



# 1. Introduzione



# HTML

**HTML** (hypertext markup language) è un linguaggio di *markup* per la creazione di **siti web e documenti** che definisce come elementi testuali, link (ipertesti) e contenuti multimediali devono essere disposti e visualizzati all'interno del contesto di una pagina web.

Non è un linguaggio di programmazione poiché non possiede, ad esempio, strutture di controllo (if / else) e di iterazione (for / while). HTML **non permette** infatti di implementare una *logica applicativa*.



# HTML

## HTML **permette di:**

- ▶ Mostrare contenuti testuali
- ▶ Collegare delle pagine tra loro attraverso dei *link* (*hypertext*)
- ▶ Visualizzare contenuti multimediali quali immagini, video, audio etc.

## Html **non permette di:**

- ▶ Decidere cosa mostrare in base al comportamento dell'utente
- ▶ Mostrare  $n$  volte un elemento a partire da una singola dichiarazione
- ▶ Effettuare chiamate a servizi di qualsiasi genere, gestire autenticazioni, sessioni, interrogazioni a database etc...



# Lavorare con HTML

Per poter iniziare a lavorare con HTML ci sono due requisiti da soddisfare:

1. un **browser** per poter visualizzare e debuggare il codice
2. un **editor di testo** per poter scrivere il codice

Non esiste una scelta obbligata in questo senso poiché molto dipende dal gusto personale. Ad ogni modo possiamo trovare delle scelte preferibili rispetto ad altre per alcune ragioni...



# Scelta del Browser

Il browser è il software utilizzato per la *navigazione di pagina web*. Non permette soltanto di visualizzare il contenuto di una pagina HTML ma analizzarne la struttura attraverso una **console di sviluppo**.

È proprio la *console di sviluppo* a giocare un ruolo chiave nella scelta di un browser: è lo strumento che più verrà utilizzato nello sviluppo di siti / applicativi per il web. In virtù di questa osservazione deve essere quanto più *user friendly* o comunque di gradimento di chi sviluppa.

Un'ottima scelta in termini di browser ricade su quei software che utilizzano *webkit*, ovvero **Google Chrome** o **Safari**. Altri browser sono...

- ▶ Mozilla Firefox
- ▶ Microsoft Edge
- ▶ Internet Explorer (preferibilmente da evitare)



# Creare un file HTML

## Quando si crea un file HTML

- ▶ Per testarne il codice non è necessario avere un web server
- ▶ Il file deve terminare con l'estensione .html \*
- ▶ Il codice viene interpretato direttamente dal browser
- ▶ Per convenzione il file di *entry point* di un sito web è *index.html*

## Ad esempio:

[www.dominio.com](http://www.dominio.com) carica il file *index.html*

[www.dominio.com/contatti.html](http://www.dominio.com/contatti.html) carica il file *contatti.html*

\* Per molto tempo veniva usata l'estensione .htm per sopperire alla mancata gestione delle estensioni di file con 4 caratteri



# Esercizi

- ▶ Creare un file HTML denominato *index*
- ▶ Verificare che venga aperto con il *browser di default*
- ▶ Aprire il file ed inserire una stringa arbitraria; Aprire il file e verificare che il browser mostri la stringa contenuta nel file



## 2. Struttura



# Il concetto di Tag

Un *tag* è assimilabile ad un blocco che concorre a comporre una pagina web. Ciascun tag definisce come il contenuto deve essere formattato e visualizzato nel *browser*. Un tag è strutturato come segue:

```
<tag>contenuto</tag>
```

- ▶ Il nome dell'elemento viene racchiuso nelle *angle brackets* (minore e maggiore)
- ▶ Spesso (ma non sempre) i tag prevedono una apertura (`<tag>`) ed una chiusura (`</tag>`) come nel caso dell'esempio precedente
- ▶ I tag che non prevedono una chiusura non possono avere del contenuto. Prima di HTML5, i tag privi di chiusura seguivano la struttura `<tag />`.



# Struttura di una pagina

Una pagina HTML segue una struttura ben definita:

- ▶ Il codice viene racchiuso all'interno del tag `<html> ... </html>`
- ▶ Si divide in due sezioni distinte da uno specifico tag: `head` e `body`

L'**head** di una pagina web contiene informazioni che non vengono direttamente mostrate nella finestra del contenuto del browser: al suo interno vi sono informazioni utili ai motori di ricerca per indicizzare la pagina (*metadata* e *keywords*), il *title* della pagina web e collegamenti a risorse quali file CSS e JavaScript.

Il **body** racchiude il contenuto della pagina da mostrare all'interno della finestra del browser (testo, immagini, etc...)



# Struttura di una pagina

```
<!DOCTYPE html>  
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```



# Doctype

Il **doctype**, ovvero la prima riga nell'esempio precedente, non è un tag HTML bensì una dichiarazione che indica al browser il tipo di versione HTML che si sta utilizzando nella pagina. È utile:

- ▶ Ai fini della *validazione* del file attraverso gli standard definiti dai consorzio
- ▶ ogni versione di HTML ha una **specifica DTD** utilizzata dai browser per interpretarne correttamente la struttura

Alcuni esempi di doctype...

- ▶ HTML5 - <!DOCTYPE html>
- ▶ HTML4.01 Strict - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
- ▶ XHTML Strict - <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/xhtml1-strict.dtd">



# Commenti

È possibile inserire dei commenti all'interno del codice HTML. Questi verranno ignorati dal browser e *non renderizzati* nel contenuto della pagina. I commenti sono comunque visibili quando si analizza il sorgente della pagina attraverso gli strumenti offerti dal browser, come la console di sviluppo o la modalità view-source.

```
<!-- commento -->
```



# Inline vs Block

Gli elementi HTML vengono distinti in due macro-categorie di elementi: *inline* e *block*. La differenza tra queste due categorie risiede nel modo in cui questi vengono renderizzati dal browser.

Gli elementi **Inline**:

- ▶ Non vengono posti a capo (su una nuova riga)
- ▶ Sono dimensionati sull'asse orizzontale per il solo spazio di cui necessita il contenuto

Gli elementi **Block**:

- ▶ Cominciano a capo (su una nuova riga)
- ▶ Sono dimensionati sull'asse orizzontale per il massimo spazio disponibile



# Esercizi

- ▶ Modificare il file *index.html* in modo che sia strutturato secondo gli standard imposti da HTML
- ▶ Aggiungere dei commenti che definiscano l'inizio e la fine dei tag *head* e *body* (es. Head starts – Head ends)



# 3. Tag testuali



# Heading Tag

Gli *Heading Tag* sono elementi *block* delegati alla formattazione dei titoli. Esistono 6 differenti tipi di heading tag: ciascuno viene distinto dall'altro attraverso uno specifico numero che determina non solo la **grandezza** dell'elemento all'interno della pagina, ma anche **l'importanza ed il peso** che ha un titolo rispetto ad un altro.

```
<h1>Titolo</h1>
```

Il tag *h1* è sicuramente quello che all'interno della pagina ha maggiore rilevanza rispetto agli altri: è importante ricordare di usare non più di un tag *h1* per beneficiare dei vantaggi che questo tag ha nei confronti della **SEO** (search engine optimization).



# Heading Tag

Per utilizzare tag di minore rilevanza è sufficiente sostituire il numero 1 con 2, 3, 4, 5 o 6. Gli altri tag di intestazione possono essere usati quante volte si vuole, senza inficiare nella buona ottimizzazione della pagina per i motori di ricerca... Ad esempio:

```
<h2>Sottotitolo</h2>
```

Essendo differente il peso dato al testo da ciascun tag di intestazione, è importante definire attentamente il contesto in cui utilizzare una determinata numerazione. Il tag h2 è sicuramente un ottimo candidato per contenere un sottotitolo, ovvero un elemento di testo che segue quanto descritto in h1.



# Paragrafi

Il tag `p` è un elemento *block* che delimita un blocco di testo sottoforma di paragrafo: il testo viene posizionato senza elementi ai lati (su una nuova riga) e con un margine superiore ed inferiore stabilito di default dal browser.

```
<p>Lorem ipsum dolor sit amet</p>
```

È possibile inserire, all'interno dei paragrafi, degli specifici tag di tipo inline con lo scopo di fornire una formattazione delimitata a delle determinate porzioni di testo.



# Grassetto

Il tag delegato alla formattazione in grassetto(*bold*) del testo è **strong**. Questo tag rappresenta un *elemento inline*, pertanto verrà inserito senza ritorni a capo o margini rispetto agli elementi adiacenti.

- ▶ Viene utilizzato all'interno di tag testuali per formattare il testo in esso contenuto in corsetto (*bold*)
- ▶ Sostituisce il vecchio tag **<b>**

Ad esempio...

```
<p>Testo normale e <strong>testo in grassetto</strong></p>
```



# Corsivo

Il tag delegato alla formattazione in corsivo (*italic*) del testo è **em** (emphasize). Questo tag definisce un elemento inline.

- ▶ Viene utilizzato all'interno di tag testuali per formattare il testo in esso contenuto in corsivo (bold)
- ▶ Sostituisce il vecchio tag *<i>*

Ad esempio...

```
<p>Testo normale e <em>testo in corsivo</em></p>
```



# Ritorno a capo

Il ritorno a capo è gestito attraverso il tag `<br>` (*break*). È particolarmente utile per...

- ▶ La scrittura di poesie
- ▶ La separazione delle righe in un indirizzo
- ▶ Etc...

Il break **non deve essere usato** per la **separazione dei paragrafi!** I singoli paragrafi infatti devono essere gestiti attraverso il tag `<p>`.

Ad esempio...

```
<p>Largo di Torre Argentina 1<br>00100, Roma</p>
```



# Link

I link sono quell'elemento che determina la proprietà *ipertestuale* di HTML. Permettono di creare dei *collegamenti testuali* a risorse di vario tipo, ad esempio:

- ▶ Pagine web, siano esse esterne o interne al dominio di origine
- ▶ Risorse di vario genere come *file da scaricare, immagini, video etc...*
- ▶ Azioni come *l'invio di email*
- ▶ Elementi all'interno della pagina, referenziati tramite id

I link sono elementi *inline* introdotti attraverso il tag `a` (anchor)...

```
<a href="indirizzo alla risorsa">testo cliccabile</a>
```



# Link

L'indirizzo alla risorsa può essere:

- ▶ Assoluto – comprende il dominio della risorsa alla quale si vuole accedere, solitamente una *risorsa esterna*. Ad esempio...
  - ▶ <http://www.repubblica.it>
  - ▶ <https://www.restatic.it/cless/main/nazionale/2016-v1/img/home/repubblica-logo.png>
- ▶ Relativo – non include il dominio della risorsa alla quale si vuole accedere poiché ci si riferisce implicitamente al dominio sul quale ci si trova ovvero una *risorsa interna*. Il percorso è *relativo* rispetto al file attuale. Ad esempio:
  - ▶ contatti.html – punta al file *contatti.html* presente nella stessa directory del file corrente
  - ▶ ../*logo.png* – punta al file *logo.png* situato nella superiore rispetto a quella attuale
  - ▶ /images/*logo.png* – punta al file *logo.png* situato all'interno della cartella *images* collocata nella *root* del dominio attuale. Equivale a <http://dominio.com/images/logo.png>



# Link

Quando si fa click su un link, questo sostituisce il *contenuto corrente* della pagina con quello puntato dall'*attributo href*. Questo tipo di comportamento è idoneo per le risorse contenute nello stesso dominio di origine ma particolarmente scomodo quando si vuole accedere a risorse esterne...

- ▶ Forzare l'apertura delle risorse esterne al proprio sito in una tab dedicata migliora l'esperienza utente
- ▶ Obbliga in qualche modo l'utente a rimanere sul nostro sito anche dopo aver consultato la risorsa esterna, evitando la fastidiosa pratica del click sul pulsante *indietro* del browser

I link vengono aperti su una tab differente da quella attuale specificando l'*attributo target* valorizzandolo con *\_blank*.



# Esercizi

- ▶ Introduci nella pagina *index.html* due link, uno ad una risorsa *interna* (ad esempio una nuova pagina *contatti.html*) ed un altro ad una risorsa *esterna*
- ▶ Il link alla risorsa esterna deve aprire una nuova tab nel browser



# 4. Attributi



# Attributi

Tutti i tag HTML possono essere corredati di una serie di *attributi* al fine di fornire maggiori informazioni per un determinato elemento. Un esempio di attributo è stato applicato pocanzi sul tag a...

- ▶ *Href* è un attributo che determina l'indirizzo della risorsa
- ▶ *Target* è un attributo che definisce se il bersaglio di una risorsa (ad esempio, una nuova tab del browser)

Alcune regole fondamentali:

- ▶ Gli attributi sono definiti in forma di **chiave="valore"**
- ▶ Si trovano nel *tag di apertura* e subito **dopo il nome del tag**, seguito da uno spazio
- ▶ Sebbene siano riconosciuti anche i singoli apici, è sempre consigliato usare i doppi apici per racchiudere il valore dell'attributo
- ▶ Gli attributi che non necessitano di un valore, per standard, vogliono un *value* uguale al nome dell'attributo o in alternativa alla stringa *true*



# Attributi globali

Sebbene ciascun elemento HTML possa avere degli *attributi propri*, esiste un set di attributi definiti **globali** che possono essere applicati a qualsiasi elemento HTML esistente. Alcuni di questi sono ad uso e consumo esclusivo in HTML5 ed opportunamente segnalati nell'elenco che segue...

| HTML5 | Attributo       | Descrizione                                                                                                 |
|-------|-----------------|-------------------------------------------------------------------------------------------------------------|
| No    | accesskey       | Permette di definire un tasto che attiva / abilita il focus su uno specifico elemento                       |
| No    | class           | Contiene un elenco di classi CSS da applicare all'elemento                                                  |
| Si    | contenteditable | Definisce il contenuto dell'elemento come editabile o non editabile                                         |
| Si    | contextmenu     | Permette di definire un menu contestuale che appare alla pressione del tasto destro del mouse sull'elemento |



# Attributi globali

| HTML5 | Attributo  | Descrizione                                                                                                                                           |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Si    | data-*     | I data attribute permettono di definire dei valori particolari sull'elemento. Si sostituisce il simbolo * con un identificatore arbitrario ed univoco |
| No    | dir        | Stabilisce la direzione del testo all'interno dell'elemento                                                                                           |
| Si    | draggable  | Describe l'elemento come trascinabile con il mouse all'interno della pagina                                                                           |
| Si    | dropzone   | Permette di definire dove i dati trascinati devono essere copiati, spostati o collegati                                                               |
| Si    | hidden     | Nasconde l'elemento                                                                                                                                   |
| No    | id         | Permette di definire un id univoco per l'elemento                                                                                                     |
| No    | lang       | Definisce la lingua del contenuto dell'elemento                                                                                                       |
| Si    | spellcheck | Forza il controllo ortografico sul contenuto digitato                                                                                                 |



# Attributi globali

| HTML5 | Attributo | Descrizione                                                                                                     |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------|
| No    | style     | Stili CSS <i>inline</i>                                                                                         |
| No    | tabindex  | Definisce l'ordine sul focus dell'elemento rispetto alla pressione del tasto <i>tab</i>                         |
| No    | title     | Ulteriori informazioni da mostrare a schermo quando il puntatore del mouse rimane in <i>hover</i> sull'elemento |
| Si    | translate | Forza la traduzione del contenuto dell'elemento                                                                 |



# Accesskey

**Accesskey** è un attributo che permette di definire un tasto che attivi o posizioni il focus sull'elemento sul quale viene definito:

- ▶ Il *focus* consiste nel posizionare il cursore su un elemento per una pronta interazione, ad esempio il riempimento di un *input field* di un form
- ▶ L'*attivazione* di un elemento consiste nell'eseguire l'*action* associata all'elemento. Nel caso di un link, ad esempio, equivale all'esecuzione del click (e quindi all'apertura del link)

Il valore associato all'attributo *accesskey* deve corrispondere al carattere premuto sulla tastiera che sarà delegato al focus/attivazione dell'elemento. Ad esempio:

```
<a href="https://www.repubblica.it" accesskey="r">Repubblica.it</a>
```

# Accesskey: attivazione

L'attivazione degli accesskey varia in base al browser ed al sistema operativo in uso...

| Browser           | Windows                     | Linux                       | Mac                           |
|-------------------|-----------------------------|-----------------------------|-------------------------------|
| Internet Explorer | [alt] + accesskey           | -                           | -                             |
| Chrome            | [alt] + accesskey           | [alt] + accesskey           | [control] + [alt] + accesskey |
| Firefox           | [alt] + [shift] + accesskey | [alt] + [shift] + accesskey | [control] + [alt] + accesskey |
| Safari            | [alt] + accesskey           | -                           | [control] + [alt] + accesskey |
| Opera             | [alt] + accesskey           | [alt] + accesskey           | [alt] + accesskey             |



# Esercizi

- ▶ Creare un file "rassegna-stampa.html" contenente un elenco di link alle maggiori testate giornalistiche online
- ▶ Applicare una access key a ciascun link costituita dalla lettera iniziale del nome della testata



# Class

L'attributo **class** specifica un set di classi CSS per un determinato elemento. Le classi CSS devono essere definite in uno *style sheet* appositamente referenziato nell'*head* del file HTML, sia esso *inline* che esterno, per applicare degli stili particolari a degli elementi.

Le classi CSS permettono inoltre di referenziare un particolare insieme di elementi all'interno del *DOM* in script *JavaScript* al fine di applicare modifiche a più elementi contemporaneamente.

```
<elemento class="nomeclasse">link</elemento>
```

Per definire più classi CSS, separarle con uno spazio. I nomi delle classi devono inoltre seguire delle regole ben definite:

- ▶ Deve iniziare con un carattere A-Z / a-z
- ▶ Possono seguire lettere, numeri, il simbolo di sottrazione e l'underscore



# Data Attributes

I **data attribute** sono particolari attributi usati per contenere informazioni arbitrarie da associare agli elementi all'interno del DOM. Il valore contenuto nei data attribute viene solitamente usato da codice JavaScript per migliorare l'esperienza utente.

I data attribute sono composti da:

- ▶ Un nome costituito da un prefisso *data-* ed un suffisso arbitrario
  - ▶ Il suffisso deve contenere solo lettere lowercase
  - ▶ Le parole che compongono il suffisso devono essere separate dal simbolo – (dash-case)
- ▶ Un valore in formato stringa

```
<element data-attribute-name="value">
```



# Draggable

L'attributo **draggable** permette di stabilire se un elemento è *trascinabile* all'interno della pagina web. Questo particolare attributo è fondamentale per creare delle operazioni di drag&drop tra elementi della pagina.

Alcuni elementi come link ed immagini sono *draggable* per default.

```
<elemento draggable="true | false | auto"></element>
```

I valori ammessi da *draggable* sono:

- ▶ true – l'attributo è *draggable*
- ▶ false – l'attributo non è *draggable*
- ▶ auto – usa le regole di default del browser



# Droppable

L'attributo **droppable** permette di stabilire il comportamento da applicare qualora su un elemento venga rilasciato un altro elemento di tipo *draggable*.

```
<elemento droppable= "copy | move | link"></elemento>
```

- ▶ copy – copia il contenuto dell'elemento trascinato
- ▶ move – sposta il contenuto dell'elemento trascinato
- ▶ link – crea un collegamento all'elemento trascinato



# Id

L'attributo **id** serve ad indicare un *identificatore univoco* per un elemento HTML: l'univocità deve essere garantita all'interno del documento HTML.

Gli *id* sono utilizzati per associare uno stile CSS ad al più un elemento all'interno del documento HTML e per manipolare l'elemento referenziato tramite JavaScript.

```
<element id="nome-identificatore"></element>
```

Gli *id* devono contenere almeno un carattere ed essere *privi di spazi*.



# Style

**Style** permette di definire degli stili CSS *inline* per l'elemento sul quale viene definito l'attributo. Le regole CSS definite attraverso l'attributo *style* hanno precedenza rispetto a tutte le altre regole definite nel tag `<style>` all'interno dell'head della pagina o nei fogli di stile CSS esterni.

Il value dell'attributo *style* consiste in una o piú regole CSS separate dal simbolo ; ...

```
<elemento style="regola-css: valore"></elemento>
```



# Title

**Title** permette di fornire informazioni aggiuntive in merito al contenuto di uno specifico elemento HTML. Viene ampiamente usato dai motori di ricerca, pertanto è opportuno valutarne il contenuto confrontandosi con un esperto SEO.

Il contenuto dell'attributo *title* viene visualizzato a schermo all'interno di un *tooltip* qualora si verifichi un evento *hover* su un elemento HTML. La sintassi...

```
<elemento title="Titolo da mostrare nel tooltip"></elemento>
```

**HTML**



# 5. Liste



# Liste

Le *liste* consistono in elenchi di elementi, solitamente testuali. HTML permette di distinguere due tipi differenti di liste...

- ▶ Ordinate (*ordered list*) distinte da un simbolo numerico che precede ciascun elemento della lista. Ad esempio...
  1. Elemento 1
  2. Elemento 2
  3. Elemento 3
- ▶ Non ordinate (*unordered list*) distinte da un simbolo generico, solitamente un elemento geometrico, che precede ciascun elemento della lista. Ad esempio...
  - Elemento 1
  - Elemento 2
  - Elemento 3



# Liste non ordinate

Le liste non ordinate si definiscono attraverso il tag `<ul>` e sono un elemento di tipo *block*. Di default, il simbolo che precede gli elementi della lista è un *disc* (cerchio nero).

```
<ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ul>
```



# Liste non ordinate

È possibile definire il simbolo posto precedentemente ciascun elemento della lista attraverso l'attributo *type*.

```
<ul type="disc | circle | square"></ul>
```

| Valore | Descrizione                    |
|--------|--------------------------------|
| disc   | Valore di default – Un cerchio |
| circle | Una circonferenza              |
| Square | Un quadrato                    |



# Liste ordinate

Le liste ordinate si definiscono attraverso il tag `<ol>` e sono un elemento di tipo *block*. A differenza delle liste *non ordinate*, le **liste ordinate** precedono ciascuno dei loro elementi con dei simboli che stabiliscono un ordine cardinale. Di default, il simbolo che precede gli elementi della lista è un numero decimale.

```
<ol>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ol>
```



# Liste ordinate

È possibile definire il simbolo posto precedentemente ciascun elemento della lista ordinata attraverso l'attributo *type*.

```
<ol type="1 | a | A | i | I"></ol>
```

| Valore | Descrizione                                        |
|--------|----------------------------------------------------|
| 1      | Valore di default – Numeri decimali (1, 2, 3)      |
| a      | Caratteri alfabetici in <i>lowercase</i> (a, b, c) |
| A      | Caratteri alfabetici in <i>uppercase</i> (A, B, C) |
| i      | Numeri romani in <i>lowercase</i> (i, ii, iii, iv) |
| I      | Numeri romani in <i>uppercase</i> (I, II, III, IV) |



# Liste ordinate - Attributi

Altri attributi utili alla gestione di liste ordinate:

- ▶ **reversed**: se presente, il valore che determina la cardinalità dell'elemento verrá presentato in ordine decrescente
- ▶ **start**: permette di definire il valore iniziale che determina la cardinalità dell'elemento nella lista



# Esercizi

- ▶ Creare un file "animali.html" contenente una lista di almeno 3 elementi per ciascun tipo dei seguenti:
  - ▶ Uccelli
  - ▶ Pesci
  - ▶ Rettili
- ▶ Applicare un data-attribute a ciascun elemento della lista



# 6. Tabelle



# Tabelle

Le **tabelle** sono usate per mostrare dati strutturando la visualizzazione in *righe e colonne*. Sono identificate dal tag `<table>` e si dividono in 3 parti ben distinte:

- ▶ Table Head – contiene la parte di intestazione situata in cima alla tabella. Viene usata per mostrare il titolo di ciascuna colonna
- ▶ Table Body – contiene il corpo della tabella ovvero quella parte delegata a visualizzare i dati all'interno delle opportune **celle**
- ▶ Table Footer – così come l'intestazione, serve a mostrare il titolo di ciascuna colonna ai piedi della tabella

Per molto tempo le tavole sono state impropriamente usate per definire in modo approssimativo la struttura (layout) di una pagina. Questo è un utilizzo assolutamente improprio delle tavole ed è del tutto scoraggiato: il layout si definisce infatti attraverso i CSS!



# Thead e Tfoot

I tag **<thead>** e **<tfoot>** servono a raggruppare le informazioni di intestazione di una tabella. I browser possono usare questi tag per gestire lo scroll separato delle varie parti della tabella, nonché ottimizzare la stampa di tabelle particolarmente grandi che possono estendersi su più pagine, mostrando nella parte superiore ed inferiore dello stampato l'intestazione della tabella.

- ▶ Contengono al più una riga
- ▶ Le celle sono individuate dal tag **<th></th>**



# Tbody

Il tag **<tbody>** serve a raggruppare i dati di una tabella secondo la struttura a colonne definita nella testata e nel pié della tabella. A differenza dei tag di intestazione e di pié di tabella, il corpo di una tabella utilizza un tag differente per la creazione di una cella in una specifica riga.

- ▶ Contiene un numero arbitrario di righe
- ▶ Le celle sono individuate dal tag `<td></td>`



# Righe e celle

Le tabelle mostrano i dati all'interno di apposite **celle** poste in corrispondenza di una o più colonne

- ▶ Ciascuna cella è preposta a contenere un valore
- ▶ Le celle sono identificate dal tag `<td>` (*table draw*)
- ▶ Le celle sono contenute all'interno di una **riga**

Le righe servono a raggruppare orizzontalmente le celle di una tabella ed a separare i valori di ciascuna riga della tabella. Una riga può contenere un numero arbitrario di elementi `td`

- ▶ La riga determina il ritorno a capo di una sequenza di `td`
- ▶ Le righe sono identificate dal tag `<tr>` (*table draw*)



# Intestazione / pié di tabella

L'intestazione di una tabella si realizza combinando i tag `thead`, `tr` e `th`:

```
<table>  
  <thead> <!-- o tfoot -->  
    <tr>  
      <th>Colonna 1</th>  
      <th>Colonna 2</th>  
      <th>Colonna 3</th>  
    </tr>  
  </thead> <!-- o /tfoot -->  
</table>
```



# Corpo della tabella

Il corpo di una tabella si realizza combinando i tag tbody, tr e td:

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



# Esercizi

- ▶ Simulare, attraverso l'uso delle tavole, la visualizzazione di un elenco di utenti con le seguenti informazioni:
  - Id
  - Nome e Cognome
  - Email
  - Ruolo utente
  - Actions (elimina, modifica etc...)
- ▶ Porre un data-attribute che distingua, per ciascun set di dati, l'id dell'utente: valutare la posizione migliore nel contesto di una tabella HTML!



# 7. Form



# Form

Un **form** è una particolare porzione di documento che contiene elementi denominati **control** (come le checkbox, i radio button, i menu dropdown o select etc...) corredati da apposite etichette. È un elemento di tipo block.

L'elemento form viene utilizzato per gestire l'interazione utente nell'ambito della compilazione di appositi campi contenenti informazioni ben definite. Ad esempio una schermata di login di un sito web è composta da un form, così come un modulo per la creazione di un profilo su una piattaforma.



# Form

HTML implementa soltanto la parte visiva del form **privando il modulo della logica che gestisce i dati**: questi devono essere gestiti da un sistema ad hoc: solitamente i dati vengono inviati ad uno script PHP o ASP che ne verifica la genuinità del contenuto ed esegue eventualmente una serie di operazioni come aggiunta / modifica / rimozione di record all'interno di un database.

Il form viene "inviaTO" attraverso uno specifico bottone di *submit*.



# Form

HTML implementa soltanto la parte visiva del form **privando il modulo della logica che gestisce i dati**: questi devono essere gestiti da un sistema ad hoc: solitamente i dati vengono inviati ad uno script PHP o ASP che ne verifica la genuinità del contenuto ed esegue eventualmente una serie di operazioni come aggiunta / modifica / rimozione di record all'interno di un database.

Il form viene "inviaTO" attraverso uno specifico bottone di *submit*.



# Form

Il Tag preposto alla creazione di form è <form></form>: tutti i campi del form e le relative etichette devono essere poste al suo interno. È possibile definire una serie di attributi che stabiliscono il comportamento del form anche rispetto alla risorsa delegata all'elaborazione dei dati...

| HTML5 | Attributo      | Descrizione                                                                                                                                                                                                  |
|-------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No    | accept-charset | Specifica la codifica dei caratteri da usare nel <i>submit</i><br>Ad es. UTF-8                                                                                                                               |
| No    | action         | L'indirizzo, assoluto o relativo, che contiene la risorsa<br>delegata all'elaborazione dei dati contenuti nei <i>fields</i>                                                                                  |
| Si    | autocomplete   | Abilita o disabilita le funzionalità di autocompletamento.<br>Accetta le stringhe on/off                                                                                                                     |
| No    | enctype        | Stabilisce la codifica con la quale inviare i dati alla risorsa<br>stabilita nell'attributo <i>action</i> . Ad es. <i>application/x-www-form-urlencoded</i> , <i>multipart/form-data</i> e <i>text/plain</i> |



# Form

| HTML5 | Attributo  | Descrizione                                                                                                                                                                                                              |
|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No    | method     | Il verb <i>HTTP</i> mediante il quale inviare il form. I valori accettati sono <i>get</i> e <i>post</i>                                                                                                                  |
| No    | name       | Permette di stabilire un nome per il form                                                                                                                                                                                |
| Si    | novalidate | Disabilita le funzionalità di validazione di default del browser affidandosi solo ed esclusivamente a validazioni di terze parti (es. JavaScript)                                                                        |
| No    | target     | Stabilisce le modalità di visualizzazione della risposta restituita dalla risorsa: <ul style="list-style-type: none"><li>• <i>_blank</i></li><li>• <i>_parent</i></li><li>• <i>_self</i></li><li>• <i>_top</i></li></ul> |



# Input field

Gli **input field** rappresentano i campi contenuti nel Form. Ciascun input field determina una coppia chiave-valore che concorrerà a definire il contenuto della richiesta inviata alla action.

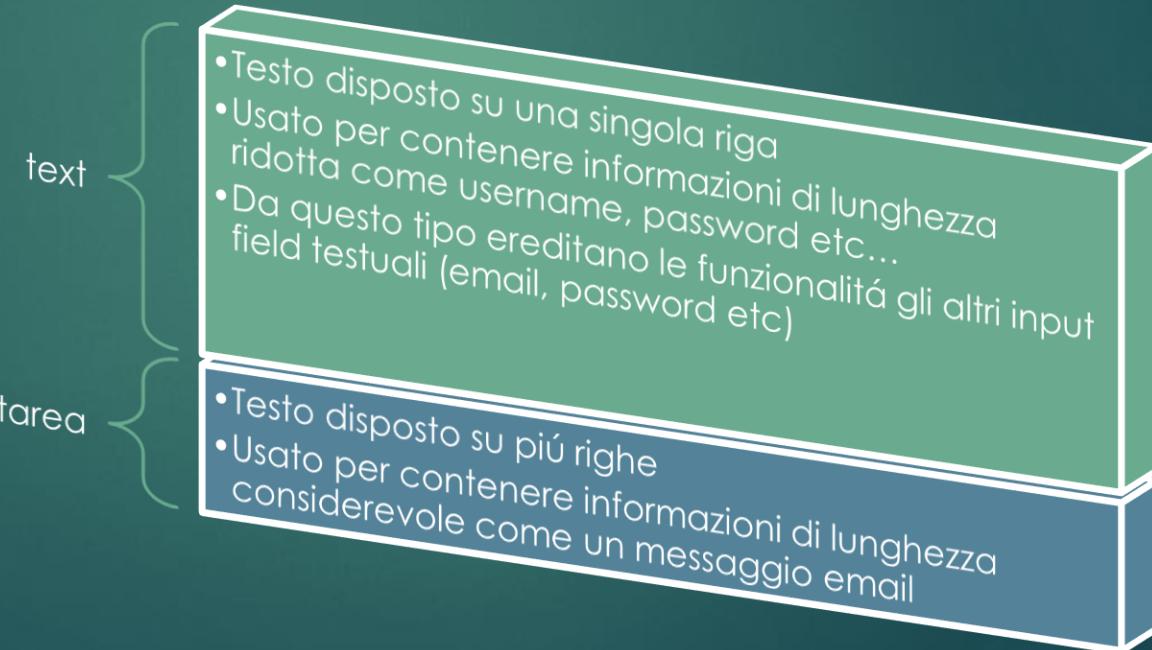
Gli input field si definiscono attraverso il tag **<input>** che **non prevede chiusura**. Esistono diversi tipi di input field:

- ▶ button – un bottone
- ▶ checkbox – una casella che rappresenta un *flag*
- ▶ email – equivalente al tipo text, richiede al browser di validare la stringa inserita affinché sia un indirizzo email
- ▶ password – equivalente al tipo text, nasconde il contenuto del campo mostrando, per ciascun carattere, un simbolo stabilito dal browser
- ▶ text – accetta una stringa di testo e viene visualizzato su una sola riga
- ▶ textarea – accetta una stringa di testo e viene visualizzato su più righe



# Input field testuali

Gli **input field testuali** sono quegli input field valorizzabili attraverso delle *stringhe di testo*. Esistono diversi tipi di input field testuali, tra i quali due risultano essere i piú importanti...





# Input field testuali

Esistono altri tipi di *input field testuali* che **estendono le funzionalità del tipo text**. Qualora il browser utilizzato non supporti questo tipo di campi input, verrá automaticamente applicata la medesima gestione usata per il tipo *text*.

L'utilizzo di questi tipi particolari permette al browser di applicare delle *regole di validazione*: definire un input field di tipo *email* ad esempio, segnalerá al browser che il valore del campo dovrá essere un indirizzo *email sintatticamente valido*.



# Input field testuali

## Date

Mostra un *datepicker* per la selezione di una data

## Datetime-local

Estende le feature di *date* con ore e minuti

## Email

Forza la validazione della stringa ad un indirizzo email

## Month

Mostra un *datepicker* per la selezione di mese - anno

## Number

Permette l'inserimento e la validazione di un numero

## Password

Il contenuto viene celato con un *bullet* per carattere



# Input field testuali

## Search

Il campo contiene un pulsante per pulire il value

## Tel

Forza la validazione ad un numero di telefono

## Time

Permette l'inserimento di ore, minuti e secondi

## URL

Forza la validazione ad un Uniform Resource Locator

## Week

Permette la selezione di una settimana dell'anno



# Input field non testuali

Gli input field **non testuali** appartengono a quella categoria di elementi che hanno un valore fisso, scelto dall'utente nel gruppo a cui appartiene per mezzo della selezione e non della digitazione come avviene per gli *input field testuali*.

Un particolare tipo di *input field* dei form che si inserisce nell'insieme dei campi *non testuali* è rappresentato dal tag *select*.





# Input field non testuali

## Button

Un bottone cliccabile. È preferibile usare il tag button

## Checkbox

Una casella che permette la selezione di più opzioni

## Color

Mostra un *color picker* per la selezione di un colore (HEX)

## File

Permette di eseguire l'upload del file selezionato

## Hidden

Un campo non visibile a schermo

# Input field non testuali

## Image

Crea un *submit button* con l'immagine specificata

## Radio

Permette di selezionare un valore tra n scelte

## Range

Mostra uno *slider* per la selezione di un numero

## Reset

Un bottone che ripristina gli *input field* ai valori di default

## Submit

Un bottone per l'invio (submit) del form alla action



# Input field: parametri

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo    | Descrizione                                                                                                                                                                                                                                                        |
|-------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No    | accept       | Specifica, per gli input field di tipo <i>file</i> , il <i>file type</i> selezionabile attraverso: <ul style="list-style-type: none"><li>• L'estensione del file</li><li>• Il <i>media type</i></li><li>• Wildcard come <i>audio/*, video/*, image/*</i></li></ul> |
| No    | alt          | Valido solo per gli input field <i>image</i> , stabilisce il testo da visualizzare in sostituzione all'immagine qualora questa non sia disponibile                                                                                                                 |
| Si    | autocomplete | Abilita o disabilita le funzionalità di autocompletamento. Accetta le stringhe on/off                                                                                                                                                                              |
| Si    | autofocus    | Stabilisce l' <i>input field</i> sul quale posizionare il focus al caricamento della pagina                                                                                                                                                                        |



# Input field: parametri

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo  | Descrizione                                                                                                                                                                                                                                                         |
|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No    | checked    | Valido per i tipi <i>checkbox</i> e <i>radio</i> , definisce gli input field preselezionati al caricamento della pagina                                                                                                                                             |
| Si    | dirname    | Quando presente, include la direzione del testo (ad es. <i>ltr</i> / <i>rtl</i> ) al <i>value</i> del campo di testo. Il valore dell'attributo deve essere equivalente a <i>nomeinput.dir</i> dove <i>nomeinput</i> è il valore inserito nell'attributo <i>name</i> |
| No    | disabled   | Disabilita l'input field                                                                                                                                                                                                                                            |
| Si    | form       | Quando un input field è collocato all'esterno del tag <i>form</i> , questo attributo permette di specificare l' <i>id</i> del form di appartenenza (dove per <i>id</i> si intende il valore del parametro <i>id</i> sul tag <i>form</i> )                           |
| Si    | formaction | Permette di specificare una <i>action</i> differente da quella specificata di default nel tag <i>form</i> . Valido solo per i tipi <i>submit</i> ed <i>image</i>                                                                                                    |



# Input field: parametri

| HTML5 | Attributo      | Descrizione                                                                                                                                                                                                                                                                                                  |
|-------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Si    | formenctype    | Valido solo sui type <i>submit</i> e <i>image</i> , stabilisce la codifica usata nell'invio del form: <ul style="list-style-type: none"><li>• application/x-www-form-urlencoded</li><li>• multipart/form-data</li><li>• text/plain</li></ul>                                                                 |
| Si    | formmethod     | Valido solo sui type <i>submit</i> e <i>image</i> , stabilisce il verbo HTTP usato nel submit del form alla action <ul style="list-style-type: none"><li>• get</li><li>• post</li></ul>                                                                                                                      |
| Si    | formnovalidate | Disabilita la validazione del browser sul singolo input field sul quale è definito                                                                                                                                                                                                                           |
| Si    | formtarget     | Valido solo sui type <i>submit</i> e <i>image</i> , stabilisce dove mostrare la risposta ricevuta dalla action: <ul style="list-style-type: none"><li>• _blank (nuova tab)</li><li>• _self (frame corrente)</li><li>• _parent (frame genitore)</li><li>• _top (sostituisce il body della finestra)</li></ul> |



# Input field: parametri

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo   | Descrizione                                                                                                              |
|-------|-------------|--------------------------------------------------------------------------------------------------------------------------|
| Si    | height      | Valido solo per il type <i>image</i> , stabilisce l'altezza in px                                                        |
| Si    | list        | Id della datalist con le opzioni da visualizzare in un menu a tendina del campo input (non funziona in IE <= 9 e Safari) |
| Si    | max         | Value massimo consentito                                                                                                 |
| No    | maxlength   | Numero massimo di caratteri consentiti                                                                                   |
| Si    | min         | Value minimo consentito                                                                                                  |
| Si    | multiple    | Permette all'utente di inserire più di un valore                                                                         |
| No    | name        | Il nome associato all' <i>input field</i>                                                                                |
| Si    | regexp      | Espressione regolare che il campo deve rispettare per essere validato                                                    |
| Si    | placeholder | Un suggerimento da mostrare nel campo quando questo non ha un <i>value</i> associato                                     |



# Input field: parametri

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo | Descrizione                                                                          |
|-------|-----------|--------------------------------------------------------------------------------------|
| No    | readonly  | L'elemento viene mostrato in sola lettura                                            |
| Si    | required  | Definisce l'elemento come richiesto affinché il form superi la validazione           |
| No    | size      | Larghezza in caratteri di un campo input                                             |
| No    | src       | Valido solo per il type <i>image</i> , indica il percorso all'immagine da utilizzare |
| Si    | step      | Intervallo da applicare all'aumento/decremento di un campo numerico                  |
| No    | type      | Stabilisce il tipo di input field                                                    |
| No    | value     | Contiene il valore associato al campo                                                |
| No    | width     | Valido solo per il type <i>image</i> , stabilisce la larghezza in px                 |



# Esercizi

- ▶ Realizzare un form di accesso (login) che contenga i seguenti field opportunamente tipizzati:
  - Email
  - Password
- ▶ Realizzare un form di registrazione che contenga i seguenti field opportunamente tipizzati:
  - Email
  - Conferma email
  - Password
  - Conferma password
  - Data di nascita
  - Consenso trattamento dati



# 8. Sezioni



# Sezioni di un documento

Concettualmente è possibile dividere una pagina web in porzioni (o sezioni) ben definite. Ad esempio...

- ▶ Un header (testata)
- ▶ Un footer (pié di pagina)
- ▶ Il content (contenuto)

Prima dell'avvento di HTML5, questo tipo di sezioni erano gestite attraverso un tag generico (div). Ad oggi sono stati introdotti tag dedicati alla strutturazione delle parti della pagina per favorire, tra l'altro, una più facile interpretazione del documento ai motori di ricerca.



# Header

Il tag **header** descrive un elemento di tipo *block* che permette di definire una serie di contenuti *introduttivi*, o un set di link di navigazione. Il tag in questione contiene, solitamente:

- ▶ Uno o più elementi di intestazione (tag h)
- ▶ Un **logo** per il progetto / attività trattato dal sito web
- ▶ Menu di navigazione
- ▶ Altre eventuali

È possibile (ma sconsigliato) definire un numero arbitrario di questo tipo di sezioni. Ad ogni modo, è **formalmente scorretto** posizionare il tag header all'interno di un altro tag header, footer o address.



# Footer

Il tag **footer** descrive un elemento di tipo *block* che permette di definire una serie di contenuti utili alla navigazione ed alla presentazione dei diritti di autore. Il tag in questione contiene, solitamente:

- ▶ Informazioni su autore e copyright del sito web
- ▶ Un menu di navigazione
- ▶ Informazioni di contatto
- ▶ Sitemap
- ▶ Shortcut per tornare alla testata (header) del sito

È possibile (ma sconsigliato) definire un numero arbitrario di questo tipo di sezioni. Ad ogni modo, è **formalmente scorretto** posizionare il tag footer all'interno di un altro tag header, footer o address.



# Menu di navigazione

Quando nasce l'esigenza di raggruppare una serie di link con lo scopo di costituire un **menu di navigazione**, è buona pratica inserire il set di elementi di navigazione all'interno del tag **nav**.

Il tag **nav** è utile in particolare

- ▶ Definire strutturalmente un menu di navigazione nel documento
- ▶ Migliorare l'esperienza utente alle persone diversamente disabili, poiché questo elemento viene ignorato dagli *screen reader* in fase di rendering



# Section

Il tag **section** rappresenta una sezione generica del documento, di tipo *block*, caratterizzata da un contenuto accomunato da uno specifico **tema**. Ad esempio, in una dashboard potremmo trovare una section per ciascuna di queste parti del template...

- ▶ Una «overview» che ricapitola le informazioni chiave (dati) all'utente
- ▶ Una sezione analitica con dei grafici di vario genere
- ▶ Una parte relativa a dei suggerimenti di utilizzo degli strumenti presenti

La **section** non va utilizzata quando l'unica esigenza consiste nell'applicare degli style CSS ad blocco ed ai suoi figli.



# Article

Il tag **article** nasce per delimitare un *articolo*, o un contenuto che è *indipendente rispetto alla pagina* nel quale è posizionato nel sito web.

Un *article* è solitamente *fruibile indipendentemente* dal resto del sito web. Alcuni esempi:

- ▶ Il post di un blog
- ▶ Un commento in un social network
- ▶ Una news sul sito web di una testata giornalistica
- ▶ Il post di un forum



# Blocchi generici

È possibile strutturare la pagina in **blocchi generici** attraverso il tag **div**. L'elemento *div*, di tipo *block*, svolge la funzione di **container** ed incapsula al suo interno altri elementi HTML con lo scopo di organizzare il documento in **sezioni distinte** alle quali applicare, eventualmente, degli stili definiti tramite CSS.

Si impiega il tag *div* anche quando **nessun tag HTML è un valido candidato rispetto al suo contenuto**. Per la scomposizione in **sezioni** della pagina ci si rifá alla slide «Section»

```
<div> ... </div>
```

# Come scegliere?





# Esercizi

- ▶ Creare una pagina che contenga:
  - Una testata con il nome di un brand ed un menu di navigazione
  - Un piede di pagina con l'indirizzo della sede dell'azienda ed i copyright
  - Del contenuto che contenga un articolo di giornale
  - Informazioni biografiche dell'autore dell'articolo

Puoi prelevare un articolo da una qualsiasi testata giornalistica e riproporlo nella pagina strutturandolo opportunamente in HTML. La sezione dedicata all'autore può contenere un *lorem ipsum*.



# 9. Multimedia



# Multimedialità

Oltre agli elementi testuali, concorrono a comporre il contenuto di una pagina web i **contenuti multimediali** quali *immagini, audio e video*.

Quando si lavora ad una pagina web è fondamentale ricordare che:

- ▶ il cervello umano recepisce in modo più facile i contenuti multimediali, specialmente le **immagini**, rispetto al testo proprio come avviene mettendo a confronto il linguaggio non verbale con quello verbale
- ▶ Il testo richiede un processo di lettura e di elaborazione *direttamente proporzionale* al quantitativo di testo

I contenuti multimediali non sono parte integrante del codice sorgente del documento (come avviene ad esempio per il testo) bensì una semplice **referenza esterna**



# Prima di cominciare...

I tag che permettono di lavorare con file multimediali introdotti in HTML5 lavorano in stretta dipendenza al tag **source**.

Il tag **source** permette di definire più di una sorgente da usare come elemento multimediale nei tag **audio**, **video** e **picture**. In particolare, viene usato per **indicare dei file alternativi** che il browser può utilizzare dipendentemente al supporto rispetto al file stesso.

```
<source src="audio.mp3">
```

| HTML5 | Attributo | Descrizione                                                                    |
|-------|-----------|--------------------------------------------------------------------------------|
| Si    | media     | Accetta una media query secondo il linguaggio CSS                              |
| Si    | src       | Permette di definire il file sorgente (per i tag <b>audio</b> e <b>video</b> ) |
| Si    | srcset    | Permette di definire il file sorgente (per il tag <b>picture</b> )             |
| Si    | type      | Il mime-type della risorsa                                                     |



# Immagini

Le immagini vengono descritte in un documento HTML mediante il tag **img** (image) che prevede due *attributi obbligatori*:

- ▶ Src ovvero la **source** (sorgente) che determina, attraverso un **indirizzo assoluto o relativo**, la posizione del file immagine da mostrare
- ▶ Alt ovvero **alternative**, un testo da visualizzare in caso di inesistenza del file immagine definito attraverso l'attributo src

Il tag **img** può inoltre essere annidato nel tag `<a>` per creare delle immagini che al click si comportano come un link testuale.

```

```



# Immagini

HTML5 introduce, tra le nuove feature, anche un nuovo tag per la gestione delle immagini: **picture**. La necessità di un nuovo tag nasce dalla richiesta, da parte degli sviluppatori, di sempre maggiore flessibilità nell'uso delle immagini nei documenti per il web.

L'uso più idoneo di questo tag consiste nella scelta di immagini in relazione al *responsive web design*. Il vantaggio è considerevole, poiché invece di avere una sola immagine scalata (con le possibili perdite di qualità che ne derivano) si possono usare, qualora necessario, immagini ad alta risoluzione.

Il browser **determinerà in autonomia l'immagine migliore** in relazione ai fattori più rilevanti. **È obbligatorio** inoltre **l'uso del tag img annidato in picture** per garantire la retrocompatibilità con i browser più vecchi.



# Audio

È possibile includere un file audio nelle pagine web da mostrare in un player integrato al browser. Attualmente, i file supportati sono:

- ▶ Mp3 – audio/mpeg
- ▶ Ogg – audio/ogg
- ▶ Wav – audio/wav

Il tag **audio** è una feature introdotta con HTML5. Permette di definire più sorgenti alternative attraverso il tag source.

```
<audio controls>  
  <source src="audio.mp3" type="audio/mpeg">  
</audio>
```

# Audio - Parametri

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo | Descrizione                                                                                                                                                                                                   |
|-------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Si    | autoplay  | Riproduzione automatica al caricamento del file                                                                                                                                                               |
| Si    | controls  | Mostra i controlli sul player come il tasto play/pausa                                                                                                                                                        |
| Si    | loop      | Riproduzione del file audio in loop                                                                                                                                                                           |
| Si    | muted     | Muta la traccia audio                                                                                                                                                                                         |
| Si    | preload   | Stabilisce come il file deve essere caricato, ad esempio al caricamento della pagina o alla pressione del tasto play <ul style="list-style-type: none"><li>• Auto</li><li>• Metadata</li><li>• None</li></ul> |
| Si    | src       | Indirizzo al file sorgente                                                                                                                                                                                    |



# Video

Così come per i file audio, anche i video sono visualizzabili in una pagina web nel il player integrato del browser. Sono supportati:

- ▶ Mp4 – video/mp4
- ▶ Ogg – video/ogg
- ▶ webm – video/webm

Il tag **video** è una feature introdotta con HTML5. Permette di definire più sorgenti alternative attraverso il tag source.

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

# Video - Parametri

© Andrea Tantimonaco @ [linkedin.com/in/andrea-tantimonaco](https://linkedin.com/in/andrea-tantimonaco) - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato

| HTML5 | Attributo | Descrizione                                                                                                                                                                                                   |
|-------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Si    | autoplay  | Riproduzione automatica al caricamento del file                                                                                                                                                               |
| Si    | controls  | Mostra i controlli sul player come il tasto play/pausa                                                                                                                                                        |
| Si    | height    | Altezza in pixel del player                                                                                                                                                                                   |
| Si    | loop      | Riproduzione del file audio in loop                                                                                                                                                                           |
| Si    | muted     | Muta la traccia audio                                                                                                                                                                                         |
| Si    | poster    | Indirizzo all'immagine di anteprima del video                                                                                                                                                                 |
| Si    | preload   | Stabilisce come il file deve essere caricato, ad esempio al caricamento della pagina o alla pressione del tasto play <ul style="list-style-type: none"><li>• Auto</li><li>• Metadata</li><li>• None</li></ul> |
| Si    | src       | Indirizzo al file sorgente                                                                                                                                                                                    |
| Si    | width     | Larghezza in pixel del player                                                                                                                                                                                 |



# Esercizi

- ▶ Aggiungere alla pagina creata nell'esercizio del paragrafo Sezioni:
  - ▶ Un logo nella testata
  - ▶ Un video all'interno dell'articolo



# 10. Includere risorse



# Risorse esterne

HTML è un linguaggio che ha, come si è visto fino ad ora, degli evidenti limiti funzionali:

- ▶ Non permette la gestione dell'**interazione utente (contenuto statico)**
- ▶ Il posizionamento degli elementi e la formattazione dei contenuti è delegato agli attributi dei tag. Tra l'altro, gli attributi che riguardano lo *styling* degli elementi sono *deprecati in HTML5*

Questi limiti sono superabili con l'utilizzo di altri linguaggi, facilmente integrabili nelle pagine web. In particolare stiamo parlando di:

- ▶ CSS
- ▶ JavaScript



# Risorse esterne

CSS

- Raggruppa gli elementi in classi ed id
- Permette di definire in modo puntuale il posizionamento degli elementi all'interno del documento

JavaScript

- Permette la gestione dell'**interazione utente**
- Usato, tra l'altro, per implementare **logiche di validazione personalizzate** nei form
- Gestione dei **processi asincroni** come chiamate a webservice



# Risorse esterne

Le risorse esterne possono essere incluse con due differenti approcci:

## Stili interni / Inline

- Il codice sorgente viene scritto direttamente nel file HTML
- Il codice è collocato tra il tag di apertura e quello di chiusura (internal) o tramite attributo sul tag (inline)

## Stili esterni

- Il codice è contenuto in un file esterno a quello HTML
- Il file viene referenziato dal tag attraverso uno specifico attributo



# Risorse esterne: CSS inline

I CSS possono essere definiti:

- ▶ Inline attraverso l'attributo **style** / internamente tramite il tag **style**
- ▶ Esternamente attraverso il tag **link**

Entrambi i tag devono essere definiti, secondo gli standard, all'interno del tag **head** del documento. Il tag **style** prevede 2 attributi:

| HTML5 | Attributo | Descrizione                                                                                           |
|-------|-----------|-------------------------------------------------------------------------------------------------------|
| No    | media     | Accetta una media query secondo il linguaggio CSS per la quale utilizzare le regole descritte nel tag |
| No    | type      | Specifica il tipo di dato. Valorizzare a text/css                                                     |



# Il tag link

Il tag **link** permette, tra l'altro, di includere dei **file CSS esterni** con metodologia outline. In generale viene impiegato per inserire nel documento delle *risorse esterne* di varia natura.

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

| HTML5 | Attributo   | Descrizione                                                                            |
|-------|-------------|----------------------------------------------------------------------------------------|
| Si    | crossorigin | Specifica come l'elemento debba gestire richieste cross-origin                         |
| No    | href        | URL della risorsa da includere                                                         |
| No    | hreflang    | Lingua del testo presente nel documento incluso                                        |
| No    | media       | Il documento verrà mostrato solo qualora venga soddisfatta la media query CSS indicata |
| No    | rel         | Relazione tra il documento corrente e quello incluso attraverso il tag link            |



# Il tag link

Il tag **link** permette, tra l'altro, di includere dei **file CSS esterni**. In generale viene impiegato per inserire nel documento delle *risorse esterne* di varia natura.

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

| HTML5 | Attributo   | Descrizione                                                                            |
|-------|-------------|----------------------------------------------------------------------------------------|
| Si    | crossorigin | Specifica come l'elemento debba gestire richieste cross-origin                         |
| No    | href        | URL della risorsa da includere                                                         |
| No    | hreflang    | Lingua del testo presente nel documento incluso                                        |
| No    | media       | Il documento verrà mostrato solo qualora venga soddisfatta la media query CSS indicata |
| No    | rel         | Relazione tra il documento corrente e quello incluso attraverso il tag link            |
| Si    | sizes       | Valido solo per rel="icon", indica la dimensione WxH                                   |



# Il tag link: attributo rel

L'attributo **rel** può essere valorizzato attraverso una serie di keyword che definiscono il tipo di risorsa che si vuole includere attraverso *link*. Ne vengono proposte alcune tra le più usate...

| Attributo  | Descrizione                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------|
| alternate  | Fornisce un link ad una versione alternativa del documento, come una versione stampabile, tradotta, etc... |
| author     | Link ad una pagina biografica dell'autore del documento                                                    |
| help       | Un link ad un file di aiuto                                                                                |
| icon       | Un'icona da usare per il documento, da mostrare sulla barra di navigazione del browser                     |
| next       | Link all'articolo successivo                                                                               |
| stylesheet | Un foglio di stile CSS                                                                                     |



# Risorse esterne: JavaScript

Il tag delegato all'inclusione di codice JavaScript all'interno di pagine web è **script**, che permette sia l'utilizzo di risorse *outline* che di codice *inline*. Le differenze:

- ▶ Il codice *inline*, così come per i CSS, viene incluso tra il tag di apertura e di chiusura
- ▶ Il sorgente esterno viene referenziato attraverso l'attributo *src*

Gli script possono essere eseguiti in diverse modalità:

- ▶ Se *async*=“*async*” il codice viene eseguito mentre il codice del documento è in fase di *parsing*
- ▶ Se non è definito *async* e l'attributo *defer* =“*defer*” il codice viene eseguito al termine del *parse* della pagina
- ▶ In assenza degli attributi *async* e *defer* lo script viene eseguito nel momento in cui il parser incontra il tag *script*



# Il tag script: attributi

| HTML5 | Attributo | Descrizione                                                                                                                          |
|-------|-----------|--------------------------------------------------------------------------------------------------------------------------------------|
| Si    | async     | Esegue lo script in modalità asincrona. Valido solo per <b>risorse esterne</b> .                                                     |
| No    | charset   | Codifica dei caratteri usata nello script. Valido solo per <b>risorse esterne</b> .                                                  |
| No    | defer     | Esegue lo script al termine del parse della pagina                                                                                   |
| No    | src       | URL della risorsa esterna                                                                                                            |
| No    | type      | Il media type dello script <ul style="list-style-type: none"><li>• application/javascript</li><li>• application/ecmascript</li></ul> |

**HTML**



# 11. Meta

# Meta

I *metadata* rappresentano delle informazioni di vario genere riguardo il documento HTML che **non** vengono mostrate direttamente bensì elaborate da *browser* e *motori di ricerca*.



HTML5 introduce dei *meta* dedicati al controllo del *viewport* ovvero la porzione di pagina visible all'utente.



# Meta

I *metadata* devono rispettare alcune regole ben definite:

- ▶ Il tag **meta** è sempre collocato all' interno dell'elemento **head**
- ▶ I metadata sono erogati in forma chiave/valore
- ▶ L'attributo *content* **deve** essere definito se è presente anche l'attributo *name* o *http-equiv*. In alternativa, l'attributo *content* **non** può essere definito



# Meta: viewport

Per **viewport** si intende la porzione di schermo visibile all'utente, variabile in base al **device** utilizzato per visualizzare il documento. Tutte le pagine web dovrebbero includere il meta tag seguente:

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

Grazie ai *meta viewport* il browser riceve istruzioni chiare su come controllare le dimensioni del documento ed il relativo *scaling*.

- ▶ **width=device-width** imposta la larghezza del documento a quella dello schermo del device
- ▶ **initial-scale=1.0** imposta lo zoom iniziale della pagina a quello di default



# Meta: viewport



**Senza** meta viewport



**Con** meta viewport

© Andrea Tantimonaco @ linkedin.com/in/andrea-tantimonaco - La riproduzione e/o l'utilizzo non autorizzato è severamente vietato



# Il tag meta: attributi

| HTML5 | Attributo  | Descrizione                                                                                                                                                                                                                                                        |
|-------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Si    | charset    | Codifica dei caratteri per il documento HTML                                                                                                                                                                                                                       |
| No    | content    | Valore associato alla keyword specificata attraverso <i>name</i> o <i>http-equiv</i>                                                                                                                                                                               |
| No    | http-equiv | Informazioni sull'HTTP Header <ul style="list-style-type: none"><li>• content-type</li><li>• default-style</li><li>• refresh</li></ul>                                                                                                                             |
| No    | name       | Specifica il nome del meta tag dando un senso al valore definito nell'attributo <i>content</i> <ul style="list-style-type: none"><li>• application-name</li><li>• author</li><li>• description</li><li>• generator</li><li>• keywords</li><li>• viewport</li></ul> |



# Meta: attributo http-equiv

**Http-equiv** permette di simulare un Header HTTP in risposta. Storicamente veniva utilizzato, tra l'altro, per impostare la codifica dei caratteri usata nella pagina web:

- ▶ **HTML4.01** - <meta http-equiv="content-type" content="text/html; charset=UTF-8">
- ▶ **HTML5** - <meta charset="UTF-8">

Come può essere valorizzato l'attributo http-equiv?

content-type

- Specifica la codifica dei caratteri nel documento – ad es. text/html; charset=UTF-8

default-style

- Definisce il foglio di stile CSS di default. Il content deve corrispondere al title specificato nel tag link

refresh

- Intervallo di tempo nel quale il documento esegue un refresh automatico della pagina



# Meta: attributo name

**Name** stabilisce il tipo di *metadata* che si vuole descrivere attraverso l'attributo *content*.

application-name

- Il nome dell'applicazione che il documento rappresenta

author

- Il nome dell'autore del documento

description

- Una breve descrizione del contenuto della pagina web da mostrare nei risultati dei motori di ricerca

generator

- Nome del software WYSIWYG con il quale è stato generato il documento

keywords

- Un elenco di keywords che meglio identificano il contenuto del documento, utili ai motori di ricerca

viewport

- Controlla la parte visible della pagina da parte dell'utente