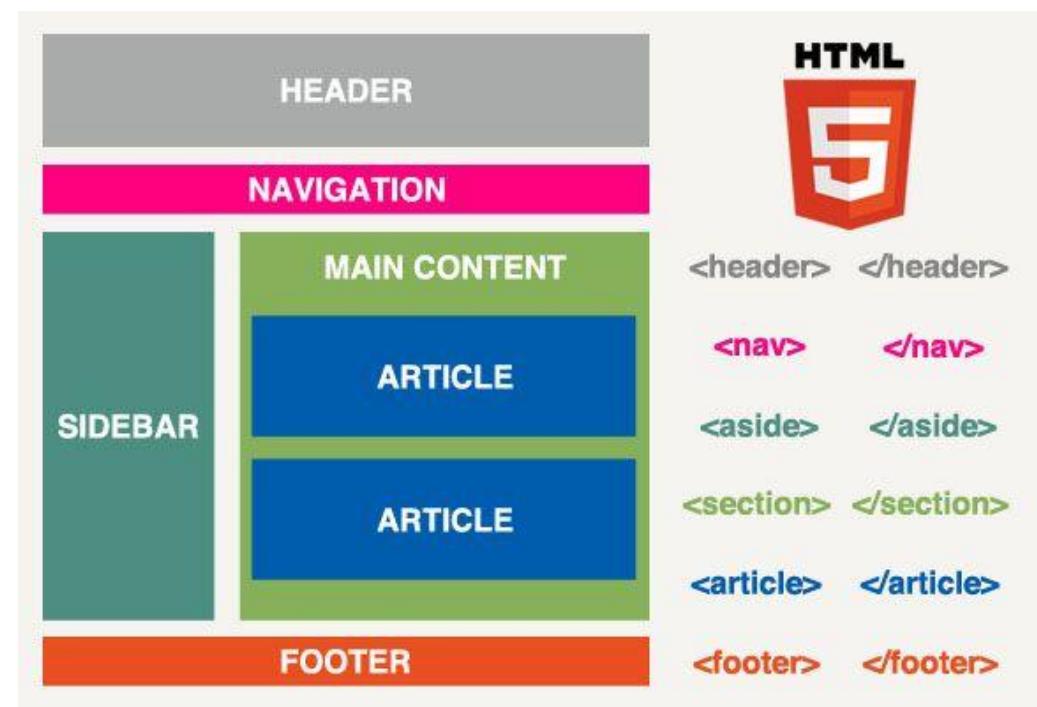
```
<body>
 <header>--- Titolo e Testata ---
 </header>
 <nav>--- Voci di Menu ---</nav>
 <main> ---Contenuto principale della
 pagina
 <article>--- Un Post ---</article>
 <article>--- Un altro Post ---</article>
 </main>
 <footer>---Footer a fine
 pagina
 </footer>
</body>
```

# Sectioning content

il gruppo contiene tutti quegli elementi studiati per ospitare contenuti atomici e semanticamente ordinati. Ovviamente non vanno per obbligatoriamente usati tutti questi elementi.

article,  
aside  
nav  
section  
Main  
Footer

Ogni elemento contiene uno specifico tipo di contenuto. Questo aiuta i motori di ricerca e gli screen reader a determinare la struttura del nostro sito.



# hgroup

---

serve per raggruppare "semanticamente" i titoli.

Rappresenta l'intestazione di una sezione quando questa è composta da titoli di più livelli (h1, sottotitoli, titoli alternativi o claim).

```
<header>
<hgroup>
<h1> A good news about my
course</h1>
<h2> How to build a web
site</h2>
<h3>Vediamo questo
articolo</h3>
</hgroup>
</header>
```

# Header

---

L'elemento `<header>` è un contenitore per altri elementi

L'elemento `<header>` non va confuso con quella che è la testata/intestazione principale di un documento

```
<header>
<h1>Questo è un titolo</h1>
<h2>Questo è un sotto-titolo</h2>
[...]
</header>
```

La natura e gli scopi dell'elemento `<header>` non dipendono dalla sua posizione nel documento, ma dai suoi contenuti (ausili alla navigazione o elementi introduttivi).

Il suo uso non è obbligatorio e in alcuni casi può risultare superfluo se non utilizzato in maniera appropriata.

# Footer

L'elemento `<footer>` deve contenere in genere **informazioni sulla sezione che lo contiene** come:

i dati di **chi ha scritto i contenuti**; collegamenti ai documenti correlati;

**i dati di copyright**;

Non necessariamente deve essere inserito solo alla fine di un documento.

Non introduce una nuova sezione e quindi non è rilevante per l'[outliner](#).

All'interno di una pagina web possono essere presenti diversi `<footer>` anche più di uno per lo stesso elemento.

---

```
<footer>
<dl>
<dt>Creato da</dt>
<dd><address>Creato r</address></dd>
<dt>Ultimo aggiornamento</dt>
<dd><time datetime="2017-11-01" pubdate>NN Novembre</time></dd>
<dd>
</dl>
<small>Testo nel piede - licenza</small>
</footer>
```

```
<footer>
<small>©2011 Autore contenuto. Design by Designer sito </small>
</footer>
```

# Section

---

Il tag **<section>**, secondo la definizione presente nella specifica HTML5, rappresenta **una sezione generica di un documento o applicazione.**

```
<article>
 <section>
 <h1>Titolo 1</h1>
 <p>Testo correlato al titolo 1.</p>
 </section>
 <section>
 <h1>Titolo 2</h1>
 <p>Testo correlato al titolo 2.</p>
 </section>
</article>
```

```
<article>
 [contenuto del post...]
 <section>
 <article>
 [commento1...]
 </article>
 <article>
 [commento2...]
 </article>
 <article>
 [commento3...]
 </article>
 <section>
 [Inserisci un nuovo commento...]
 </section>
 </section>
</article>
```

L'elemento **<section>** non deve essere utilizzato in sostituzione del **<div>** per impostare graficamente la pagina; inoltre è fortemente consigliato utilizzare i **<div>** anche quando risultano più convenienti per gli script.

L'elemento **<section>** non deve essere preferito all'elemento **<article>** quando i contenuti possono essere ripubblicati anche su altri siti.

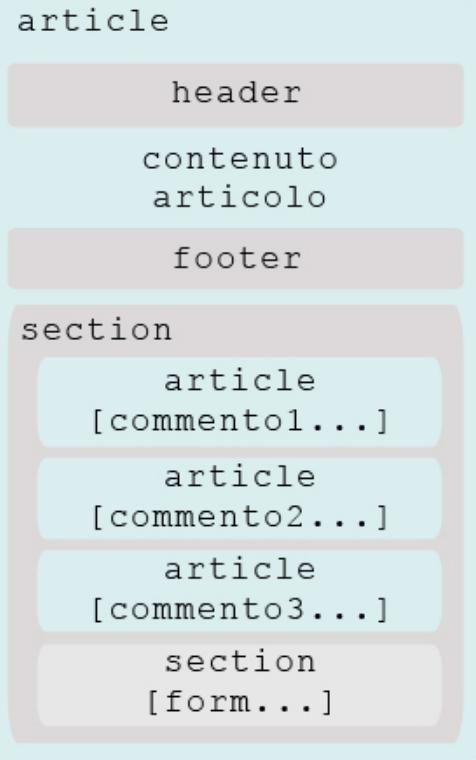
L'elemento **<section>** e l'elemento **<article>** non sono indipendenti ed esclusivi: possiamo avere sia un **<article>** all'interno di un **<section>** che viceversa

# Article

Il tag `<article>` rappresenta una sezione autonoma in un documento, pagina, applicazione o sito; infatti è potenzialmente ridistribuibile o riutilizzabile, e quindi ripubblicabile in parte o interamente in diverse pagine quando gli elementi `<article>` sono nidificati, gli `<article>` interni rappresentano gli articoli che sono in linea di principio relativi al contenuto dell'`<article>` esterno. Ad esempio, un blog che accetta commenti dagli utenti potrebbe rappresentarli come `<article>` figli annidati all'interno dell'elemento padre `<article>`.

Le informazioni relative all'autore dell'`<article>` non devono essere replicate all'interno degli elementi nidificati all'interno dello stesso; L'elemento `<time>` con l'attributo `pubdate` può essere utilizzato per definire la data di pubblicazione dell'`<article>`.

```
<section>
<h1>L'ultimo post</h1><article>
<header>
<time datetime="2010-11-22" pubdate>Lunedì 22 Novembre</time>
<h2>Nuove scoperte sul tag video!</h2>
</header>
<p>
Attraverso un utilizzo sapiente del tag canvas è possibile leggere uno stream di dati proveniente da un tag video e <mark>manipolarlo in tempo reale</mark>.
</p>
<footer>
<dt>autore: </dt>
<dd><address>Angelo Imbelli</address></dd>
<dt>categoria: </dt>
<dd>m...</dd>
<dt>tags: </dt>
<dd>video,</dd>
<dd>canvas,</dd>
<dt>permalink: </dt>
<dd>...</dd>
<dt>rank:</dt>
<dd><meter value="3.0" min="0.0" max="5">...</meter></dd>
</footer>
<section>
<h3>Commenti:</h3>
<article>
<h4>
<time datetime="2010-11-22" pubdate>L'...</time>
Angelo Imbelli ha scritto:
</h4>
<p>C'è un bell'esempio sulla rete: effetto a...
<footer>
<address><a href="mailto:ambelli@mbe...</address>
</footer>
</article>
<article>
<h4>
<time datetime="2010-11-23" pubdate>N...</time>
Sandro Paganotti ha scritto:
</h4>
<p>Bellissimo! Grazie per la segnalazione!</p>
<footer>
<address>Sandro Paganotti</address>
</footer>
</article>
<section>
<h4>Inserisci un nuovo commento:</h4>
<form>
[campi form per inserire un nuovo commento]
</form>
</section>
</section>
</article>
</section>
```



# Article

---

```
<section>
<h1>L'ultimo post</h1><article>
<header>
 <time datetime="2010-11-22" pubdate>Lunedì 22 Novembre</time>
 <h2>Nuove scoperte sul tag video!</h2>
</header>
<p>
 Attraverso un utilizzo sapiente del tag canvas è possibile leggere uno
 stream
 di dati proveniente da un tag video e <mark>manipolarlo in tempo
 reale</mark>.
</p>
<footer>
 <dl>
 <dt>autore: </dt>
 <dd><address>Sandro
Paganotti</address></dd>
 <dt>categoria: </dt>
 <dd>multimedia,</dd>
 <dt>tags: </dt>
 <dd>video,</dd>
 <dd>canvas,</dd>
 <dt>permalink: </dt>
 <dd><a href="2010/22/11/nuove-scoperte-sul-tag-
video">permalink,</dd>
 <dt>rank:</dt>
 <dd><meter value="3.0" min="0.0" max="5.0" optimum="5.0">ranked
3/5</meter></dd>
 </dl>
</footer>
```

```
<section>
<h3>Commenti</h3>
<article>
 <h4>
 <time datetime="2010-11-22" pubdate>Lunedì 22 Novembre</time>
 Angelo Imbelli ha scritto:
 </h4>
 <p>C'è un bell'esempio sulla rete: effetto ambi-light!</p>
 <footer>
 <address>Angelo Imbelli</address>
 </footer>
</article>
<article>
 <h4>
 <time datetime="2010-11-23" pubdate>Martedì 23 Novembre</time>
 Sandro Paganotti ha scritto:
 </h4>
 <p>Bellissimo! Grazie per la segnalazione!</p>
 <footer>
 <address>Sandro
Paganotti</address>
 </footer>
</article>
<section>
 <h4>Inserisci un nuovo commento:</h4>
 <form>
 [campi form per inserire un nuovo commento]
 </form>
</section>
<section>
</article>
</section>
```

# Nav

---

Il tag **<nav>** è uno degli elementi introdotti nelle specifiche HTML5 di più facile comprensione. Infatti, rappresenta una sezione di una pagina che **contiene link (collegamenti) ad altre pagine o a parti interne dello stesso documento**; quindi, in breve, una sezione contenente **link di navigazione**

```
<nav>

 Questo è un link
 Questo è un link
 Questo è un link
 Questo è un link
 [...]

</nav>
```

# Aside

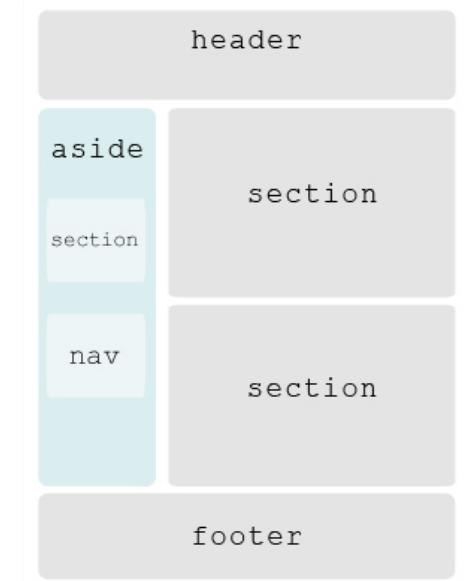
---

L'elemento `<aside>` rappresenta una sezione di una pagina costituita da informazioni che sono marginalmente correlate al contenuto dell'elemento padre che la contiene, e che potrebbero essere considerate distinte da quest'ultimo. Questo è ciò che viene indicato nelle specifiche HTML5, ma è facile immaginare l'utilità del tag `<aside>` semplicemente pensandolo come un **contenitore di approfondimento** in cui possiamo inserire gruppi di link, pubblicità, bookmark e così via.

```
<aside>
 <h1>Sidebar</h1>
 <section>
 <h2>Ricerca nel form:</h2>
 <form name="ricerca" method="post" action="/search">
 <label> Parola chiave:
 <input type="search" autocomplete="on" placeholder="article, section, ..." name="keyword" required maxlength="50">
 </label>
 <input type="submit" value="ricerca">
 </form>
 </section>
 <nav>
 <h2>Categorie</h2>

 multimedia
 marcatori testuali
 forms

 </nav>
</aside>
```



# Riferimenti bibliografici

---

I contenuti sono tratti dal sito [html.it](http://html.it), [w3schools.com](http://w3schools.com) e rielaborati dal docente