

BREVE GUIDA AL LINGUAGGIO



Materiale della prof.ssa Raffaella Costantino
Rivisto ed ampliato dal prof. Rio Chierago

Versione 5.0 del 12/12/2024

Cos'è IL LINGUAGGIO HTML?

"Hyper Text Markup Language" ovvero "Linguaggio a marcatori per ipertesti"

È un linguaggio usato per descrivere **documenti ipertestuali** disponibili nel Web

È un **linguaggio di marcatura del testo** che descrive il contenuto e il formato di una pagina Web ...

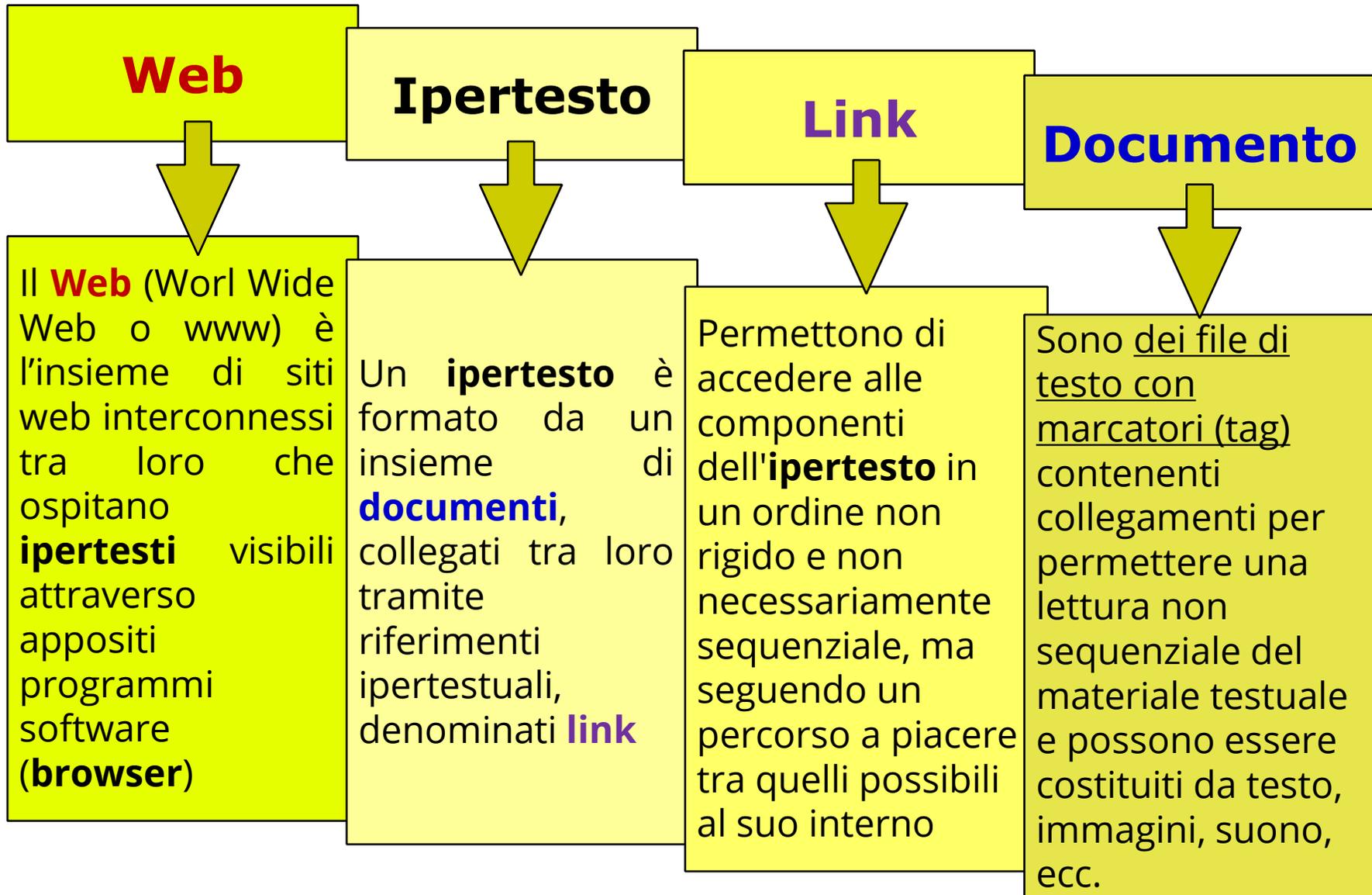
... e per fare ciò usa un insieme di parole chiave denominati **tag** o **marcatori**



ATTENZIONE:
Non è un linguaggio di programmazione



Impariamo un po' di termini



Caratteristiche principali

Ogni documento ipertestuale scritto in HTML deve essere contenuto in un file la cui estensione di solito è **.html** oppure **.htm**

Ogni **elemento** del documento (testo, immagine, tabella, elenco, link, ecc.) viene contrassegnato (marcato) da un opportuno **tag**

I tag verranno interpretati dal **browser** (es. **Firefox, Chrome, Edge, Opera**, ecc.) in modo da visualizzare l'effetto finale desiderato

Editor documenti ipertestuali

Per scrivere una pagina in HTML, è necessario utilizzare un **software** che permetta di andare effettivamente poi a salvare il codice scritto nel formato di pagina giusto (che convenientemente si chiama anch'esso HTML).

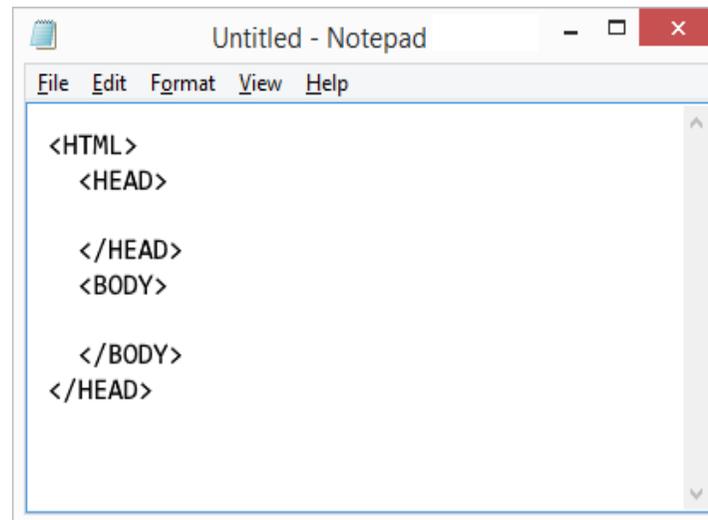
HTML è uno standard aperto e gratuito e non serve spendere soldi per iniziare a costruire pagine, anzi sul vostro computer avete già preinstallato un programma che vi consente di iniziare a familiarizzare con il codice

Su **UNIX**

```
[root@localhost gfg_prac]# vi geeksforgeeks
~
~
~
~
~
~
~
```

Editor **vi**

Su **WINDOWS**



```
Untitled - Notepad
File Edit Format View Help
<HTML>
  <HEAD>

  </HEAD>
  <BODY>

  </BODY>
</HEAD>
```

Editor **Blocco note**

Su **MAC**



```
Hello world
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/
html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta http-equiv="Content-Style-Type" content="text/css">
  <title></title>
  <meta name="Generator" content="Cocoa HTML Writer">
  <meta name="CocoaVersion" content="1539.1">
  <style type="text/css">
    p.p1 {margin: 0.0px 0.0px 0.0px 0.0px; font: 13.0px Courier}
    span.s1 {font-kerning: none}
  </style>
</head>
<body>
  <p class="p1"><span class="s1">&lt;HTML&gt;</span></p>
  <p class="p1"><span class="s1"> &lt;HEAD&gt;</span></p>
  <p class="p1"><span class="s1"> &lt;TITLE&gt;</span></p>
  <p class="p1"><span class="s1"> Hello world<span class="Apple-
```

Editor **TextEdit**

N.B. Questi non sono software comodi per scrivere pagine HTML perché la sintassi non viene evidenziata e in generale l'interfaccia è un po' troppo semplice.

Meglio utilizzare (in ambiente WINDOWS) l'editor **NOTEPAD++** scaricabile [qui](#)

Tag

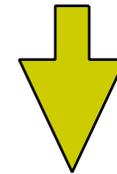
Gli elementi sono, di solito, racchiusi tra due **tag**, uno di apertura e uno di chiusura.

N.B. Alcuni tag non hanno la chiusura.

Il **tag di apertura** è costituito da una sequenza di caratteri racchiusa tra parentesi angolari: **<tag>** Quello di chiusura è identico a quello di apertura ma i caratteri vengono preceduti dallo **slash: </tag>**

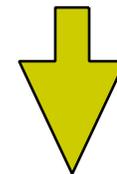
Ad esempio:

****grassetto****



grassetto

<I>italico**</I>**



italico

N.B. sebbene il linguaggio HTML non sia case sensitive si consiglia vivamente di scrivere i tag (di apertura e di chiusura) in MAIUSCOLO e di usare opportunamente l'INDENTAZIONE

Inizio e Fine

Un documento HTML comincia con il tag **<HTML>** e finisce con il tag **</HTML>**; tutte le informazioni contenute nel documento devono essere indicate tra questi due tag.

```
<HTML>  
  . . .  
  . . .  
  . . .  
</HTML>
```

Sezioni di una pagina HTML

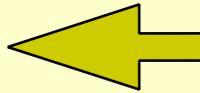
All'interno dei tag **<HTML>** ... **</HTML>** la sintassi HTML richiede due sezioni

<HTML>

<HEAD>

...

</HEAD>

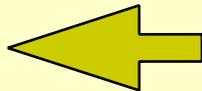


Nella sezione **HEAD**, delimitata dai tag **<HEAD>** e **</HEAD>**, vengono inserite le informazioni generali riguardanti l'intero documento e di solito non vengono visualizzate dal browser

<BODY>

...

</BODY>



All'interno della sezione **BODY**, delimitata dai tag **<BODY>** e **</BODY>**, sono indicati tutti gli elementi effettivamente visualizzate nel documento

</HTML>

Un primo esempio

```
<HTML>
  <HEAD>
    <TITLE> La mia pagina HTML </TITLE>
  </HEAD>
  <BODY>
    <B>La mia pagina HTML</B><BR>
    Io sono <I>Raffaella Costantino</I>
    e mi fa piacere conoscervi :- )
  </BODY>
</HTML>
```

Analizziamo e proviamo ad usare i nuovi **<tag>** presenti in questo esempio

Commenti 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.

Le **funzioni più comuni** svolte dai commenti in HTML sono:

- **Scrivere annotazioni** generiche al markup;
- **Rendere inattive porzioni di codice** in fase di debug o per effettuare dei test;
- **Segnalare la chiusura di blocchi di codice** per evitare confusione.

Passiamo subito alla pratica. Per aggiungere un commento la sintassi è piuttosto semplice, si racchiude il testo tra `<!--` e `-->`:

```
<!-- questo è un commento -->
```

Commenti HTML

Possiamo racchiudere in un commento anche **molte righe** e, spesso, i commenti sono utilizzati per disattivare alcune parti del markup: inserendo un tag in un commento lasciamo infatti che esso sia semplicemente ignorato dal browser.

```
<!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

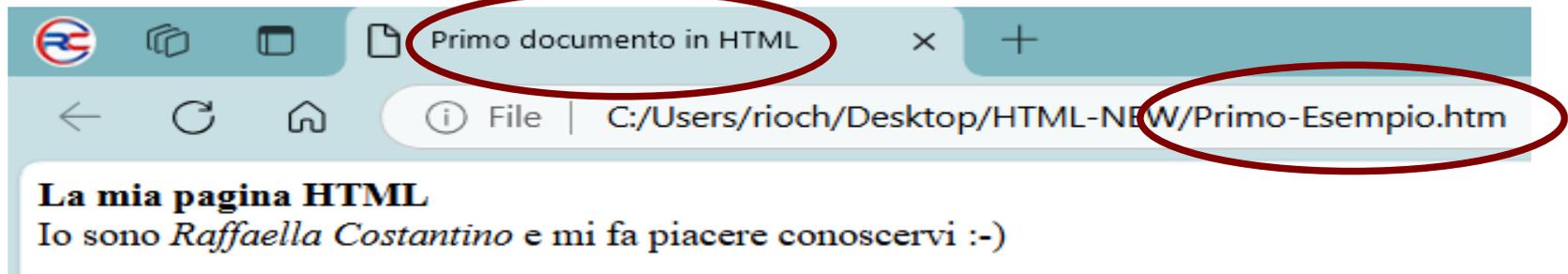
  <p>Non voglio vedere questo paragrafo</p>

  -->
</body>
</html>
```

<TITLE> ... </TITLE>

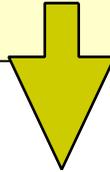
```
<TITLE>Primo documento in HTML</TITLE>
```

Il tag **<TITLE> ... </TITLE>** imposta il **titolo** che verrà visualizzato sulla barra del titolo del browser o sull'etichetta della scheda



<BODY> ... </BODY>

```
...  
<BODY>  
  <B>La mia pagina HTML</B><BR>  
  Io sono <I>Raffaella Costantino</I>  
  e mi fa piacere conoscervi :-)  
</BODY>  
...
```



Nel tag **<BODY> ... </BODY>** vengono inseriti tutti i contenuti di una pagina quali **titoli, paragrafi, tabelle, link, immagini, video** e molto altro ancora. Al tag **<BODY>** appartiene la sola funzione di indicare i contenuti della pagina Web mentre l'aspetto grafico viene gestito dai **fogli di stile CSS**

Attributi del tag <BODY

bgcolor – viene utilizzato per definire un colore per lo sfondo della pagina

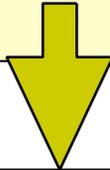
background – viene utilizzato per assegnare un'immagine di sfondo (antagonista di bgcolor)

text – viene utilizzato per definire un colore per il testo

link – viene utilizzato per definire un colore per i link

alink – viene utilizzato per definire un colore per i link attivi

vlink – viene utilizzato per definire un colore per i link già visitati



```
<BODY bgcolor="#7FFFD4"  
background="/images/sfondo.jpg"  
text="red"  
link="#0000FF"  
alink="#FF6600"  
vlink="#990099" >
```

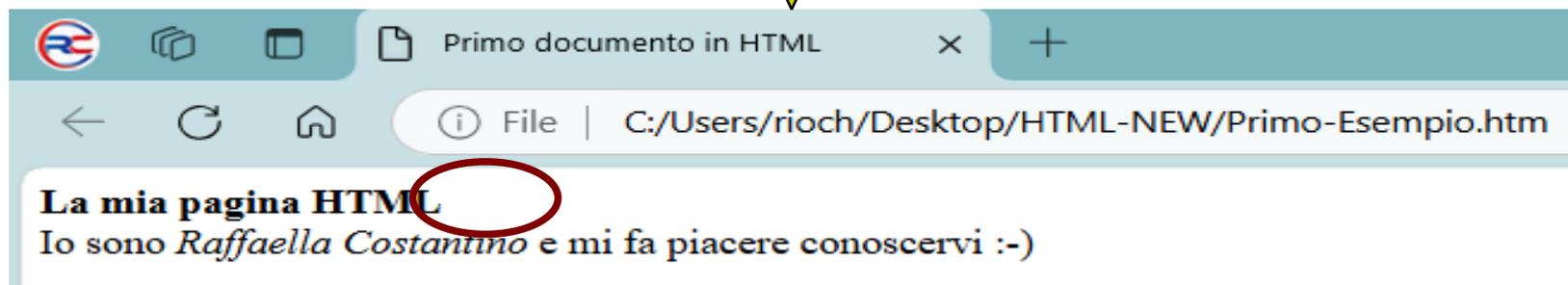
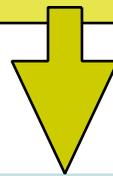
N.B. Con l'avvento di HTML 5 tutti questi attributi sono stati classificati come deprecati.

Nella nuova filosofia del linguaggio, voluta dai vertici del W3C, il markup HTML deve servire unicamente per definire gli aspetti strutturali del documento, mentre il suo aspetto grafico deve essere delegato ai CSS.

Elenco colori HTML visibile sul sito <https://toolset.mrw.it/html/colori-del-web.html>


```
<B>La mia pagina HTML</B><BR>  
Io sono <I>Raffaella Costantino</I>  
e mi fa piacere conoscervi :-)
```

Il tag **
** permette di **andare a capo**. E' uno dei tag che non prevede chiusura e per questo viene detto tag aperto.



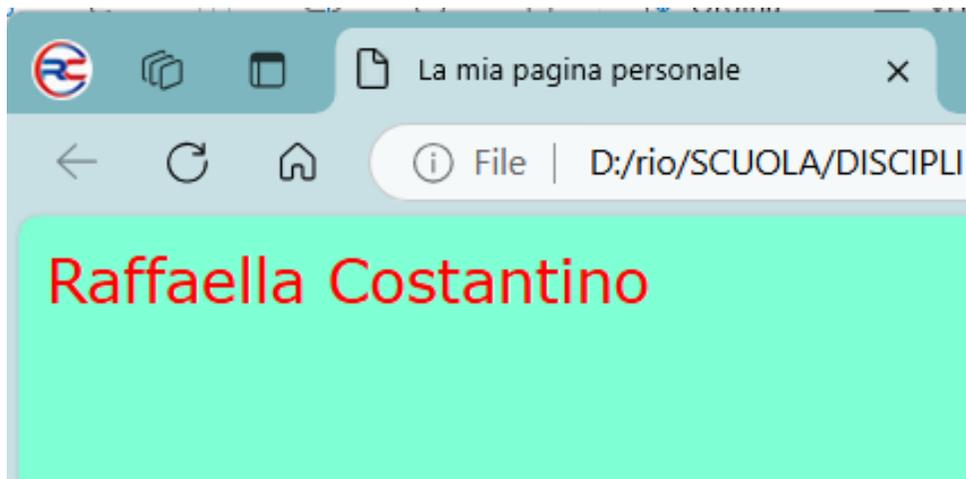
Elementi di formattazione HTML

Gli elementi di formattazione sono stati progettati per visualizzare tipi di testo speciali:

<code> ... </code>	- Testo in grassetto
<code> ... </code>	- Testo importante
<code><I> ... </I></code>	- Testo in corsivo
<code><U> ... </U></code>	- Testo sottolineato
<code> ... </code>	- Testo enfaticizzato
<code><MARK> ... </MARK></code>	- Testo contrassegnato
<code><SMALL> ... </SMALL></code>	- Testo più piccolo
<code><SUB> ... </SUB></code>	- Testo in pedice
<code><SUP> ... </SUP></code>	- Testo in apice

Un secondo esempio

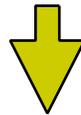
```
<HTML>
  <HEAD>
    <TITLE>La mia pagina personale</TITLE>
  </HEAD>
  <BODY bgcolor = "#7FFFD4" >
    <FONT face = "Verdana" color = "red" size = "3"
      Io sono <I>Raffaella Costantino</I></FONT><BR>
  </BODY>
</HTML>
```



Analizziamo e proviamo ad usare il nuovo **<tag>** presente in questo esempio

 ...

Nel **linguaggio HTML** il tag **** definisce principalmente le dimensioni, il colore e il tipo di carattere di un testo. È un tag deprecato, ormai sostituito dai fogli di stile CSS, ma è ancora perfettamente funzionante



Tra gli attributi principali di questo tag:

face : indica il tipo di font di carattere da utilizzare (es. *Arial, Verdana, Times New Roman, ecc...*eventualmente sottoforma di lista separata da virgola).

*N.B. La **face di default** è "Times new Roman".*

color : Indica il colore da applicare al testo (da definire con **una serie di 6 caratteri alfanumerici preceduti da cancelletto #** oppure con **nome in inglese...**a tal proposito consultare il seguente [link](#)).

*N.B. Il **color di default** è "**black**" ossia "#000000"*

size : Definisce la grandezza dei caratteri (da esprimere con **un numero assoluto intero** da 1 a 7)

*N.B. la **size di default** è il "2"*

 ...

Per visualizzare un testo in **grassetto** di **colore rosso** posso scrivere:



```
<FONT color = "#FF0000"><B>Hello World!</B></FONT>
```

Il risultato sullo schermo dell'utente è il seguente

Hello World!

Per visualizzare un testo in *corsivo* sottolineato in **Verdana** di **colore blu** posso scrivere:



```
<FONT face = Verdana color = "#0000FF">  
  <I><U>Hello World!</U></B>  
</FONT>
```

Il risultato sullo schermo dell'utente è il seguente

Hello World!

 ...

Per visualizzare un testo variando tutte le opzioni di **size** posso scrivere:



```
<FONT size = "1">Hello World! </FONT><BR>  
<FONT size = "2">Hello World! </FONT><BR>  
<FONT size = "3">Hello World! </FONT><BR>  
<FONT size = "4">Hello World! </FONT><BR>  
<FONT size = "5">Hello World! </FONT><BR>  
<FONT size = "6">Hello World! </FONT><BR>  
<FONT size = "7">Hello World! </FONT>
```

Il risultato sullo schermo dell'utente è il seguente

Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!

<A> ...

I **link** sono formati essenzialmente da **due** componenti:

- Il **contenuto "attivo"** ovvero quella parte della pagina che può essere «*cliccata*» per spostarsi ad un'altra pagina (tipicamente un testo o un'immagine). Essendo la parte fondamentale dell'interazione, l'utente deve essere sempre in grado di distinguerlo dal resto del contenuto della pagina.

- **La risorsa** verso cui il collegamento punta.

Come abbiamo già detto può essere o un'altra pagina (sullo stesso sito o su un sito diverso) (**link esterni**), oppure un collegamento "interno" a un punto della pagina in cui si trova (**link interni**).

Il tag <A> ... crea un **link** e rende il documento un ipertesto. All'attributo obbligatorio **href** può essere assegnato l'indirizzo di una pagina "esterna" (**link esterni**) o una sezione "interna" della medesima pagina (**link interni**) a cui "saltare".

N.B. **A** sta per *anchor* in inglese

Attributi del tag <A>

href – serve per specificare l'indirizzo (URL) della pagina a cui "saltare"

title – serve per specificare un testo esplicativo per l'elemento a cui l'attributo è riferito

Questa spiegazione aggiuntiva favorisce l'accessibilità del sito anche ai disabili.

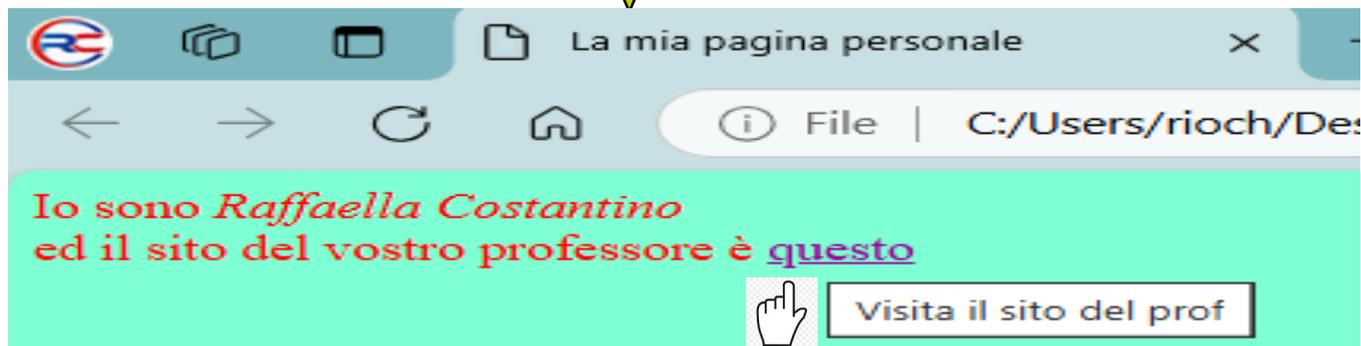
target – serve per specificare in quale finestra la pagina linkata deve essere aperta

Valori possibili sono:

_blank : la pagina linkata sarà aperta in una pagina diversa da quella che ospita il link

_self : la pagina linkata sarà aperta nella stessa pagina che ospita il link

```
<A title = "Visita il sito del prof"  
href = "http://www.riochierego.it"  
target = "_blank" > questo  
</A>
```



<A> ...

I link esterni

La navigazione tra le pagine web e i download avvengono secondo un protocollo definito **HTTP** (Hypertext Transfer Protocol).

Esistono altri **protocolli associati ad altre funzioni** come ad esempio l'FTP definito per trasferire files tra cartelle locali e remote.

Un protocollo specifico è denominato **mailto**. Serve ad attivare un programma di posta elettronica.

Questo protocollo è utilizzabile nei **link** con la seguente modalità:

```
<A href="mailto:pippo@gmail.com">Invia una e-mail a Pippo</A>
```

Il click sul collegamento aprirà il programma di posta elettronica predefinito dell'utente.

<A> ...

I link interni

È possibile anche creare un **indice interno** al documento, utilizzando link «interni». Per creare un link interno si utilizza l'attributo **name** in abbinata ad **href**.

*(N.B. l'attributo **name** del linguaggio HTML ha molti campi di applicazione ed è usato in molti suoi elementi come i FORM ma anche come selettore nei linguaggi di programmazione JavaScript e PHP)*

L'attributo **name** definisce il nome del punto da contrassegnare all'interno di una stessa pagina. *(Si noti che questa tecnica ha un senso in pagine relativamente lunghe che comportano un abbondante scroll).*

```
<A name = "primo">Lorem ipsum factum ..eccetera..</A>
```

La mancanza dell'attributo **href** rende questi contrassegni «**invisibili**» all'utente. Tuttavia essi servono come riferimento per lo «**scroll**» **della pagina** che avviene cliccando un secondo elemento così definito:

```
<A href = "#primo">Vai al primo paragrafo </A>
```

Il cancelletto indica che il collegamento deve cercare un contrassegno chiamato «**primo**» all'interno della pagina stessa.

Per creare un indice interno alla pagina si **procede dunque in due fasi distinte**:

1. creazione del contrassegno a cui puntare (attraverso l'attributo **name**)
2. creazione del link con relativo riferimento interno (con l'attributo **href** con **#**)

NB: è fondamentale non confondere le due fasi.

Immagini: formati consentiti

Le immagini ovviamente possono essere inserite non solo come sfondo di una pagina web, ma anche come contenuto.

I formati ammessi nel Web sono sostanzialmente tre:

- **GIF** (Graphic Interchange Format):

le GIF sono immagini con non più di **256 colori** (dunque con colori piatti e senza sfumature), come grafici o icone. Supporta la **trasparenza** anche se non con le sfumature possibili col il PNG.

- **JPG** (Joint Photographic experts Group)

supporta i **16M di colori**, è l'acronimo del gruppo di ricerca che ha ideato questo formato, idoneo per le immagini di qualità fotografica. **Non supporta le trasparenze**. Permette livelli di **compressione** regolabili in funzione della qualità richiesta.

- **PNG** (Portable Network Graphic)

un tipo di immagine introdotto più recentemente, elaborato dal W3C per risolvere i problemi di copyright del formato **GIF** (che è appunto proprietario); anch'esso supporta i **16M di colori** ed è ormai letto da tutti i browser e offre alcune caratteristiche che gli altri formati non hanno in primis il supporto al canale alfa che lo rende adatto a creare **trasparenze** molto morbide. Il livello di compressione è in genere intermedio tra **GIF** e **JPG**.

NB: qualsiasi altro formato non è supportato (psd, tiff, ecc) e impedisce la visualizzazione dell'immagine nella pagina oppure attiva la funzione di download del file.

Il tag e le immagini

Il **tag** per inserire una immagine è **** la cui sintassi di base è la seguente:

```
<IMG src = "<nome_immagine>" >
```

L'attributo essenziale è **src**, che significa **«source»**, ovvero origine; tramite **src** si specifica **l'indirizzo web** da cui recuperare l'immagine da visualizzare.

Le immagini sono il primo elemento **«esterno»** alla pagina che trattiamo.

Occorre comprendere come **l'accesso ai contenuti di un sito sia per lo più un processo ASINCRONO**, ovvero regolato da flussi di informazioni che hanno tempi correlati ma non sincronizzati.

Un esempio può chiarire meglio il concetto: Immaginiamo una pagina web con due immagini al suo interno. Il browser per prima cosa scaricherà il file HTML e lo interpreterà. Dalla lettura del codice capirà che deve scaricare altri 2 files (per esempio due file **jpg**), inizierà il download, ma nel frattempo inizierà il rendering del testo. Il **rendering completo** (e l'eventuale riassetto del layout) avverrà nel momento in cui entrambi i JPG saranno nella cache.

N.B. Per questo motivo se le immagini sono delle grafiche necessarie al layout della pagina, occorre fare attenzione a renderle il più «leggere» possibile per velocizzare l'impaginazione ed evitare fastidiosi effetti di riassetto/riallineamento delle varie parti della pagina.

Come abbiamo già detto il tag **** permette di **inserire un'immagine** all'interno di una pagina web.

Tra gli **attributi** principali di questo **tag**:

src : (*obbligatorio*) viene assegnato il nome, con eventuale percorso (**path**), del file contenente l'immagine da visualizzare.

alt : contiene una descrizione testuale dell'immagine da visualizzare necessaria al fine di garantire la piena accessibilità del documento agli utenti con *handicap* visivi (*tooltip*).

width : specifica l'altezza in pixel dell'immagine

height : specifica la larghezza in pixel dell'immagine

*N.B se **width** e **height** non vengono specificati l'immagine sarà mostrata nelle sue dimensioni originali*

border : è lo spessore del bordo in pixel. Se lo spessore è impostato a **0** non viene applicato alcun bordo all'immagine, mentre se viene omesso il bordo è di **1 pixel** e, nel caso l'immagine sia in un tag **<A>** assume gli stili dei link.

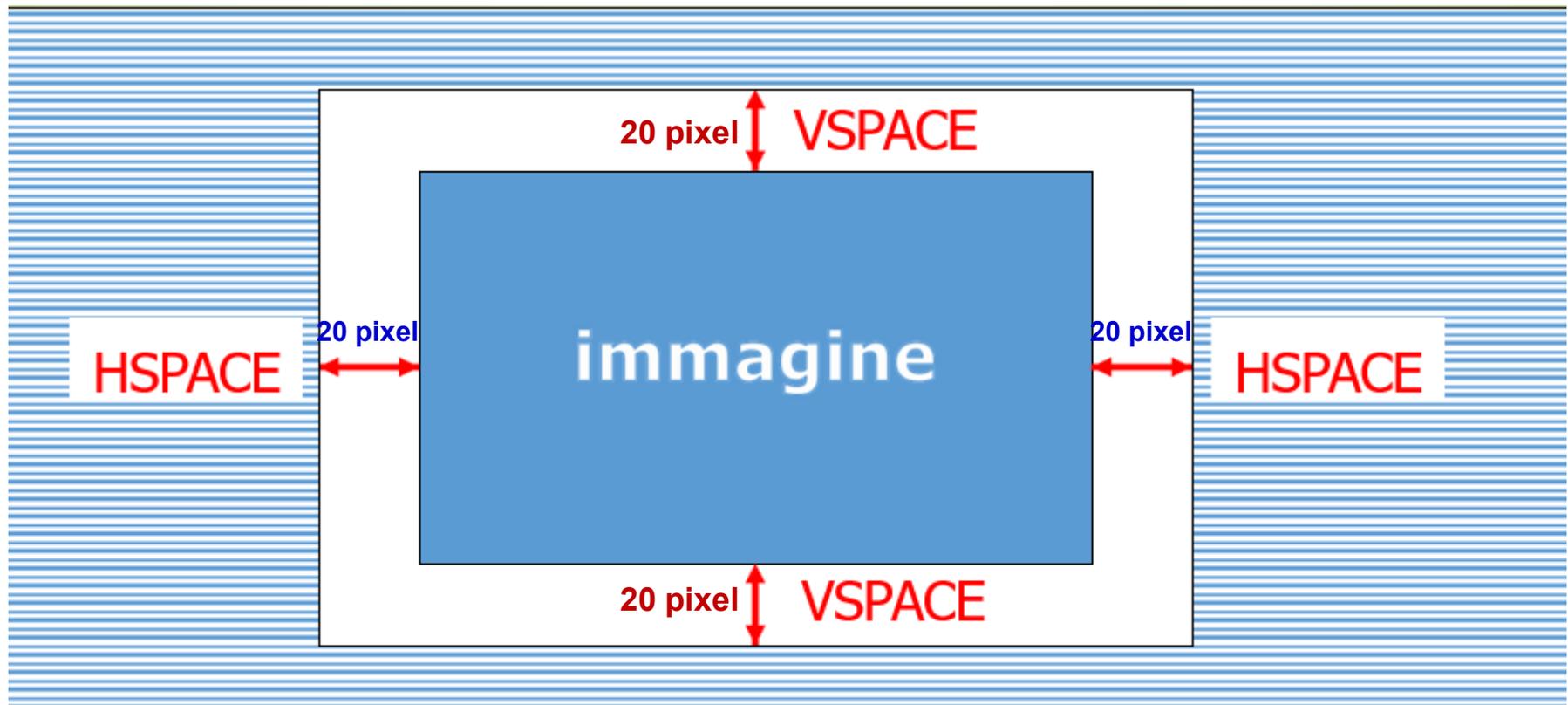
hspace : regola **orizzontalmente** lo spazio (in pixel) tra immagine e testo

vspace : regola **verticalmente** lo spazio (in pixel) tra immagine e testo

align : consente l'allineamento dell'immagine rispetto al flusso del testo

EXTRA: funzionamento degli attributi **hspace** e **vspace**

```
<IMG src = "<immagine>" hspace = "20" vspace = "20" >
```



EXTRA: valori possibili dell'attributo **align**

```
<IMG src = "<nome_immagine>" align = "valore" >
```



Tag e tag <A>

Per **rendere un'immagine un collegamento ipertestuale** è sufficiente nidificarla all'interno di un tag <A> :

```
<A href = "http://www.iuav.it" >  
  <IMG src = "immagine.gif" border = "0" >  
</A>
```

Si noti l'attributo **border** a zero per evitare un antiestetico bordo attorno all'immagine.

Oppure anche:

```
<A href = "http://www.iuav.it" > Università Iuav di Venezia <BR>  
  <IMG src = "immagine.gif" border = "0" >  
</A>
```

dove abbiamo un link composito formato da **testo** (codice in rosso) e immagine a seguire (codice in blu), entrambi parte del collegamento ipertestuale.

Un terzo esempio

```
<HTML>
  <HEAD>
    <TITLE>La mia pagina personale</TITLE>
  </HEAD>
  <BODY bgcolor = "#7FFFD4" text = "red" >
    Io sono <I>Raffaella Costantino</I><BR>
    <IMG src = "images/io.jpg"
      alt = "Questa sono io!"><BR>
    ed il sito del vostro professore è
    <A href = "http://www.riochierego.it"
      target = "_blank" >questo</A>
  </BODY>
</HTML>
```

Il risultato alla fine sarà ...



... questo

A screenshot of a web browser window. The browser has a single tab titled "La mia pagina personale". The address bar shows the file path "C:/Users/rioch/Desktop/HTML-NEW/Terzo-Esempio.htm". The page content is on a light green background and includes the text "Io sono *Raffaella Costantino*" in red, an illustration of a female teacher with a chalkboard, and the text "ed il sito del vostro professore è [questo](#)" in red, where "questo" is underlined and blue.

Io sono *Raffaella Costantino*

An illustration of a female teacher with brown hair, wearing a blue blazer and a black skirt, holding a blue folder and pointing with a stick at a green chalkboard. The chalkboard has some faint white lines on it.

ed il sito del vostro professore è [questo](#)

Ora provate voi ...

Personalizzate la vostra pagina personale cercando i **tag** e gli **attributi** adatti per svolgere le seguenti operazioni:

1. Dare un vostro titolo alla vostra pagina iniziale
2. Descrivervi brevemente usando grassetto ed italico.
3. Creare un link ad una pagina nuova contenente una vostra immagine.
4. Nella pagina nuova inserite un link per ritornare alla vostra pagina personale

Creiamo un sito comune #1

- 1) Create una cartella **mioSito** e all'interno create la nuova cartella **images**.
- 2) Ponete la vostra immagine nella cartella **images**.
- 3) Editate la vostra pagina personale chiamandola **<cognome>.htm** contenente una breve presentazione e la vostra immagine.
- 4) Spedite ad ogni vostro compagno di classe la **pagina** così creata e **l'immagine** che avete usato.
- 5) Salvate tutte le pagine ricevute nella cartella **mioSito** e tutte le immagini nella cartella **images**.
- 6) Create ora la pagina **index.htm** nella cartella **mioSito**, elencando i nomi dei compagni come [link](#) alle relative pagine ed in ciascuna di esse aggiungete il [link](#) per ritornare alla pagina indice di partenza.

<P> ... </P>

Il tag **<P>** ... **</P>** permette di creare un **paragrafo**

Il **paragrafo** è utilizzato soprattutto per agevolare la lettura di un testo tramite il web browser.

L'uso dei **paragrafi** è molto utile perché suddividono il testo in blocchi, separati tra loro da un'**interlinea bianca**, ognuno dei quali contiene un particolare aspetto del discorso.

I **paragrafi** rendono più facile la lettura dei contenuti in una pagina web perché la lettura di un testo online sul monitor del PC è decisamente molto più stancante rispetto a quella su un libro.

Attributo principale del tag **<P>**:

align – indica come deve essere allineato il testo nella pagina

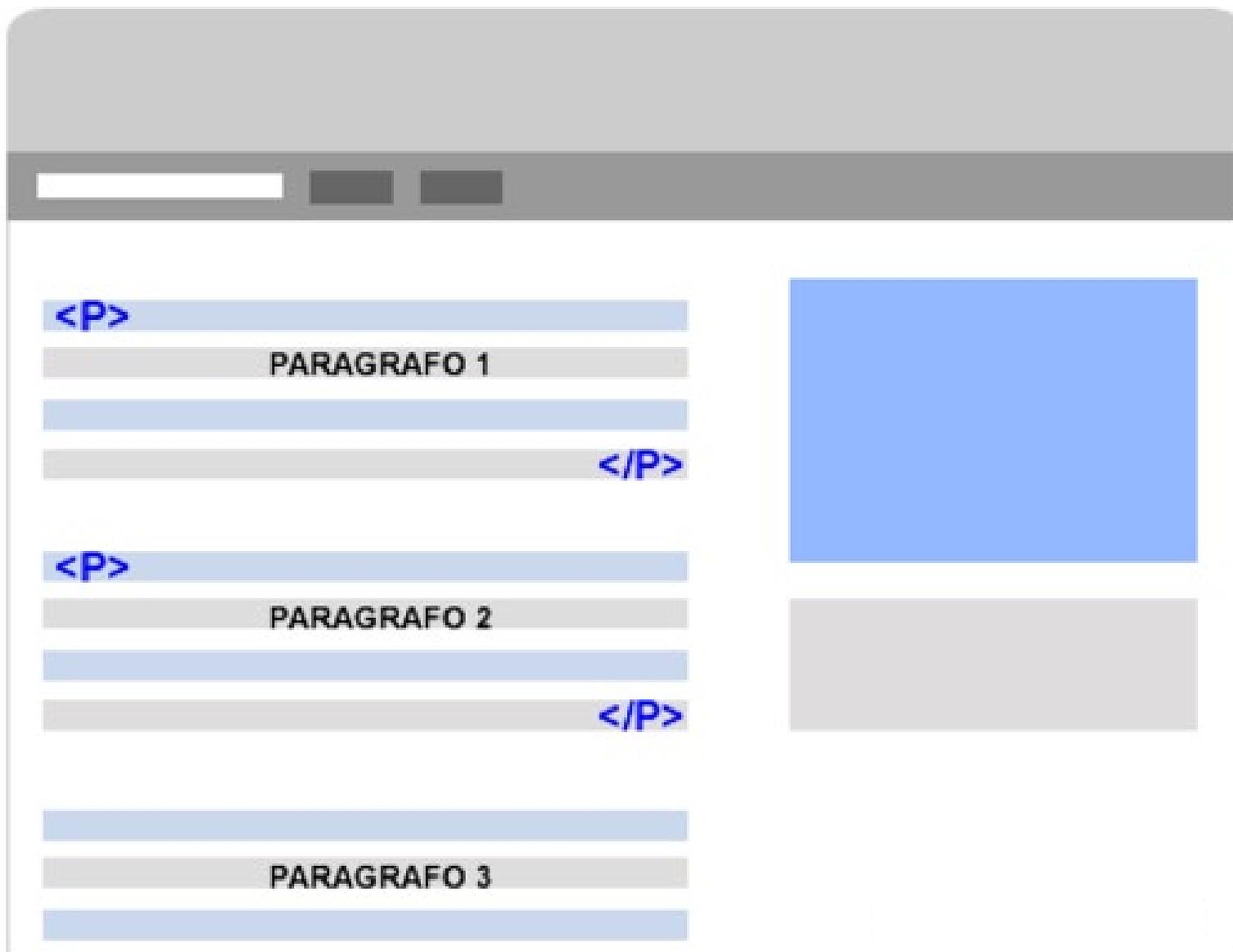
Valori possibili sono:

left : testo del paragrafo allineato a sinistra (default)

center : testo del paragrafo allineato al centro

right : testo del paragrafo allineato a destra

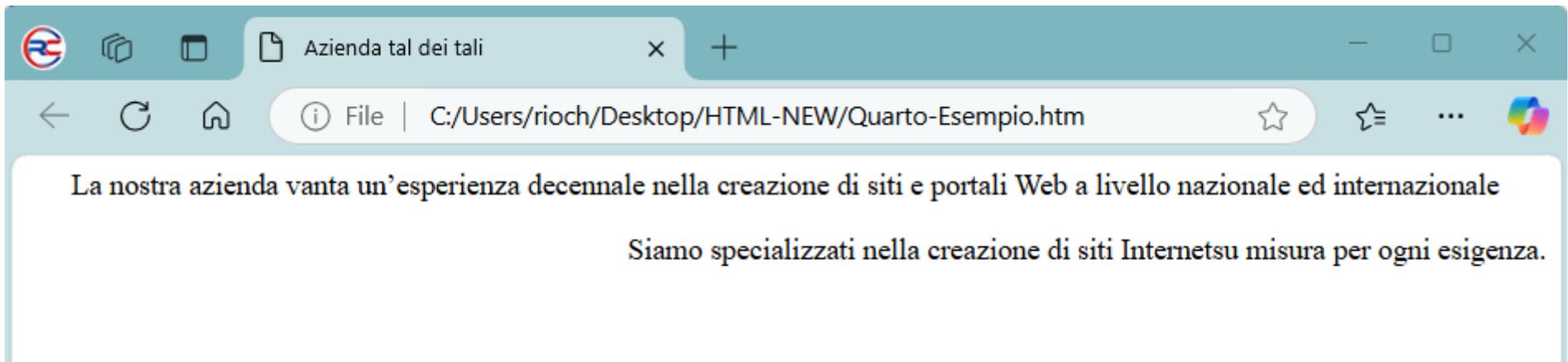
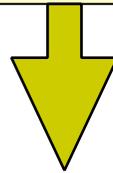
<P> ... </P>



<P> ... </P>

Esempio pratico

```
<P align = "center" >La nostra azienda vanta  
un'esperienza decennale nella creazione di siti e portali  
Web a livello nazionale ed internazionale</P>  
<P align = "right" >Siamo specializzati nella creazione  
di siti Internet su misura per ogni esigenza.</P>
```



Meglio `<P>` o `
`?

In alternativa al **paragrafo** abbiamo già visto che è possibile separare i contenuti testuali di una pagina web tramite il tag `
` di **interruzione di riga**.

Tuttavia, per i paragrafi è meglio usare il tag `<P>` perché dà al contenuto un'organizzazione più logica.

Inoltre, i tag `<P>` gestiscono meglio la formattazione del testo con **i fogli di stile a cascata CSS** (**Cascading Style Sheet**).



```
<P>Primo blocco testuale</P>  
<P>Secondo blocco testuale</P>
```

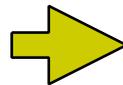
```
Primo blocco testuale<BR>  
Secondo blocco testuale
```

`<Hn> ... </Hn>`

Il tag `<Hn> ... </Hn>` permette di creare un **titolo nella pagina**;
il numero *n* varia **1** (dimensione massima) a **6** (dimensione minima)



```
<H1>Esempio di Titolo 1</H1>  
<H2>Esempio di Titolo 2</H2>  
<H3>Esempio di Titolo 3</H3>  
<H4>Esempio di Titolo 4</H4>  
<H5>Esempio di Titolo 5</H5>  
<H6>Esempio di Titolo 6</H6>
```



<HR>

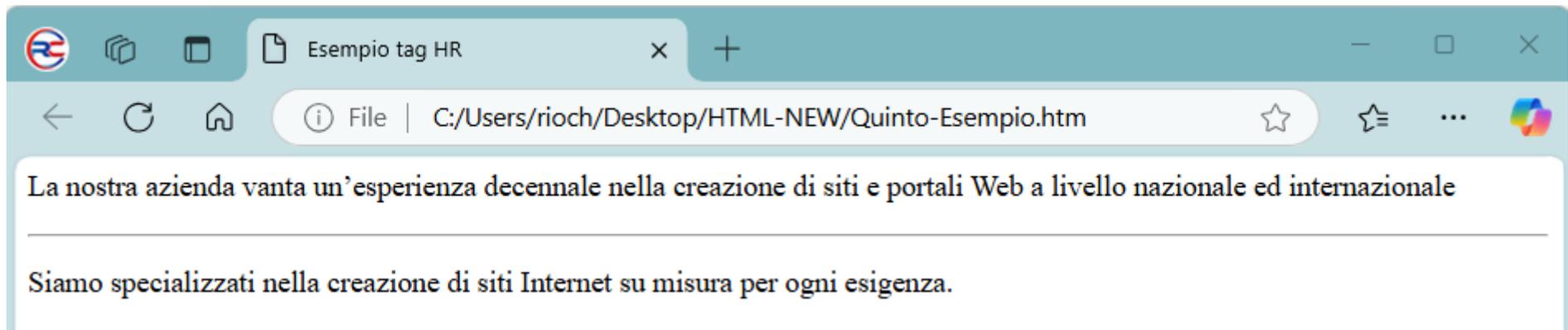
Il tag **<HR>** permette di inserire una **barra orizzontale**



```
<P>La nostra azienda vanta un'esperienza decennale nella  
creazione di siti e portali Web a livello nazionale ed  
internazionale.</P>
```

```
<HR>
```

```
<P>Siamo specializzati nella creazione di siti Internet su  
misura per ogni esigenza.</P>
```

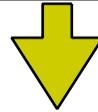


Elenchi puntati e numerati

Il tag ` ... ` permette di creare un **elenco puntato** (*unordered list*).

Il tag ` ... ` permette di creare un **elenco numerato** (*ordered list*).

Per entrambi il tag ` ... ` definisce un **elemento dell'elenco** (*list item*)



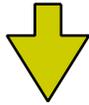
```
<UL type="circle">  
  <LI>banner</LI>  
  <LI>barra di navigazione</LI>  
</UL>
```



- o banner
- o barra di navigazione

Elenchi puntati e numerati

Il tag ` ... ` prevede l'attributo **type** per **variare la tipologia** dei punti elenchi e l'attributo **start** per **variare il valore di partenza dell'elenco** dei punti elenchi (N.B. solo di quelli numerici ossia con **type = "1"** che è il valore di default)



```
<OL>
  <LI>contenuto principale</LI>
  <LI>colonna di destra</LI>
</OL>
```



3. contenuto principale
4. colonna di destra



```
<OL type = "A">
  <LI>contenuto principale</LI>
  <LI>colonna di destra</LI>
</OL>
```



A. contenuto principale
B. colonna di destra

Elenchi puntati e numerati

Il tag ` ... ` prevede l'attributo **type** per **variare la tipologia del segno grafico** dei punti elenchi (il tipo di segno grafico di default utilizzato per individuare gli elementi dell'elenco dipende dal browser, ma di solito è un "pallino pieno")

```
<UL>
  <LI>contenuto principale</LI>
  <LI>colonna di destra</LI>
</UL>
```

- contenuto principale
- colonna di destra

```
<UL type = "square">
  <LI>contenuto principale</LI>
  <LI>colonna di destra</LI>
</UL>
```

- contenuto principale
- colonna di destra

Elenchi puntati e numerati

Valore attributo **type**
per tag **** ... ****

Valore dell'attributo type	Stile di enumerazione	
type="1" (è così di default)	numeri arabi	<pre><ol type="1"> primo secondo terzo </pre> <ol style="list-style-type: none"> 1. primo 2. secondo 3. terzo
type="a"	alfabeto minuscolo	<pre><ol type="a"> primo secondo terzo </pre> <ol style="list-style-type: none"> a. primo b. secondo c. terzo
type="A"	alfabeto maiuscolo	<pre><ol type="A"> primo secondo terzo </pre> <ol style="list-style-type: none"> A. primo B. secondo C. terzo
type="i"	numeri romani minuscoli	<pre><ol type="i"> primo secondo terzo </pre> <ol style="list-style-type: none"> i. primo ii. secondo iii. terzo
type="I"	numeri romani maiuscoli	<pre><ol type="I"> primo secondo terzo </pre> <ol style="list-style-type: none"> I. primo II. secondo III. terzo

Valore attributo **type**
per tag **** ... ****

Valore dell'attributo type	Stile di enumerazione	
type="disc" (è così di default)	visualizza un "pallino" pieno . È la visualizzazione di default	<pre><ul type="disc"> primo secondo terzo </pre> <ul style="list-style-type: none"> • primo • secondo • terzo
type="circle"	visualizza un cerchio vuoto al proprio interno	<pre><ul type="circle"> primo secondo terzo </pre> <ul style="list-style-type: none"> ○ primo ○ secondo ○ terzo
type="square"	Visualizza un quadrato pieno al proprio interno	<pre><ul type="square"> primo secondo terzo </pre> <ul style="list-style-type: none"> ▪ primo ▪ secondo ▪ terzo

NB: anche l'attributo type è **deprecato**, perché si tratta di una caratteristica che riguarda la formattazione, e dunque andrebbe effettuata utilizzando i fogli di stile CSS (possibilità di scegliere un'immagine (ad esempio una GIF) come segno distintivo per l'elenco puntato)

Elenchi puntati e numerati

Modalità alternativa per la determinazione della tipologia di punto elenco

attributo *style* proprietà due punti valore

```
<UL style = "list-style-type:circle">  
  <LI>banner</LI>  
  <LI>barra di navigazione</LI>  
</UL>
```



- o banner
- o barra di navigazione

EXTRA : l'attributo **style**

style è un attributo di **<tag>**, esattamente come **src**, **href**, **width** o altri.

Ciò significa che non può essere applicato da solo, ma deve sempre "appoggiarsi" a un **<tag>**, di cui costituisce una specifica.

La differenza è, come visto, che **style** può essere applicato indifferentemente a qualunque tag

La sintassi dell'attributo **style** è leggermente complessa ed è legata ai fogli di stile CSS (esattamente in modalità **in linea**)

attributo **style** proprietà due punti valore punto e virgola

```
<P style="color:red;text-align:center;font-family:Arial">  
    Paragrafo rosso, centrato e in Arial  
</P>
```

APPROFONDIMENTO: Come usare i fogli di stile CSS

https://www.w3schools.com/css/css_howto.asp

Per ottenere questo elenco ...

Schema sito Web

1. Intestazione

- banner
- barra di navigazione

2. Corpo del documento

- contenuto principale
- colonnadi destra

3. Piè di pagina

- copyright
- informazioni legali

... il codice HTML sarà il seguente

```
→ 1 <!doctype html>
→ 2 <html lang="it">
→ 3 <head>
4   <meta charset="utf-8" />
5   <title>GigaWeb Software</title>
6 </head>
7 <body>
8   <h1>Schema sito Web</h1>
9   <ol type="1">
10    <li>Intestazione
11      <ul style="list-style-type:circle">
12        <li>banner</li>
13        <li>barra di navigazione</li>
14      </ul>
15    </li>
16    <li>Corpo del documento
17      <ul style="list-style-type:disc">
18        <li>contenuto principale</li>
19        <li>colonnadi destra</li>
20      </ul>
21    </li>
→ 22    <li>Pi&egrave; di pagina
23      <ul style="list-style-type:square">
24        <li>copyright</li>
25        <li>informazioni legali</li>
26      </ul>
27    </li>
28  </ol>
29 </body>
30 </html>
```

Da questo codice HTML si ottiene...

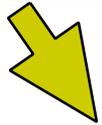
```
1 <!doctype html>
2 <html lang="it">
3 <head>
4   <meta charset="utf-8" />
5   <title>Lezioni HTML</title>
6 </head>
7 <body>
8   <h1>3<sup>a</sup> lezione HTML</h1>
9   <h3>Partecipanti</h3>
10  <hr />
11
12  <p>Elenco dei partecipanti</p>
13  <ol type="1">
14    <li>Pascal Blaise</li>
15    <li>Leibniz Gottfried Wihelm</li>
16    <li>Babbage Charles</li>
17    <li>Byron Augusta Ada</li>
18    <li>Hollerith Herman</li>
19    <li>Zuse Konrad</li>
20    <li>Aiken Howard</li>
21    <li><a href="turing/turing.html" target="_blank"
22      title="Visista il sito di A. Turing">Turing Alan</a></li>
23    <li>Wirth Niklaus</li>
24    <li>Jobs Steve</li>
25    <li>Wozniak Steve</li>
26  </ol>
27
28 </body>
29 </html>
```

Il risultato è ...



... questo altro elenco

... il seguente



3^a lezione HTML

Partecipanti

Elenco dei partecipanti

1. Pascal Blaise
2. Leibniz Gottfried Wihelm
3. Babbage Charles
4. Byron Augusta Ada
5. Hollerith Herman
6. Zuse Konrad
7. Aiken Howard
8. [Turing Alan](#)
9. Wirth Niklaus
10. Jobs Steve
11. Wozniak Steve

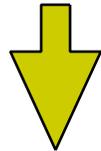
Creiamo un sito comune #2

- 1) Modificate la vostra pagina personale **<cognome>.htm** dopo aver cercato i **tag** e gli **attributi** segnalati con le frecce nell'esempio precedente (righe [1](#), [2](#) e [4](#)).
- 2) Cercate su Internet le **entità HTML** (riga [22](#)) e usatele nel vostro sito quando necessario.
- 3) Variate la pagina **index.htm** creando un [elenco](#) di link, scegliendo il tipo di elenco che preferite.
- 4) Una volta modificata la vostra pagina personale speditela nuovamente ai vostri compagni.

<!DOCTYPE>

La dichiarazione **<!DOCTYPE>** non è un tag HTML, è una **dichiarazione** per il browser che indica con quale **versione di HTML** è stata scritta la pagina.

Va posta sempre all'inizio della pagina fuori dal tag <HTML> e per **HTML5** è la seguente:



```
<!DOCTYPE HTML>
```

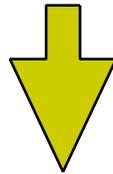
Il suo compito principale è quello di dichiarare il tipo di documento che stai presentando. In termini tecnici parliamo di **Document Type Definition** o **DTD** dove si chiarisce il codice usato nel documento HTML.

Inoltre in questa stringa puoi avere indicazioni sul fatto che il documento è pubblico o meno, sulla versione HTML supportata, sulla lingua del documento in questione.

Anche se la presenza è scontata, bisogna sempre verificare questo elemento.

L'attributo **lang**

L'attributo **lang** del tag **<HTML>** specifica la **lingua utilizzata nel documento**



```
<HTML lang = "it">
```

```
...
```

```
</HTML>
```

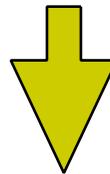
<META>

Il tag **<META>** fornisce **informazioni (o metadati) sul documento HTML.**

I **metadati** non vengono visualizzati ma sono compresi dal computer che sa interpretarli, dai motori di ricerca e da altri servizi Web.

Questo tag viene usato per fornire una descrizione della pagina, le keywords, l'autore del documento, l'ultima modifica, il **set di caratteri usato**, ecc.

E' un tag che va posto nella sezione **<HEAD>**

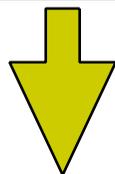


```
<HEAD>  
    <META charset = "utf-8">  
    ...  
</HEAD>
```

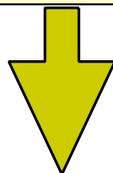
Entità HTML

Le **entità HTML** sono dei particolari **codici** che servono **per rappresentare caratteri speciali** che potrebbero avere problemi di visualizzazione su computer e sistemi operativi differenti.

Il codice per inserire una entità nella pagina è ***&entità;***



Piè di pagina



Piè di pagina

Entità HTML più comuni

SIMBOLO	ENTITA' HTML
à	à
á	á
è	è
é	é
ò	ò
ó	ó
ù	ù
ú	ú
ì	ì
í	í
<	<
>	>
“	"
’	´
°	º
&	&
€	€



Nota Bene
Per avere un
elenco più
completo
delle possibili
entità e della
loro codifica
in linguaggio
HTML, è
possibile
visitare
questo [URL](#)

Le Tabelle: i tag `<TABLE>`, `<TR>` e `<TD>`

Le **tabelle HTML** hanno avuto un'importanza molto elevata nella realizzazione dei siti web visto che sono state largamente **impiegate per realizzare i layout** delle pagine oltre che per organizzare i contenuti per righe e colonne.

Oggi **l'impaginazione di una pagina HTML mediante tabelle è sconsigliato e deprecato dal W3C** che, prima con i **fogli di stile**, poi con le **novità dell'HTML 5**, ha definito modalità più efficaci e versatili per suddividere le pagine nelle diverse sezioni di contenuto.

in ogni caso le **TABELLE** sono definite con il tag `<TABLE> ... </TABLE>`.

Una tabella è divisa in **RIGHE** tramite il tag `<TR> ... </TR>`, e ogni riga è suddivisa in **CELLE** con il tag `<TD> ... </TD>` (per cui le **colonne** sono di fatto una conseguenza di tale suddivisione)

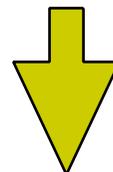
Ogni tag `<TR>` conterrà **tanti** tag `<TD>` quante sono le CELLE (colonne in senso improprio) della tabella.

I dati di ogni CELLA saranno contenuti nei singoli tag `<TD>`

NOTA BENE

TR sta per **TABLE ROW**

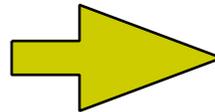
TD sta per **TABLE DATA CELLS**



Vediamo ora un esempio ...

Le Tabelle: i tag <TABLE>, <TR> e <TD>

```
<HTML>
  <HEAD>
    <TITLE>Esempio di tabella</TITLE>
  </HEAD>
  <BODY>
    <TABLE border = "1">
      <TR>
        <TD>Cognome</TD>
        <TD>Nome</TD>
        <TD>Qualifica</TD>
      </TR>
      <TR>
        <TD>ROSSI</TD>
        <TD>MARIO</TD>
        <TD>DIRIGENTE</TD>
      </TR>
      <TR>
        <TD>VERDI</TD>
        <TD>LUIGI</TD>
        <TD>IMPIEGATO</TD>
      </TR>
      <TR>
        <TD>BIANCHI</TD>
        <TD>DAVIDE</TD>
        <TD>OPERAIO</TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```



Cognome	Nome	Qualifica
ROSSI	MARIO	DIRIGENTE
VERDI	LUIGI	IMPIEGATO
BIANCHI	DAVIDE	OPERAIO

Il tag <TABLE> : principali **attributi**

border – specifica la larghezza dei bordi di una tabella (in pixel)

cellspacing – specifica la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo.

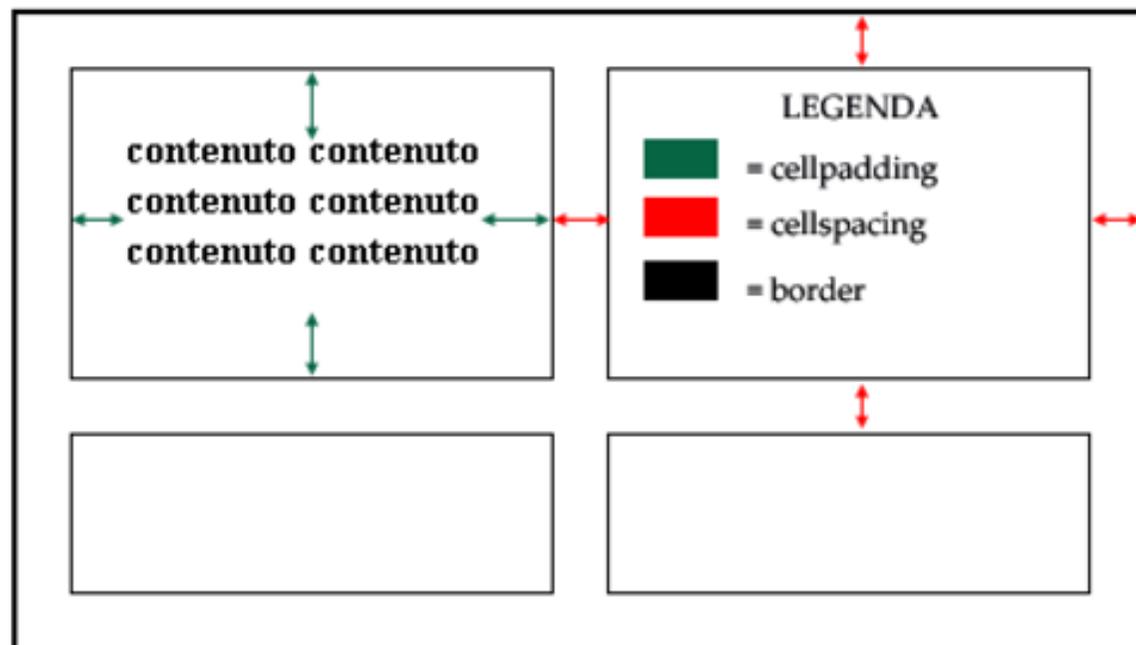
Di **default** è **un pixel**, dunque occorrerà sempre azzerarlo esplicitamente, quando non lo si desidera.

cellpadding – indica la distanza tra il contenuto della cella e il bordo.

Se il valore viene indicato con un numero intero, la distanza è espressa in **pixel**; il cellpadding tuttavia può anche essere espresso in percentuale.

Di **default** tale distanza è **nulla**

N.B. I rapporti tra gli attributi che abbiamo appena esaminato sono regolati come segue



Il tag **<TABLE>** : altri **attributi**

width – specifica la larghezza totale della tabella (in pixel o in %)

align – allinea la tabella relativamente al tag che la contiene

Valori possibili:

left : allinea la tabella a **sinistra** della pagina web

center : allinea la tabella al **centro** della pagina web

right : allinea la tabella a **destra** della pagina web

bgcolor – permette di impostare il colore di sfondo per l'intera tabella.

*Valori possibili: gli stessi di quelli relativi all'attributo **color** del tag **BODY** di cui abbiamo già parlato*

background – permette di impostare un'immagine di sfondo per l'intera tabella.

*Valori possibili: gli stessi di quelli relativi all'attributo **src** del tag **IMG** di cui abbiamo già parlato*

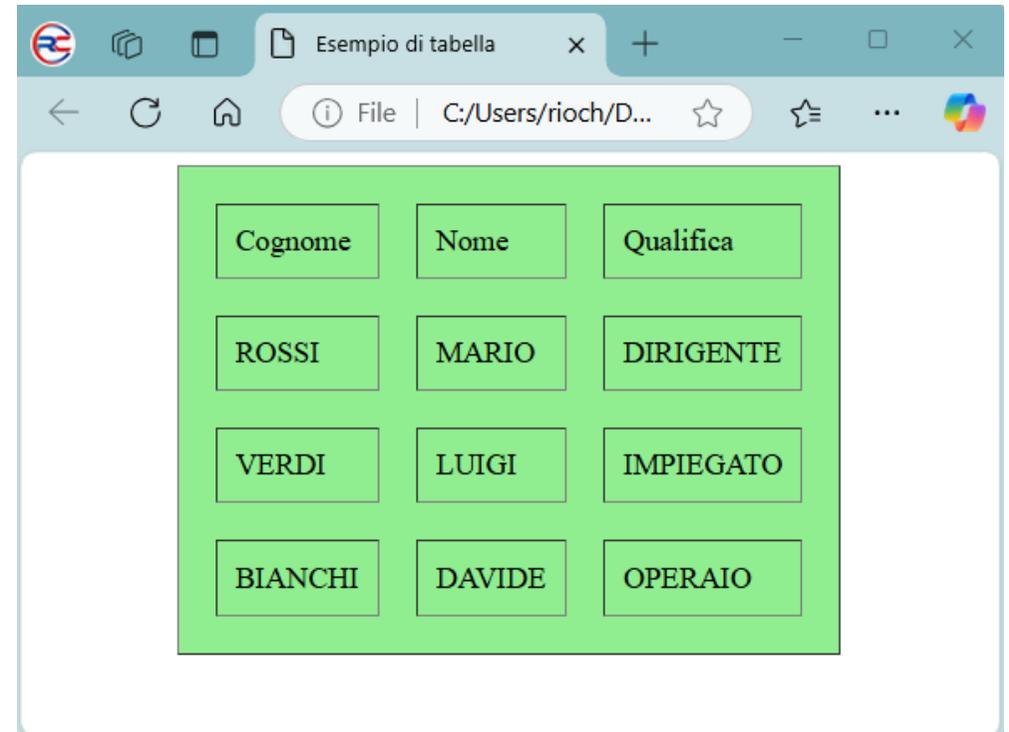
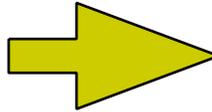
bordercolor – permette di impostare il colore del bordo l'intera tabella.

*Valori possibili: gli stessi di quelli relativi all'attributo **color** del tag **BODY** di cui abbiamo già parlato*

N.B. Nella versione HTML5 gli attributi **align**, **bgcolor**, **background** e **bordercolor** sono oramai **deprecati** in quanto occorrerebbe usare **i fogli di stile CSS**

Il tag <TABLE> : esempio

```
<HTML>
  <HEAD>
    <TITLE>Esempio di tabella</TITLE>
  </HEAD>
  <BODY>
    <TABLE border = "1"
      cellspacing = "20"
      cellpadding = "10"
      width = "300"
      align = "center"
      bgcolor = "lightgreen" >
      <TR>
        <TD>Cognome</TD>
        <TD>Nome</TD>
        <TD>Qualifica</TD>
      </TR>
      <TR>
        <TD>ROSSI</TD>
        <TD>MARIO</TD>
        <TD>DIRIGENTE</TD>
      </TR>
      <TR>
        <TD>VERDI</TD>
        <TD>LUIGI</TD>
        <TD>IMPIEGATO</TD>
      </TR>
      <TR>
        <TD>BIANCHI</TD>
        <TD>DAVIDE</TD>
        <TD>OPERAIO</TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```



Cognome	Nome	Qualifica
ROSSI	MARIO	DIRIGENTE
VERDI	LUIGI	IMPIEGATO
BIANCHI	DAVIDE	OPERAIO

I tag <TR> e <TD>: **attributi** principali

height – specifica l'altezza totale della cella (in pixel o in %)

width – specifica la larghezza totale della cella (in pixel o in %)

align – allinea il contenuto orizzontale delle celle

Valori possibili:

left : allinea a **sinistra** il contenuto delle celle della tabella

center : allinea al **centro** il contenuto delle celle della tabella

right : allinea a **destra** il contenuto delle celle della tabella

valign – allinea il contenuto verticale delle celle

Valori possibili:

top : allinea in **alto** rispetto alla baseline il contenuto delle celle della tabella

middle : allinea al **centro** rispetto alla baseline il contenuto delle celle della tabella

bottom : allinea in **basso** rispetto alla baseline il contenuto delle celle della tabella

bgcolor – permette di impostare il colore di sfondo per l'intera cella.

Valori possibili: gli stessi di quelli relativi all'attributo **color** di tag **BODY** di cui abbiamo già parlato

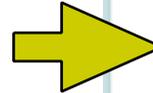
background – permette di impostare un'immagine di sfondo per l'intera cella.

Valori possibili: gli stessi di quelli relativi all'attributo **src** del tag **IMG** di cui abbiamo già parlato

N.B. Gli attributi **align** , **valign** , **bgcolor** e **background** sono oramai deprecati in quanto occorrerebbe usare **i fogli di stile CSS**

I tag <TR> e <TD>: esempio

```
<HTML>
<HEAD>
  <TITLE>Esempio di tabella</TITLE>
</HEAD>
<BODY>
  <TABLE border = "1" width = "300" cellspacing= "5"
    align = "center" bgcolor = "lightgreen" >
    <TR height = "30" align = "center"
      bgcolor = "red" >
      <TD>Cognome</TD>
      <TD>Nome</TD>
      <TD>Qualifica</TD>
    </TR>
    <TR height = "40" align = "left"
      bgcolor = "lightblue">
      <TD valign = "top">ROSSI</TD>
      <TD valign = "middle">MARIO</TD>
      <TD valign = "bottom">DIRIGENTE</TD>
    </TR>
    <TR height = "50" align = "center"
      bgcolor = "yellow">
      <TD valign = "top">VERDI</TD>
      <TD valign = "middle">LUIGI</TD>
      <TD valign = "bottom">IMPIEGATO</TD>
    </TR>
    <TR height = "60" align = "right"
      bgcolor = "orange">
      <TD valign = "top">BIANCHI</TD>
      <TD valign = "middle">DAVIDE</TD>
      <TD valign = "bottom">OPERAIO</TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
```



Cognome	Nome	Qualifica
ROSSI	MARIO	DIRIGENTE
VERDI	LUIGI	IMPIEGATO
BIANCHI	DAVIDE	OPERAIO

Il tag <TH>

Il tag <TH> serve a definire una cella di **intestazione** di una tabella HTML.

Il tag <TH> è un sottoelemento del tag <TR> che serve, come abbiamo già visto a definire, le **righe** di una tabella, che a sua volta è un sottoelemento del tag <TABLE> (riferito alla struttura semplice di una tabella HTML).

Il tag <TH> serve a definire **celle di intestazione**, generalmente ad inizio tabella nella prima riga da non confondere con il tag <TD> che serve ad inserire **celle generiche** con informazioni di vario tipo

Riassumendo:

I tag <TABLE> , <TR> , <TH> e <TD> sono i tag che servono a creare **tabelle**.

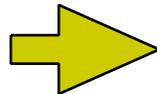
In particolare il tag <TABLE> deve racchiudere SEMPRE tutti gli altri mentre il tag <TR> indica **l'inizio di una riga** in quanto nel linguaggio HTML si ragiona **sempre prima per righe**, quindi sarà sempre il tag <TR> a contenere il tag <TD> (o eventualmente il tag il <TH>)

I tag <TH> e <TD> identificano **entrambi una cella**, ma il primo identifica **una cella che fa da intestazione** alla riga o alla colonna mentre il secondo identifica **una normale cella**

"Arricchiamo" il template di una tabella

Per scrivere una tabella che fornisca una rappresentazione **più chiara dei dati dei dati in esso contenuti** introduciamo un **template** leggermente più ricco:

```
<TABLE border = "1" cellspacing = "0">
  <CAPTION>
    I miei dati
  </CAPTION>
  <THEAD>
    <TR>
      <TH>Colonna 1</TH>
      <TH>Colonna 2</TH>
    </TR>
  </THEAD>
  <TFOOT>
    <TR>
      <TD>Totale 1</TD>
      <TD>Totale 2</TD>
    </TR>
  </TFOOT>
  <TBODY>
    <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>
  </TBODY>
</TABLE>
```



Colonna 1	Colonna 2
Dato 1,1	Dato 1,2
Dato 2,1	Dato 2,2
Dato 3,1	Dato 3,2
Totale 1	Totale 2

Diagram illustrating the structure of the table with labels: CAPTION, THEAD, TFOOT, and TBODY.

"Arricchiamo" il template di una tabella

I **tag** nuovi introdotti sono tutti definiti come "**facoltativi**", nel senso che la tabella non si "rompe" se manca uno di essi. Tuttavia si tratta di elementi fondamentali per la **semantica delle tabelle**. Esaminiamoli:

Tag	Descrizione
<CAPTION>	È una didascalia che ci permette di dare una contestualizzazione ai dati e rendere più chiaro il loro significato.
<THEAD>	Serve per raggruppare le righe che rappresentano l'intestazione della tabella.
<TH>	" <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.
<TBODY>	Raggruppa le righe che contengono il corpo della tabella, spesso con i dati veri e propri.
<TFOOT>	Racchiude le righe che utilizziamo come footer della tabella, in cui si possono inserire dei dati di riepilogo, somme, medie, etc.

Interessante notare che i tag **THEAD**, **TBODY** e **TFOOT**, creano dei raggruppamenti di righe e che il tag **TFOOT** può apparire, nel markup, anche prima del tag **TBODY**, questo per consentire ai **crawler** (chiamati anche **web crawler**, **spider** o **robot** ossia **software** con cui di solito un **motore di ricerca** analizza ed indicizza i contenuti presenti in **Internet**) di leggere **prima** i dati di sintesi e **poi** le informazioni più dettagliate.

"Arricchiamo" il template di una tabella

Schema o template generale di una tabella

			} Titolo <CAPTION>
			} Intestazione <THEAD>
			} Corpo <TBODY>
			} Piede <TFOOT>

Creiamo un sito comune #3

- 1) Variate la vostra pagina **index.htm** creando una [tabella](#) contenente una intestazione comune nella prima riga, e due colonne: una contenente la foto dei vostri compagni di classe (fatela) ed un'altra contenente l'elenco dei nominativi dei partecipanti
- 2) Aggiungete ad entrambi i "valori" (nominativo e foto) un link alle pagine personali create nelle precedenti lezioni.

Il risultato dovrà essere simile all'esempio che segue.

Esempio

3^a lezione HTML

Partecipanti



Pascal Blaise

Leibniz Gottfried Wilhelm

Babbage Charles

Byron Augusta Ada

Hollerith Herman

Zuse Konrad

Aiken Howard

Turing Alan

Wirth Niklaus

Jobs Steve

Wozniak Steve

I tag <TH> e <TD>: **attributi** speciali

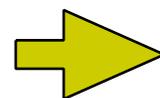
L'attributo **rowspan** permette di **unire verticalmente più celle** in una tabella.

L'attributo **colspan** permette di **unire orizzontalmente più celle** in una tabella.

L'attributo **colspan**: approfondimento

Prendiamo ad esempio **una tabella 3x3** un cui vogliamo che una cella si estenda su **2 colonne**:

```
<HTML>
  <HEAD>
    <TITLE>Esempio di tabella (COLSPAN)</TITLE>
  </HEAD>
  <BODY>
    <TABLE border = "1" cellspacing = "0">
      <TR>
        <TD colspan="2">doppia cella</TD>
        <TD>cella</TD>
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
        <TD>cella</TD>
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
        <TD>cella</TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```

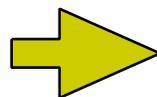


doppia cella	cella	
cella	cella	cella
cella	cella	cella

L'attributo **rowspan**: approfondimento

Sempre relativamente alla nostra **tabella 3x3** supponiamo di volere che una cella che si estenda su **2 righe** :

```
<HTML>
  <HEAD>
    <TITLE>Esempio di tabella (ROWSPAN)</TITLE>
  </HEAD>
  <BODY>
    <TABLE border = "1" cellspacing = "0">
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
        <TD>cella</TD>
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
        <TD rowspan="2">doppia cella
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```

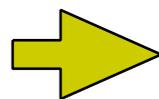


cella	cella	cella
cella	cella	doppia cella
cella	cella	

Gli attributi **rowspan** e **colspan**: approfondimento

Effettuiamo ora il merge dei due esempi appena mostrati:

```
<HTML>
  <HEAD>
    <TITLE>Esempio di tabella (COLSPAN e ROWSPAN)</TITLE>
  </HEAD>
  <BODY>
    <TABLE border = "1" cellspacing = "0">
      <TR>
        <TD colspan="2">doppia cella</TD>
        <TD>cella</TD>
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
        <TD rowspan="2">doppia cella</TD>
      </TR>
      <TR>
        <TD>cella</TD>
        <TD>cella</TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```



doppia cella		cella
cella	cella	doppia cella
cella	cella	

Creiamo un sito comune #4

- 1) Cercate su Internet come creare un [link ad un indirizzo di e-mail o ad un sito web](#)
- 2) Qual è la differenza fra un link e un [bookmark](#)?
Usate il vostro sito per sperimentare.
- 3) Variate la pagina personale aggiungendo due link, uno per scrivervi una e-mail, l'altro per collegarsi esternamente ad un eventuale vostro sito web personale.
- 4) Procedete variando tutte le pagine dei partecipanti come indicato al punto (3), oppure rispeditevi le pagine personali opportunamente modificate.

Il risultato dovrà essere simile all'esempio che segue.

Il tag <IFRAME>

Un **IFRAME** (**I**nline **f**rame) è un **elemento HTML** che consente di incorporare un altro documento HTML nella pagina corrente.

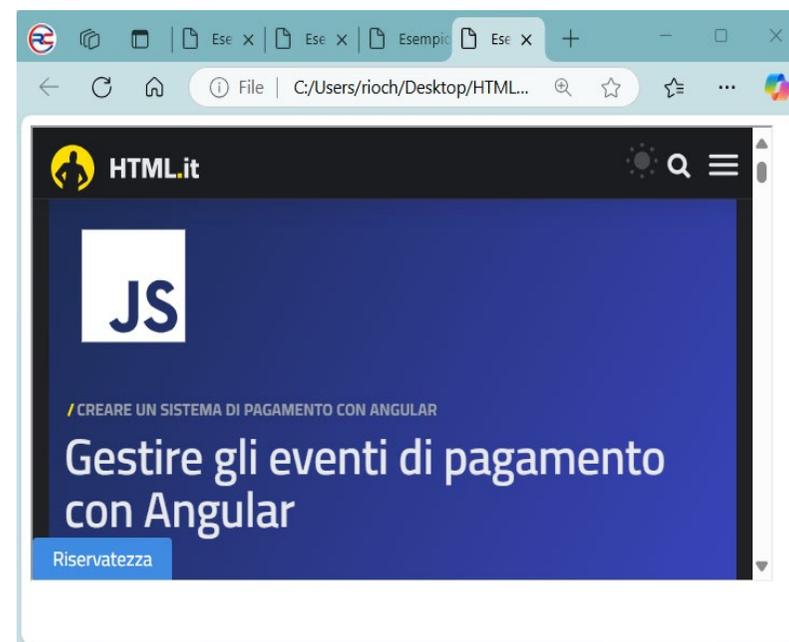
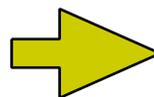
Gli **Inline frame** hanno molte applicazioni pratiche che estendono il design standard e la funzionalità di una pagina web. Le più importanti sono:

- **incorporamento di contenuti multimediali** (file video, file audio ed animazioni) da piattaforme come **Youtube, Vimeo, Soundcloud**, etc.
- **integrazione di applicazioni di terze parti** (es. widget di **Google Maps** o un **gateway di pagamento**)
- **visualizzazione di pubblicità** (potendo così separare il suo contenuto dalla nostra pagina web)
- **caricamento di contenuti esterni** (come **feed di notizie, aggiornamenti meteo** o **post sui social media**)
- **migliorare design e layout** permettendo la creazione di pagine web con layout più complessi
- creare un sito all'interno di un sito permettendo ai visitatori di navigarlo senza lasciare la pagina originale (utile per ospitare **micrositi** o **cataloghi di prodotti**)

Aggiungere un tag **IFRAME** a un documento HTML

Questo è il modo universale per aggiungere un iframe a una pagina web. È necessario modificare il codice HTML della pagina e inserire il tag **<IFRAME>** che crea l'area in cui il contenuto esterno verrà visualizzato.

```
<IFRAME  
src = "https://www.html.it/"  
width = "560"  
height = "315"  
title = "Guida HTML">  
</IFRAME>
```



NOTA BENE

Ci sono altri attributi che modificano le proprietà dell'**iframe** in vari modi, come l'attributo **sandbox**, l'attributo **name**, l'attributo **srcdoc**, l'attributo **allow**, l'attributo **target**, l'attributo **loading** e molti altri che vi invito a scoprire da soli all'URL

https://www.w3schools.com/tags/tag_iframe.ASP

Aggiungere un tag **IFRAME** a un documento HTML

Analizziamo gli **elementi che compongono il tag <IFRAME>**

- Il tag di apertura **<IFRAME>** identifica l'elemento come un IFRAME.
- L'attributo **src** definisce l'indirizzo URL della sorgente IFRAME. La risorsa esterna trovata su questo indirizzo verrà visualizzata nella sezione IFRAME sulla pagina web.
- L'attributo **height** definisce la dimensione verticale dell'elemento IFRAME. Puoi utilizzare una dimensione fissa (pixel) o automatica (impostata come percentuale della dimensione dello schermo).
- Gli attributi **width** definiscono la larghezza dell'IFRAME. Simile all'altezza, anche la larghezza può assumere valori fissi o automatici.
- L'attributo **title** fornisce una descrizione dell'IFRAME, che può essere utilizzata dai motori di ricerca e dai visitatori che utilizzano tecnologie di assistenza.
- Il tag **</IFRAME>** chiude il codice IFRAME.

I Cascading Style Sheets o CSS

Essenzialmente il CSS (o fogli di stile) è un linguaggio che gestisce il **design** e la **presentazione** delle pagine web (cioè l'aspetto estetico) e lavora in combinazione con l'HTML, o HyperText Markup Language (linguaggio a marcatori per ipertesti), il quale invece gestisce i contenuti delle pagine.

Con il **CSS** è possibile stabilire le regole secondo cui il sito dovrà mostrare le informazioni, mantenendo tutti i comandi relativi alla formattazione e allo stile (font, colori, dimensioni, ecc) separati da quelli relativi al contenuto.

Esistono due tipi di fogli di stile CSS: **interni** ed **esterni**.

- il CSS **interno** prevede che il **codice CSS** sia scritto direttamente all'interno del documento HTML di cui si vuole modificare l'aspetto grafico ed è possibile farlo adottando due tecniche differenti:
 - a) Applicare uno stile **inline**
 - b) Usare un **foglio di stile incorporato**
- il CSS **esterno** prevede di inserire tutte le istruzioni CSS che si intende utilizzare all'interno di un semplice file di testo che dovrà poi essere salvato con estensione **.css**

I Cascading Style Sheets o CSS INTERNI (1)

a) Applicare uno stile **inline**

Si dice **in linea (inline)** l'utilizzo delle proprietà di stile direttamente all'interno del tag HTML che si desidera formattare.

Così facendo le varie proprietà vengono scritte all'interno dell'attributo **style** il quale **può essere applicato ad ogni tag HTML della pagina**.

Vediamo un esempio:

```
<P style = "color : #FF0000;">Testo rosso</P>
```

Nel nostro esempio abbiamo applicato il colore rosso ad un paragrafo di testo.

Nota Bene: questa tecnica non è consigliata in quanto non realizza completamente l'auspicata separazione di struttura e formattazione. Il suo utilizzo, seppur non del tutto deprecabile, dovrebbe essere limitato a singoli casi specifici e non certo utilizzato quale tecnica abituale di web design!

I Cascading Style Sheets o CSS INTERNI (2)

b) Usare un foglio di stile incorporato

All'interno del documento HTML possono essere definiti dei veri e propri fogli di stile.

Questi consistono in un **insieme di istruzioni CSS** (riguardanti più elementi della pagina) che vengono racchiuse all'interno dei tag **<STYLE>** e **</STYLE>**.

Normalmente queste istruzioni vengono inserite nell'intestazione del documento HTML (tra **<HEAD>** e **</HEAD>**) ma nulla vieta che siano inserite anche all'interno del **body**.

```
<HTML>
  <HEAD>
    <TITLE>Esempio di CSS INCORPORATO</TITLE>
    <STYLE type="text/css">
      P { color: #FF0000; }
    </STYLE>
  </HEAD>
  <BODY>
    <P>Testo rosso</P>
  </BODY>
</HTML>
```

I Cascading Style Sheets o CSS INTERNI (3)

```
<HTML>
  <HEAD>
    <TITLE>Esempio di CSS INCORPORATO</TITLE>
  </HEAD>
  <BODY>
    <P style = "color:#FF0000;">Testo rosso</P>
  </BODY>
</HTML>
```

A

```
<HTML>
  <HEAD>
    <TITLE>Esempio di CSS INCORPORATO</TITLE>
    <STYLE type = "text/css">
      P { color: #FF0000; }
    </STYLE>
  </HEAD>
  <BODY>
    <P>Testo rosso</P>
  </BODY>
</HTML>
```

B

Il risultato che otteniamo con i due segmenti di codice HTML è lo stesso, ma esistono una serie di differenze logiche e pratiche tra l'implementazione **inline (A)** e quella attraverso **un foglio di stile (B)**.

Infatti:

assegnando uno stile ad un unico elemento in una pagina si otterrà quello stile solo per quello specifico elemento, come se lo si trattasse in stile HTML classico;

assegnando invece uno stile nell'intestazione della pagina ad un tag, tutti gli elementi dello stesso tipo di quella pagina assumeranno lo stesso stile.

I Cascading Style Sheets o CSS ESTERNI (1)

Come detto è possibile definire anche **esternamente** un foglio di stile. Così facendo si realizza quella perfetta separazione tra struttura e formattazione che era negli intenti dei creatori dei CSS.

In questo senso possiamo dire che questa soluzione rappresenta la tecnica implementativa "**ideale**" per i CSS.

Per creare un foglio di stile esterno è sufficiente inserire **tutte le istruzioni CSS all'interno di un semplice file di testo** che dovrà poi essere salvato con estensione **.css**, ad esempio **style.css**, come di convenzione fanno la maggior parte dei software per la gestione dei CSS e la maggior parte dei Webmaster.

Nulla vieta, tuttavia, di utilizzare altri nomi perché ciò che conta è **l'estensione del file**.

Una volta creato **il file CSS esterno** sarà sufficiente **richiamarlo** nell'header della pagina attraverso il tag <LINK>, (vedi esempio che segue)

I Cascading Style Sheets o CSS ESTERNI (2)

```
<HTML>
  <HEAD>
    <TITLE>Esempio CSS esterno</title>
    <LINK rel="stylesheet" type="text/css" href="style.css">
  </HEAD>
  <BODY>
    <P>Testo rosso</P>
  </BODY>
</HTML>
```

Volendo analizzare il tag **LINK** col quale abbiamo **importato** il nostro foglio di stile CSS:

- l'attributo **rel** indica la relazione tra il documento HTML ed il file linkato;
- l'attributo **type** definisce il tipo di file (nel nostro caso **text/css**);
- l'attributo **href**, infine, fa riferimento al **nome** ed al **percorso (path)** del file CSS da richiamare all'interno della pagina.

Il codice contenuto all'interno del file **style.css** (che in questo caso si troverà nella stessa directory del nostro file HTML) del nostro esempio sarà il seguente:

```
P { color: #FF0000; }
```

I Cascading Style Sheets o CSS ESTERNI (3)

Il codice contenuto all'interno del file **style.css** (che in questo caso si troverà nella stessa directory del nostro file HTML) del nostro esempio sarà il seguente:

```
P { color: #FF0000; }
```

e l'effetto che si otterrà sarà identico a quello degli esempi precedenti, con la differenza che **con i fogli di stile interni** sarà necessario **ripetere la stilizzazione pagina per pagina** (se non elemento per elemento se si utilizza la tecnica inline) mentre con **i fogli di stile esterni** lo stesso stile potrà essere utilizzato su tutti i documenti consentendo modifiche molto più veloci e su scala globale intervenendo su un unico file (il file **.css** appunto).

CONCLUSIONI

Si noti che le diverse tecniche (stile **inline**, foglio di **stile incorporato** ed **esterno**) non sono tra lo incompatibili. Quindi all'interno di uno stesso documento HTML nulla vieta di utilizzare le diverse tecniche.

E nel caso di conflitti? Se cioè per lo stesso elemento venissero definiti stili differenti nel foglio di stile e poi mediante una stilizzazione inline?

In questo caso, normalmente, "vince" la formattazione in linea (salvo casi particolari).

Riferimenti

Il linguaggio HTML in poche pagine:

https://vision.unipv.it/stm-cim/m_at09/stmcim-html.pdf

Prima Guida al linguaggio HTML:

<https://www.html.it/guide/guida-html/>

Seconda Guida al linguaggio HTML:

<https://aulab.it/guide-avanzate/1/guida-html-e-css-in-italiano/>

HTML Tutorial – w3schools.com (inglese):

<http://www.w3schools.com/html/default.asp>