

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
</body>
</html>
```

Web Coding

*Prototipazione di ipertesti e siti web in HTML
ed introduzione alla creazione di stili grafici con fogli di stile CSS.*

Unità Didattica UD01: Funzionamento del browser. Basi di HTML, formattazione del testo

prof. Giovanni Borga

Hyper Text Markup Language - HTML

Introduzione e funzionamento dei Browsers

L'HTML e i browser

L'HTML (Hyper Text Markup Language) è il linguaggio con cui si indica come i vari elementi vanno disposti in una pagina Web.

Un documento html non è nient'altro che un **file di testo** con delle indicazioni sul colore delle scritte, sulla posizione delle immagini all'interno della pagina, su come far scorrere il testo, e altre cose di questo genere.

L'HTML NON è un linguaggio di programmazione. Più propriamente è un linguaggio di «formattazione»

Il Browser è il programma che si usa per navigare nel Web. Svolge principalmente due compiti:

- scarica i vari files che si trovano su un computer remoto (il server) e che fanno riferimento a un determinato indirizzo
- legge i documenti scritti in html e, a seconda delle indicazioni ivi contenute, visualizza la pagina in un modo, piuttosto che in un altro; inoltre i vari files associati a quel documento (ad esempio le immagini, o i contenuti multimediali) vengono disposti secondo le indicazioni del codice

I browser attualmente più diffusi:

Chrome, Firefox, Edge,
Safari, Opera



L'HTML e i browser

Acquisire una mentalità multi-browser

Per lo sviluppo dei siti non occorre tanto conoscere nei minimi dettagli il codice HTML, quanto piuttosto sapere come il codice HTML verrà visualizzato sul computer dell'utente.

Infatti uno dei lavori più difficili è quello di riuscire a far vedere correttamente il proprio sito con i browser e le piattaforme più svariate.

La cache

I file scaricati dal web vengono memorizzati in una particolare cartella del computer che prende il nome di:

cronologia o «cache»



Può succedere - soprattutto a chi è alle prime armi - di continuare a modificare un file, ma di non riuscire a vederne le modifiche. Questo succede perché la pagina visualizzata è sempre quella vecchia memorizzata nella cache.

In molti browser è possibile impostare le preferenze in modo che venga sempre cancellata la cache quanto si chiude il programma.

Motore di rendering

La visualizzazione di un file html da parte del browser prende il nome di

rendering della pagina

Il cosiddetto «**motore di rendering**» è quella sezione del browser che si occupa di visualizzare la pagina applicando stili, layout ed elementi *embedded* secondo quanto indicato nel codice HTML.

In qualsiasi momento è possibile visualizzare il codice HTML delle pagine che stiamo visitando utilizzando la funzione di visualizzazione sorgente:



Funzionamento di un browser

Hypertext Markup Language ("Linguaggio di contrassegno per gli Iper testi") non è un linguaggio di programmazione ma un linguaggio di contrassegno (o 'di marcatura'), che permette di indicare come disporre gli elementi all'interno di una pagina: le indicazioni vengono date attraverso degli appositi marcatori, detti "**tag**".

- *Alcuni linguaggi di programmazione sono: C, C++, Visual Basic, Pascal, Java;*
- *Alcuni linguaggi di scripting sono: PHP, ASP, PERL, JavaScript.*

Ciò significa che l'HTML:

- a) non ha meccanismi che consentono di prendere delle decisioni ("in questa situazione fai questo, in quest'altra fai quest'altro")
- b) Non è in grado di compiere delle iterazioni ("ripeti questa cosa, finché non succede questo"), né ha altri costrutti propri della programmazione.

Pur essendo dotato di una sua sintassi, non presuppone la logica ferrea e inappuntabile dei linguaggi di programmazione: se vi dimenticate di chiudere un tag, non verranno prodotti dei messaggi di errore; se non rispettate la sintassi probabilmente non otterrete la visualizzazione della pagina che desiderate, ma nient'altro. A volte può essere necessario adottare dei "trucchetti", non proprio da manuale, pur di visualizzare la pagina correttamente con i diversi browser.

Lo standard HTML

L'organizzazione che si occupa di standardizzare la sintassi del linguaggio HTML è il **W3C: World Wide Web Consortium**.

Ha rilasciato diverse versioni di questo linguaggio (HTML 2.0, HTML 3.2, HTML 4.0); e - da un certo punto in poi - l'HTML si è evoluto in **XHTML** (HTML riformulato con le regole sintattiche del XML).

In questo corso, inizieremo con la versione 4.01 rilasciata il 24 dicembre 1999, i cui tag di base sono sostanzialmente rimasti invariati, per poi passare alla 5.x che semplifica alcuni aspetti.

La maggioranza dei browsers supporta già da tempo la **versione HTML 5** che però è rilasciata ufficialmente ancora in versione draft.

HTML 5 non soppianta la versione 4 ma integra nuove funzionalità e ottimizza alcuni aspetti.



HTML 4.01 Specification

W3C Recommendation 24 December 1999

This version:

<http://www.w3.org/TR/1999/REC-html401-19991224>
([plain text \[794Kb\]](#), [gzip'ed tar archive of HTML files \[371Kb\]](#), [a \[963Kb\]](#))

Latest version of HTML 4.01:

<http://www.w3.org/TR/html401>

Latest version of HTML 4:

<http://www.w3.org/TR/html4>

Latest version of HTML:

<http://www.w3.org/TR/html>

Previous version of HTML 4.01:

<http://www.w3.org/TR/1999/PR-html40-19990824>

Previous HTML 4 Recommendation:

<http://www.w3.org/TR/1998/REC-html40-19980424>

Editors:

[Dave Raggett](mailto:dsr@w3.org) <dsr@w3.org>

Arnaud Le Hors, W3C

Ian Jacobs, W3C

Da HTML 4.01 a HTML 5

- Regole più stringenti per la **strutturazione del testo** in capitoli, paragrafi e sezioni;
- Nuovi elementi di controllo per i **menu di navigazione**;
- Miglioramenti ed estensione degli elementi di controllo per i **moduli**;
- Nuovi elementi specifici per il controllo di **contenuti multimediali** (tag <video> e <audio>);
- **Estensione degli attributi** di quasi tutti i tag, specialmente quelli finalizzati all'accessibilità, finora previsti solo per alcuni tag;
- Supporto per il **Canvas** che permette di utilizzare JavaScript per creare animazioni e grafica bitmap;
- Introduzione della **geolocalizzazione**, dovuta ad una forte espansione di sistemi operativi mobili (quali Android e iOS, tra i più diffusi);
- Introduzione di un sistema alternativo ai normali cookie, più efficiente, chiamato **Web Storage** che consente anche un notevole risparmio di banda;
- Standardizzazione di programmi JavaScript, chiamati **Web Workers** e possibilità di utilizzare siti offline;
- Sostituzione del tag **doctype** inutilmente prolisso, con un semplice <!DOCTYPE html>
- Deprecati o **eliminati elementi di scarso o nessun utilizzo** effettivo

Estensioni dei file

Apriete o create un documento con il blocco note, e salvate il file in qualche cartella del vostro computer con **estensione "html"**, ad esempio **miaPagina.html**.

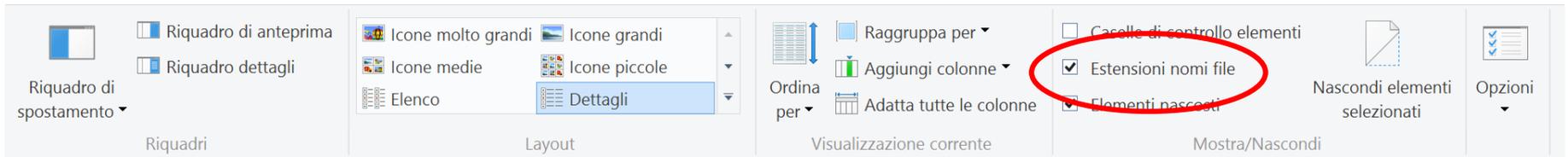
Una pagina web è «semplicemente» un file ASCII con estensione .html

Fino a qualche tempo fa si era soliti attribuire ai file l'estensione htm, ma questo avveniva perché il dos e poi Windows 3.1 non erano in grado di gestire i file con nomi di grandezza superiore a 8 caratteri ed estensione superiore alle 3 lettere. Dunque .html era diventato .htm, così come .jpeg era diventato .jpg.

Si possono utilizzare liberamente entrambe le estensioni ma ormai la consuetudine è quella di utilizzare .html

Normalmente il sistema operativo associa automaticamente le estensioni .htm e .html al browser predefinito.

NB: Se accade che il browser non «riconosce» la vostra pagina html verificate di non aver salvato con una «doppia estensione» (es. .html.txt). Questo può succedere se il vostro sistema operativo è impostato per nascondere le estensioni per i files già associati ad un'applicazione.



Struttura dei tag HTML

Sintassi del linguaggio

I «mattoncini» della pagina HTML: i TAG

Come abbiamo detto in precedenza, all'interno di ogni pagina è presente una **serie di marcatori**.

I marcatori di cui si compone una pagina HTML prendono il nome di TAG

Ai tag viene affidata la formattazione e disposizione dei contenuti.

Hanno differenti nomi a seconda della loro funzione.

<TAG attributi>contenuto</TAG>

Sintassi di base:

I TAG vanno inseriti con le seguenti regole sintattiche:

- L'apertura del tag è una stringa tra **parentesi uncinate** (<TAG>)
- La chiusura del tag viene indicata con una "/" (è il simbolo comunemente detto "**slash**". </TAG>.
- Il primo elemento dopo l'apertura della prima parentesi è detto «**nome**» del tag e può essere seguito da «attributi»
- Nel tag di chiusura non può esserci nulla tranne lo slash e il nome del tag
- Ciò che sta tra il tag di apertura e il tag di chiusura è detto «**contenuto**» del tag.

Esempi di tag e struttura degli attributi

Ecco un esempio, con una sintassi che serve a disporre un testo giustificato a destra:

```
<p align="right">testo</p>
```

Dall'esempio è evidente che la **struttura di un attributo** è: attributo="valore" ; la gamma di attributi è definito e dipende dallo specifico tag. I valori degli attributi vanno indicati tra doppi apici.

In definitiva la **struttura complessiva di un tag** è la seguente:

```
<TAG attributo_1="valore1" attributo_2="valore2">contenuto</TAG>
```

Alcuni particolari tag non hanno contenuto - perché ad esempio indicano solo la posizione di alcuni elementi (ad es. il tag delle immagini), conseguentemente questi tag non hanno la chiusura. Vengono detti appunto **tag vuoti**.

La loro forma in origine era dunque: <TAG attributi>, tuttavia con l'adozione delle specifiche XHTML si rende necessario lo slash prima della seconda parentesi; ecco un esempio di tag per inserimento di un'immagine:

```
<IMG widht="20" height="20" src="logo.gif" alt="logo aziendale" />
```

Disposizione dei tag

Una pagina HTML può essere vista come un «flusso» sequenziale di tag dove però occorre rispettare la gerarchia di apertura e chiusura dei tag senza intersecarli.

Questa sequenza è corretta:

```
<TAG1>contenuto 1<TAG2>contenuto 2</TAG2></TAG1>
```

Quest'altra no:

```
<TAG1>contenuto 1<TAG2></TAG1>contenuto 2</TAG2>
```

Nella sequenza corretta è altrettanto corretto affermare che il TAG2 è un contenuto del TAG1. Tecnicamente il TAG2 è annidato nel TAG1. Possiamo scrivere il codice anche andando a capo per renderlo più leggibile:

```
<TAG1>  
contenuto 1  
<TAG2>contenuto 2</TAG2>  
</TAG1>
```

Si consideri infine la proprietà di base per cui **i contenuti nel flusso HTML tutte le interruzioni di riga e tutti gli spazi vengono renderizzati come un unico spazio.**

Per inserire più spazi o più righe esistono apposite codifiche. Vedremo più avanti che alcuni tag costituiscono interruzione del flusso e quindi mandano il contenuto a nuova riga mentre altri no.

Annidamento e indentazione

La logica di tipo strettamente gerarchico dell'HTML fa sì che sia quantomeno conveniente «indentare» le righe di codice in modo da rendere più leggibile il listato e limitare l'inserimento di errori come ad esempio l'omissione del tag di chiusura o l'errato annidamento.

Ad esempio il seguente frammento di codice:

```
<TAG1 attributi>contenuto 1<TAG2>contenuto 2</TAG2></TAG1>
```

E' più leggibile se scritto così:

```
<TAG1 attributi>  
contenuto 1  
  <TAG2>  
    contenuto 2  
  </TAG2>  
</TAG1>
```

Ma ancor meglio così:

```
<TAG1 attributi>  
  contenuto 1  
    <TAG2>  
      contenuto 2  
    </TAG2>  
  </TAG1>
```

In questo molti editor aiutano lo sviluppatore con l'indentazione automatica e la colorazione del codice. Per indentare si utilizza generalmente il carattere di tabulazione (TAB).

Commenti

Un strategia adottata universalmente da tutti gli sviluppatori per **rendere più comprensibile un listato** (esiste una sintassi per i commenti per ogni linguaggio), è quella di inserire dei «commenti» nei punti significativi.

Si tratta di indicazioni significative per lo sviluppatore, ma sostanzialmente «invisibili» al browser; i commenti quindi non hanno alcuna influenza nel rendering della pagina.

La sintassi dei commenti HTML è la seguente:

```
<!-- questo è un commento HTML -->
```

Il commento può anche comprendere più righe di codice:

```
<!--  
    questo è un commento  
    che comprende più righe di testo  
    <b>testo in grassetto</b>  
-->
```

Si noti che tutti i tag compresi nei commenti non vengono renderizzati.

Commenti

```
<html>  
<!-- qui inizia la pagina -->  
  <head>  
    <!-- qui l'intestazione -->  
      <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">  
      <title>  
        <!-- qui il titolo -->  
        HTML.it  
      </title>  
    </head>  
    <body>  
      <!-- qui il contenuto visibile della pagina-->  
      Lorem ipsum factum ...  
    </body>  
</html>
```

Maiuscolo / minuscolo

L'**HTML** è «**case insensitive**», cioè indipendente dal maiuscolo e minuscolo. E' quindi del tutto indifferente scrivere tag e attributi in maiuscolo o in minuscolo.

<P ALIGN="RIGHT">

e

<p align="right">

vengono renderizzati allo stesso modo dal browser.

NB: naturalmente i contenuti sono invece «case sensitive» in quanto sono ciò che viene visualizzato materialmente nella pagina.

Fino a qualche tempo fa, per aumentare la leggibilità del codice, era consuetudine scrivere in maiuscolo il nome del tag (es: <P>) e in minuscolo gli attributi (es: align="right"). Quindi: <P align="right">.

Tuttavia oggi, per analogia con l'XHTML (che è figlio dell'XML e dell'HTML ed è «case sensitive») è **consigliabile scrivere tutto in minuscolo**, fermo restando che se per errore dovessimo scrivere dei tag in maiuscolo ciò non costituisce un errore.

Struttura di una pagina HTML

Le parti fondamentali che compongono una pagina web

Macrostruttura della pagina

Una pagina HTML è sostanzialmente composta di una riga che definisce un **tipo di documento** tra i diversi standard definiti nel corso degli anni, e il **documento** stesso che a sua volta è composto di intestazione e corpo. **Nel corpo viene inserita la gran parte dei contenuti da visualizzare.**



La pagina HTML tipo

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Titolo della pagina</title>
  </head>
  <body>
    contenuto ....
  </body>
</html>
```

La pagina tipo in HTML 5

```
<!doctype html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Titolo della pagina</title>  
  </head>  
  <body>  
    contenuto ...  
  </body>  
</html>
```

Tipo del documento: la «doctype declaration»

Detta DTD (Document Doctype Declaration), la prima riga di codice di una pagina HTML definisce quale tipo di definizione si intende utilizzare.

I più comuni doctype sono:

- **HTML 4.01 Strict**
- **HTML 4.01 Transitional**
- **HTML 4.01 Frameset**
- **XHTML 1.0 Strict**
- **XHTML 1.0 Transitional**
- **XHTML 1.0 Frameset**
- **XHTML 1.1**
- **HTML 5**

*Le **versioni** sono sostanzialmente 3: HTML 4 e 5 e XHTML già descritte nella prima lezione.*

*Le **varianti** sono: Strict, Transitional, Frameset*

La variante Strict non contiene gli elementi «deprecati» a differenza della Transitional. Frameset è uguale Transitional ma consente in più l'uso dei frames.

Il doctype HTML 4.1 Transitional è ancora molto usato

Tuttavia si è pienamente affermato il **HTML 5 che è quello che utilizzeremo nel corso.**

Doctype declaration 4.1

La DTD è la prima riga del listato HTML:

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

!DOCTYPE

Apertura del doctype

HTML

Indica che il linguaggio utilizzato per il documento corrente è l'HTML

PUBLIC

Indica che il documento è pubblico

-//W3C//DTD HTML 4.01 Transitional//EN

Indica che il documento si riferisce ad una specifica del W3C, che il DTD è HTML versione 4.01 Transitional e che la lingua è l'inglese

http://www.w3.org/TR/html4/loose.dtd

E' la URL alla quale è raggiungibile la specifica relativa alla DTD del documento; questa indicazione è facoltativa (e nella prassi frequentemente omessa).

L'intestazione (HEAD)

Nell'HEAD della pagina vengono inseriti elementi non immediatamente percepibili, ma che riguardano il modo in cui il documento deve essere letto e interpretato dal browser.

`<head></head>`

In particolare in questa sezione vengono inseriti 4 fondamentali elementi:

- Il **Title** (il titolo della pagina che compare sulla barra della finestra del browser)
- I **meta-tag** (ad esempio quelli a beneficio dei motori di ricerca)
- Gli **script** di carattere generale (come ad esempio le definizioni di funzioni)
- I riferimenti ai **fogli di stile**

Il tag head non ha attributi.

Il tag TITLE

Il TITLE è il titolo della pagina e compare in alto sulla barra del browser.

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

È bene compilarlo da subito, onde evitare di dimenticarlo e di avere pagine senza titolo. Questo elemento è strategico per almeno due motivi:

- 1) Viene visualizzato sulle **barre delle finestre** ma soprattutto delle **schede** di molti browser e consente l'individuazione dei diversi contenuti a colpo d'occhio senza dover visualizzare necessariamente la pagina;
- 2) Il valore del tag è la parola chiave che assume l'importanza maggiore nell'indicizzazione automatica effettuata dai **motori di ricerca**.

Set di caratteri

Si utilizzano i CHARSET per indicare al browser quale set di caratteri utilizzare nel rendering:

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

Questo meta-tag indica l'utilizzo del set iso-8859-1, ovvero il set di caratteri latini utilizzati nei paesi occidentali.

<meta charset="utf-8">

Questo meta-tag (espresso nella forma breve prevista dall'HTML 5) indica l'utilizzo del set esteso utf-8, detto anche Unicode, che comprende il set base dell'iso-8859-1 più una serie di altri caratteri usati in quasi tutte le lingue vive e in alcune lingue morte, nonché simboli matematici e chimici, cartografici, l'alfabeto Braille, ideogrammi ecc.

In teoria, qualsiasi codifica dei caratteri può essere utilizzata, ma nessun browser è in grado di interpretarle tutte. L'utilizzo di una codifica molto ampia come l'unicode aumenta la possibilità che la maggioranza dei browser visualizzino correttamente i contenuti.

Il corpo (BODY)

Nel BODY si inserisce la gran parte del contenuto di una pagina.

`<body></body>`

Il tag BODY può essere pensato come il «foglio» su cui disponiamo i contenuti ovvero la superficie sopra la quale vengono renderizzati tutti gli altri elementi.

Tale superficie dispone dei seguenti attributi:

- BGCOLOR (colore di sfondo)
- BACKGROUND (immagine di sfondo)
- LEFTMARGIN e TOPMARGIN (marginari in orizzontale e in verticale)
- **TEXT (colore del testo)**
- LINK (colore dei collegamenti ipertestuali)
- ALINK (colore dei link attivi)
- VLINK (colore dei link visitati)

Oltre ai cosiddetti «attributi globali», in particolare quelli per i fogli di stile: STYLE, CLASS e ID

Va detto però che tutti gli attributi di cui sopra, tranne TEXT, sono considerati DEPRECATI in virtù dell'adozione dei fogli di stile

Elementi HTML di base

Body, titoli, paragrafi blocchi di testo e contenitori

Attributi deprecati

Molti degli attributi che vedremo in questa unità didattica e nelle seguenti sono considerati **DEPRECATI**, ovvero obsoleti in quanto sostituiti da specifiche più moderne ed efficaci.

E' però importante conoscerle perché sono comunque supportate dai browser e utilizzate da moltissimi sviluppatori.

Dal punto di vista didattico inoltre, lo studio di questi elementi è utili per **comprendere evoluzione e potenzialità del linguaggio**, soprattutto per quanto riguarda **l'integrazione con i fogli di stile CSS**,

Colore di sfondo della pagina

Il colore di sfondo di una pagina si applica mediante un attributo del tag body: BGCOLOR.

Es.

```
<body bgcolor="blue">
```

bgcolor sta per "background color", cioè "colore di sfondo". Un set di colori è disponibile utilizzando apposite parole chiave in inglese, ma tutta la gamma RGB è applicabile mediante una codifica esadecimale.

colore	parola chiave	notazione esadecimale
arancione	orange	#FFA500
blu	blue	#0000FF
bianco	white	#FFFFFF
giallo	yellow	#FFFF00
grigio	gray	#808080
marrone	brown	#A52A2A
nero	black	#000000
rosso	red	#FF0000
verde	green	#008000
viola	violet	#EE82EE

Codifica dei colori in HTML

Nella storia del web e dei dispositivi informatici, la cosiddetta «profondità colore», ovvero il numero di toni cromatici visualizzabili, dagli schermi digitali è progressivamente passata dai 256 ai 65K per arrivare oggi a **16M di colori** possibili grazie all'utilizzo di una codifica a 24bit.

Poiché, soprattutto in passato, non vi era modo di sapere quale video utilizzasse ogni utente è stata codificata una cosiddetta "palette sicura" che comprende i 256 colori che sicuramente chiunque sarebbe stato in grado di visualizzare correttamente. Si tratta della cosiddetta palette «web safe» che ormai è comunque quasi completamente caduta in disuso.

Per indicare un colore si usa la seguente codifica:

#RRGGBB

Dove i tre gruppi di caratteri possono assumere i valori esadecimali da 00 a FF (ovvero da 0 a 256 in decimale) RR è la componente rossa, GG la componente verde, BB la componente blu.

black (#000000)	silver (#C0C0C0)	gray (#808080)	white (#FFFFFF)
maroon (#800000)	red (#FF0000)	purple (#800080)	fuchsia (#FF00FF)
green (#008000)	lime (#00FF00)	olive (#808000)	yellow (#FFFF00)
navy (#000080)	blue (#0000FF)	teal (#008080)	aqua (#00FFFF)

Codifica dei colori in HTML

Si noti che con la codifica #RRGGBB si possono ottenere i colori fondamentali rosso, verde e blu ponendo a FF la rispettiva coppia di caratteri: **#FF0000**, **#00FF00**, **#0000FF**.

Inoltre si possono ottenere i colori «secondari» giallo, ciano e magenta ponendo a FF due coppie: **#FFFF00**, **#00FFFF**, **#FF00FF**.

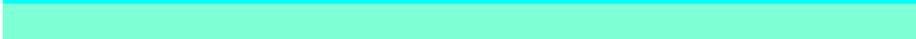
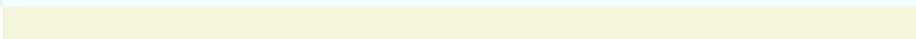
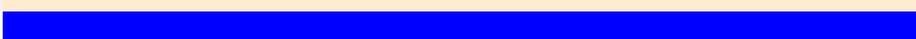
Infine bianco e nero si ottengono con **#FFFFFF** e **#000000**.

I toni di grigio si possono ottenere con tutte le combinazioni di tre coppie uguali, da **#010101** a **#FEFEFE**

*NB. Esiste la possibilità di utilizzare una notazione a tre caratteri che permette di definire 256 colori: es. **#A61** che corrisponde a #AA6611*

Codifica dei colori in HTML

Per 140 colori sono stati inoltre attribuiti dei nomi specifici. Ne riportiamo alcuni a titolo esemplificativo:

Nome colore	HEX	Colore
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
...	...	

I nomi colore possono essere utilizzati liberamente al posto del codice colore.

Usare immagini come sfondo

Per inserire un'immagine come sfondo è sufficiente utilizzare la seguente sintassi:

```
<body background="imgSfondo.gif">
```

Per ora presupponiamo che l'immagine di sfondo si trovi nella stessa cartella della nostra pagina HTML, quindi inseriamo il nome del file immagine senza percorso.

NB: L'immagine di sfondo verrà ripetuta in orizzontale e in verticale. (Tramite i CSS si può tuttavia personalizzare la modalità di ripetizione in orizzontale e in verticale).

È anche possibile combinare l'attributo background con quello bgcolor, in modo che mentre l'immagine di sfondo viene caricata, venga comunque visualizzata una colorazione della pagina:

```
<body bgcolor="#0000ff" background="imgSfondo.gif">
```

È utile assegnare sempre un colore alla pagina anche quando lo sfondo della pagina è bianco (al massimo assegnare bgcolor="#FFFFFF"). Infatti, come impostazione predefinita, il browser assegna alla pagina il colore di sfondo che l'utente ha impostato nella finestra del sistema operativo: quindi se l'utente ha impostato uno sfondo nero e voi non avete assegnato nessun colore di sfondo alla pagina, la vostra pagina sarà nera.

Dal punto di vista estetico, l'utilizzo di immagini come sfondo va ponderata accuratamente per evitare di creare pagine eccessivamente cariche, poco gradevoli e di rendere difficile la lettura dei contenuti testuali.

Un esempio introduttivo sui CSS. Gli stili per il BODY

Tutti gli attributi del body che abbiamo visto finora (da eccezione dell'attributo "lang") sono caratteristiche che riguardano il layout della nostra pagina HTML. Come si può vedere, con una sintassi di questo genere:

```
<body leftmargin="0" topmargin="0" background="sfondo1.gif" bgcolor="#66CCFF" lang="it">  
...  
</body>
```

il layout e il contenuto sono mischiati tra loro. Gli attributi "background" e "bgcolor" sono, come si diceva, deprecati nelle specifiche del W3C. In un approccio di impaginazione che utilizzi i fogli di stile, l'aspetto che riguarda la visualizzazione deve essere separato dal contenuto.

Il nostro body deve quindi essere ripulito degli attributi di stile per diventare:

```
<body lang="it">
```

mentre **le regole** che indicano come visualizzare lo sfondo **vanno inserite altrove** nel documento.

Un esempio introduttivo sui CSS. Immagini si sfondo

Le regole di utilizzo dei fogli di stile (i CSS) verranno trattate nelle prossime lezioni, ma vediamo una breve anticipazione:

I fogli di Stile sono estremamente potenti e offrono una grande versatilità sia nella formattazione dei contenuti sia nella loro disposizione nella pagina. Con riferimento alla slide precedente, osserviamo che per applicare l'immagine di sfondo sia possibile utilizzare la seguente sintassi:

```
<body style="background-image: url(sfondo.gif)">
```

si noti l'utilizzo dell'**attributo STYLE** che contiene la regola per l'applicazione dello sfondo.

Nell'attributo STYLE possono essere inserite più regole separate da un punto e virgola. Ad esempio vediamo come è possibile impedire la ripetizione dell'immagine:

```
<div style="background-image:url(sfondo.gif); background-repeat:no-repeat">
```

oppure mantenere «fissa» l'immagine di sfondo anziché permettere che scorra con lo scroll della pagina:

```
<div style="background-image:url(sfondo.gif); background-attachment:fixed">
```

Colore di base per il testo

Se non si imposta nessun colore per il testo, esso viene di default reso in nero.

Abbiamo però la possibilità di assegnare un colore per il testo di tutta la pagina, semplicemente utilizzando l'attributo TEXT del tag body:

```
<body text="blue">
```

Tuttavia **il nero non sempre è leggibile con tutti i colori di sfondo**. Tipicamente i siti sono classificati in due tipologie: testo chiaro su fondo scuro oppure testo scuro su fondo chiaro.

Se ad esempio vogliamo utilizzare uno sfondo blu è necessario utilizzare un colore chiaro per il testo, altrimenti si corre il rischio di rendere illeggibile il contenuto.

Questo esempio:

```
<body bgcolor="#0000ff" text="#ffffff">
```

produce una pagina con fondo blu e testo bianco.

Colore di base per i link

I link sono i collegamenti ipertestuali e per questo sono elementi «sensibili» alle operazioni dell'utente. Assumono quindi diversi «stati», per ognuno dei quali è possibile stabilire alcune proprietà, in primis il colore:

stato	attributo HTML 4.1	descrizione
normale	Link	Normalmente il link quando si trova «a riposo» deve essere evidenziato in modo tale che sia facile per l'utente individuarlo. Nell'HTML standard il link è sempre sottolineato (è possibile eliminare la sottolineatura soltanto usando i CSS) e di colore blu (#0000FF).
visitato	visited	Un link è visitato, quando l'URL della pagina compare nella cronologia dell'utente. Di default i link visitati sono di color violetto (#800080).
attivo	active	Il collegamento è attivo nel momento in cui il link è stato cliccato e sta avvenendo il passaggio da una pagina all'altra. Non si tratta di una caratteristica particolarmente utile oggi, ma quando i modem avevano una velocità molto inferiore a quella odierna, vedere un link "attivo" era comunque un'indicazione sul fatto che qualcosa stava avvenendo. Un'ulteriore condizione in cui un link si rileva "attivo" è quando si utilizza il tasto destro del mouse su di lui. Potremmo dire che un link è attivo quando «ha il focus».
mouse over	(solo CSS: hover)	Con l'HTML 4.01 al passaggio del mouse sul link si può fare ben poco, coi fogli di stile invece è possibile creare qualche effetto di visualizzazione.

La sintassi HTML 4.1 è dunque: **<body link="red" alink="yellow" vlink="green">**

Titoli

Nulla ci vieta di scrivere direttamente all'interno del tag body, come già abbiamo visto negli esempi precedenti, senza utilizzare nessun altro tag.

E' tuttavia più corretto e versatile racchiudere il testo in appositi tag a seconda della funzione che il testo deve svolgere. La pagina risulterà così più semplice da leggere, da modificare e formattare.

Il primo gruppo di tag orientato a delimitare i testi è quello dei **titoli**.

- <h1>titolo 1</h1>** Titolo di primo livello
- <h2>titolo 2</h2>** Titolo di secondo livello
- <h3>titolo 3</h3>** Titolo di terzo livello
- <h4>titolo 4</h4>** Titolo di quarto livello
- <h5>titolo 5</h5>** Titolo di quinto livello
- <h6>titolo 6</h6>** Titolo di sesto livello

"H" sta per "heading", cioè titolo: le grandezze previste sono sei. **Dall'<h1>, che è il più importante, si va via via degradando fino all' <h6>** e il testo è formattato in **grassetto**.

L'heading occupa tutta la larghezza dell'elemento in cui è contenuto e lascia una riga vuota prima e dopo di sé in quanto è uno degli **elementi «di blocco»**.

Paragrafi

Il secondo tag finalizzato a delimitare i testi è quello del **paragrafo**.

Il paragrafo è l'unità di base entro cui suddividere un testo. La sintassi è la seguente:

```
<p>...</p>
```

Anche il paragrafo occupa tutta la larghezza dell'elemento in cui è contenuto. (ad esempio, se il paragrafo è contenuto nel body occuperà tutta la larghezza della pagina).

Il paragrafo ha l'attributo ALIGN che serve per definire l'allineamento nel testo:

```
<p align="center">...</p>
```

L'attributo ALIGN può assumere 4 valori: «left», «right», «center», «justify».

Anche il tag <P> lascia una riga vuota prima della sua apertura e dopo la sua chiusura essendo elemento «di blocco».

Blocchi di testo DIV

Il terzo tag utilizzato per delimitare i testi è il blocco di testo, ovvero il **DIV**.

Il DIV va a capo ma, a differenza del paragrafo, non lascia spazi prima e dopo la sua apertura. E' sempre un elemento «di blocco». La sintassi è la seguente:

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

Elemento «di blocco» per eccellenza, il DIV è ampiamente utilizzato per creare il layout della pagina soprattutto utilizzando i fogli di stile.

In HTML 4.01 possiede anch'esso l'attributo ALIGN come il paragrafo:

```
<div align="center">...</div>
```

Contenitori

Il quarto e ultimo tag utilizzato per delimitare i testi è il contenitore **SPAN**.

La sintassi è la seguente:

...

Lo SPAN è un contenitore generico che può essere annidato in altri elementi come i paragrafi o i titoli.

SPAN è l'**elemento «inline»** per eccellenza. «Inline» è il contrario di «di blocco», ovvero indica un tag che non manda a capo il testo ma lo fa continuare sulla stessa linea. Si usa in modo particolare per formattare piccole porzioni di testo all'interno dei blocchi più generali.

L'utilizzo di SPAN con i fogli di stile offre la maggiore versatilità.

Disposizione del testo

Le **differenze** tra <P>, <DIV> e sono dunque le seguenti:

- **P** lascia spazio prima e dopo la propria chiusura
- **DIV** non lascia spazio prima e dopo la propria chiusura ma, essendo un elemento di blocco, va a capo
- **SPAN**, essendo un elemento inline, non va a capo

Per quel che riguarda gli heading (<h1>, ..., </h6>) occorre ricordare che la grandezza del carattere è sempre relativa, ovvero varia a seconda delle impostazioni che l'utente ha sul proprio browser.

Tutti questi «tag-contenitori» permettono di allineare il testo utilizzando l'attributo **ALIGN**, tuttavia va ricordato che tale attributo **è disapprovato dal W3C**, dal momento che per allineare il testo bisognerebbe invece utilizzare i fogli di stile. E' stato definitivamente eliminato con l'HTML 5 ma è comunque ancora supportato da tutti i browser e ampiamente utilizzato.

Disposizione del testo

Per **andare a capo** molti webmaster utilizzano l'apertura arbitraria di paragrafi che non contengono nulla e che vengono lasciati aperti:

```
<p>  
<p>  
<p>
```

Si tratta in buona sostanza di un errore, visto che per andare a capo esiste un apposito tag, il BR:

```
<br />
```

BR sta per «break», cioè «interruzione». Per andare a capo è quindi sufficiente scrivere un BR. Per saltare una riga ne occorrono due:

```
<br /><br />
```

BR è un tag vuoto e quindi non ha l'elemento di chiusura. Come per gli altri tag vuoti, in osservanza all'XHTML va scritto con uno spazio e uno slash prima della seconda parentesi.

Disposizione del testo

Un ulteriore tag per dividere la pagina in parti è **HR** («horizontal rule») che serve per tracciare una linea orizzontale. Si tratta di un altro tag vuoto; la sintassi è la seguente:

```
<hr />
```

e produce un effetto simile al seguente:



Questo tag ha anche alcuni attributi (sempre deprecati, perché la formattazione andrebbe fatta con i CSS): L'attributo **NOSHADA** evita di sfumare la linea, **SIZE** indica l'altezza in pixel, **WIDTH** è la larghezza in pixel o in percentuale, **ALIGN** l'allineamento.

```
<hr noshade size="5" width="50%" align="center" />
```

Il risultato è simile al seguente:



HR è ormai quasi inutilizzato, sia per l'effetto grafico non propriamente gradevole, sia per la più ampia versatilità che si può ottenere da un lato con i fogli di stile, dall'altro con le grafiche raster (GIF, PNG ecc).

Gli stili fondamentali

Formattazione di base del testo in una pagina HTML

Stili fisici e stili logici

Tipograficamente attribuire lo **stile ad un testo** un testo significa definire la variante del carattere (tondo, corsivo, grassetto ecc).

Nell' HTML invece possiamo dire che gli stili possono avere sia la finalità di assegnare caratteristiche grafiche, sia quella di attribuire un ruolo al TAG (che conseguentemente verrà rappresentato con una grafica appropriata)

La distinzione è specificata a livello di documentazione e letteratura ma è piuttosto sottile. Sostanzialmente, in taluni casi avremo stili che determineranno aspetti grafici indipendentemente dalla funzione del tag, in altri stili che determineranno più specificatamente uno specifico «ruolo» per il tag.

Il linea generale è sufficiente memorizzare l'effetto prodotto dai diversi stili indipendentemente dal ruolo che questi attribuiscono al tag.

Grassetto

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code>...</code>	Questo <code>testo</code> è in grassetto	Questo testo è in grassetto	Grassetto

Lo stile "grassetto" dovrebbe essere ottenuto con i fogli di stile (così come tutte le formattazioni), ma evidentemente la possibilità di ottenere un testo in grassetto semplicemente scrivendo "`testo`". E' di fatto ancora talmente comoda da non essere ancora caduta in disuso.

Corsivo

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code><i>...</i></code>	Questo <code><i>testo</i></code> è in corsivo	Questo <i>testo</i> è in corsivo	Corsivo

Attenzione a non abusare del corsivo perché la leggibilità del corsivo nei browser è generalmente inferiore rispetto al carattere normale.

Sottolineato

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code><u>...</u></code>	Questo <code><i>testo</i></code> è sottolineato	Questo <u>testo</u> è sottolineato	Sottolineato

Nel web le sottolineature del testo sono da evitare, per non confondere il lettore con i link. Un testo sottolineato indica infatti la presenza di un collegamento ipertestuale.

Barrato

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code><strike>...</strike></code>	Questa <code><strike>parola</strike></code> è sbagliata	Questa parola è sbagliata	Barrato

Utilizzato, ad esempio, per l'inserimento di «Correzioni» in un testo.

Preformattato

Molti tag tradiscono l'origine scientifica e informatica del Web (sono presenti tag per blocchi di codice di programmazione, per definizioni, per l'indicazione delle variabili ecc). E' il caso di PRE, utile per riportare frammenti di codice di programmazione nel testo della pagina, e gli APICI/PEDICI.

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code><pre>...</pre></code>	Codice PHP :<pre>PHP_FUNCTION {zval **parameters; zval *value; char* str;</pre>	Codice PHP: PHP_FUNCTION { zval **parameters; zval *value; char* str;	Preformattato

Apice / Pedice

TAG	SINTASSI	VISUALIZZAZIONE	DESCRIZIONE
<code><sup>...</sup></code>	$E=mc^{2}$	$E=mc^2$	Apice
<code><sub>...</sub></code>	Acqua= H_2O	Acqua= H_2O	Pedice

Font del testo

Il tipo di carattere (cioè il "font") che il browser visualizza di default è il «Times», ottimo per la carta stampata poiché dotato di "grazie" (abbellimenti tipografici delle lettere, che aiutano per la leggibilità), non altrettanto leggibile sullo schermo (anzi, si ottiene l'effetto contrario).

Si preferisce utilizzare dei caratteri senza grazie, ad esempio:

TIMES
Times

VERDANA
Verdana

ARIAL
Arial

HELVETICA
Helvetica

Dimensioni, colore, tipo di carattere si possono assegnare con il tag FONT

...

ATTENZIONE: il tag è disapprovato dal W3C.

La formattazione del testo in tutti i siti moderni viene effettuata per mezzo dei fogli di stile CSS.

Font del testo

Il primo attributo di FONT serve per attribuire il tipo di carattere, ed è FACE:

TAG	SINTASSI	VISUALIZZAZIONE
	<code>Testo in Arial</code>	Testo in Arial
<code>...</code>	<code>Testo in Times</code>	Testo in Times
	<code>Testo in Comic</code>	Testo in Comic

ATTENZIONE: In HTML non è possibile utilizzare dei font speciali, ovvero font che non siano quelli presenti nel sistema operativo degli utenti. Ci sono tuttavia molti font che sono presenti di default su tutti i sistemi più diffusi. *(NB – con i fogli di stile è possibile invece applicare anche font diversi da quelli installati sul client)*

Per questo motivo è bene tener conto di due accorgimenti:

- **scegliere font «sicuri»**, che siano cioè senz'altro presenti sul pc dell'utente;
- **non indicare un solo font**, ma una serie di font che gradualmente si allontanano dal risultato che vorremmo ottenere, ma non di molto, fino ad indicare la famiglia a cui il nostra carattere appartiene.

Indicando la famiglia il browser dell'utente cercherà di trovare nella propria cartella dei fonts il primo carattere indicato, se non lo trova passerà al secondo, e solo come ultima scelta sceglierà di utilizzare il carattere predefinito.

Colore del testo

Il secondo attributo di FONT serve per attribuire il colore al carattere, ed è ovviamente COLOR:

TAG	SINTASSI	VISUALIZZAZIONE
<code>...</code>	<code>Testo in blu</code>	Testo in blu
	<code>Testo in rosso</code>	Testo in rosso

Nell'indicazione del colore è possibile utilizzare sia i codici mnemonici sia quelli esadecimali.

Dimensione del testo

Il terzo attributo di FONT serve infine per attribuire la dimensione del carattere, ed è SIZE.

In HTML si distinguono due fondamentali modalità di attribuzione della dimensione:

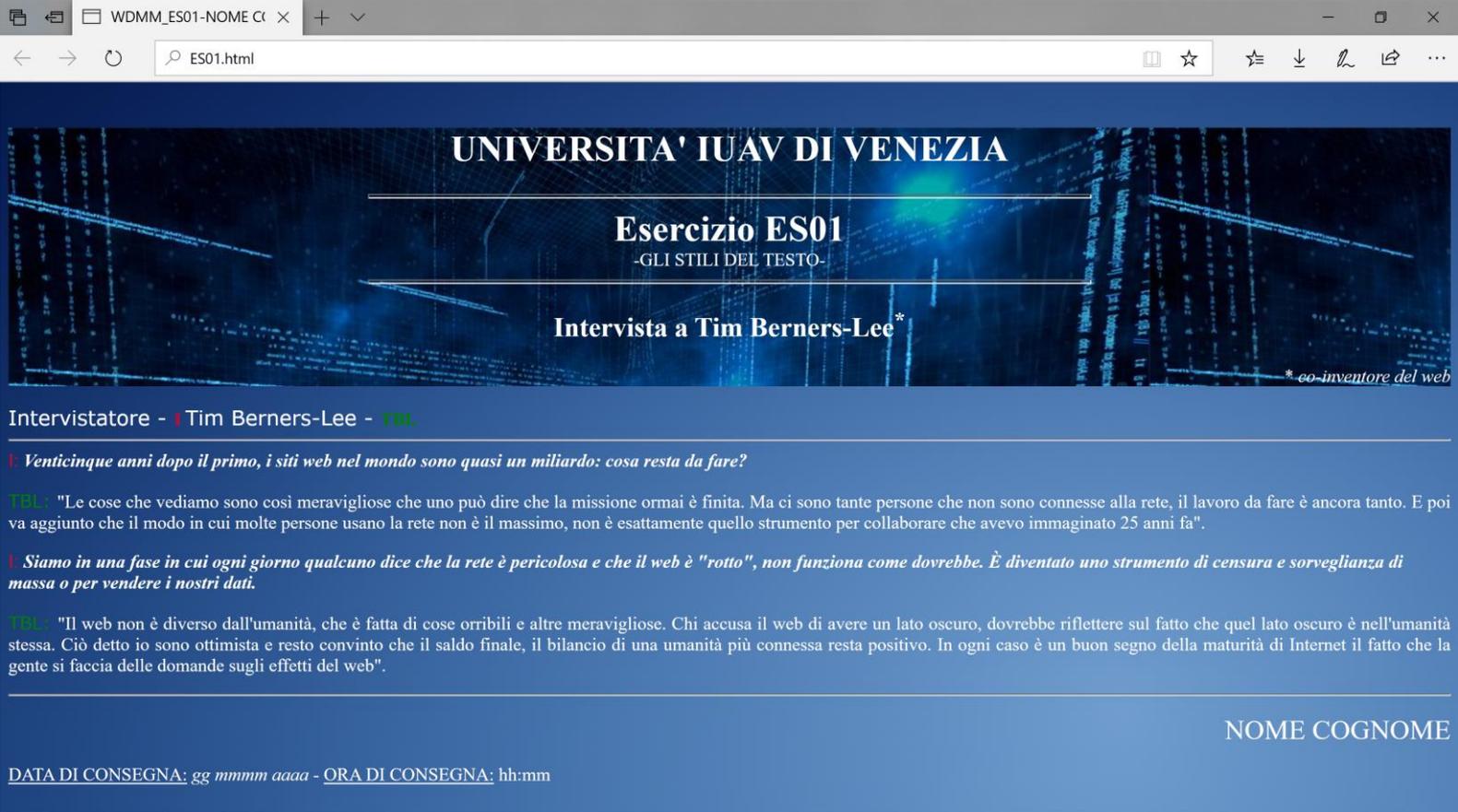
- Valori **assoluti** (con numeri interi da 1 a 7)
- Valori **relativi** (relativi alla dimensione che il testo assume nella sua specifica posizione)

TAG	SINTASSI	VISUALIZZAZIONE
...	<code>Testo dim.2</code>	Testo dim.2
	<code>Testo dim.5</code>	Testo dim.5
	<code>Testo dim.7</code>	Testo dim.7
	<code>Testo meno 1</code>	Testo meno 1
	<code>Testo più 2</code>	Testo più 2
	<code>Testo più 4</code>	Testo più 4

NB: i valori relativi agiscono sulla dimensione assunta dal testo nella specifica posizione. Es se applico un +2 all'interno di un tag che ha dimensione 5 ottengo una dimensione di 7, non di 5.

Esercizio n.1

Realizzare una pagina HTML con il seguente contenuto e aspetto



WDMM_ES01-NOME CC X + v

ES01.html

UNIVERSITA' IUAV DI VENEZIA

Esercizio ES01

-GLI STILI DEL TESTO-

Intervista a Tim Berners-Lee*

* *co-inventore del web*

Intervistatore - **I** Tim Berners-Lee - **TBL**

I *Venticinque anni dopo il primo, i siti web nel mondo sono quasi un miliardo: cosa resta da fare?*

TBL: "Le cose che vediamo sono così meravigliose che uno può dire che la missione ormai è finita. Ma ci sono tante persone che non sono connesse alla rete, il lavoro da fare è ancora tanto. E poi va aggiunto che il modo in cui molte persone usano la rete non è il massimo, non è esattamente quello strumento per collaborare che avevo immaginato 25 anni fa".

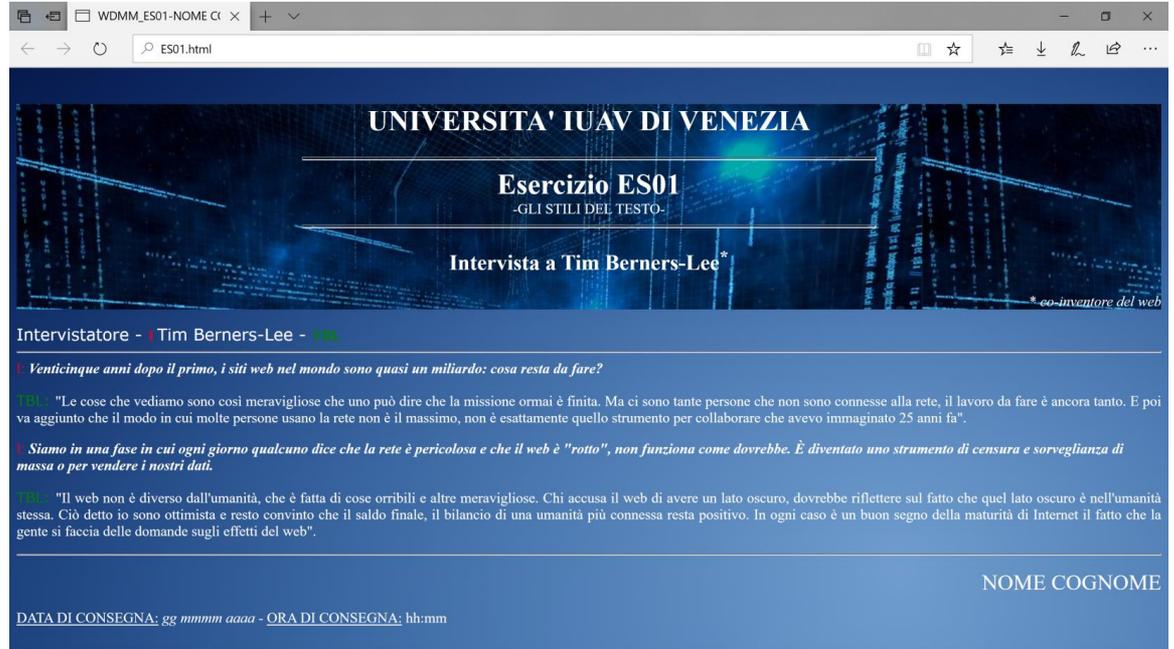
I *Siamo in una fase in cui ogni giorno qualcuno dice che la rete è pericolosa e che il web è "rotto", non funziona come dovrebbe. È diventato uno strumento di censura e sorveglianza di massa o per vendere i nostri dati.*

TBL: "Il web non è diverso dall'umanità, che è fatta di cose orribili e altre meravigliose. Chi accusa il web di avere un lato oscuro, dovrebbe riflettere sul fatto che quel lato oscuro è nell'umanità stessa. Ciò detto io sono ottimista e resto convinto che il saldo finale, il bilancio di una umanità più connessa resta positivo. In ogni caso è un buon segno della maturità di Internet il fatto che la gente si faccia delle domande sugli effetti del web".

NOME COGNOME

DATA DI CONSEGNA: gg mmmm aaaa - ORA DI CONSEGNA: hh:mm

Link materiali



WDMM_ES01-NOME CO X

ES01.html

UNIVERSITA' IUAV DI VENEZIA

Esercizio ES01
-GLI STILI DEL TESTO-

Intervista a Tim Berners-Lee*

*co-inventore del web

Intervistatore - Tim Berners-Lee - TBL

! *Venticinque anni dopo il primo, i siti web nel mondo sono quasi un miliardo: cosa resta da fare?*

TBL: "Le cose che vediamo sono così meravigliose che uno può dire che la missione ormai è finita. Ma ci sono tante persone che non sono connesse alla rete, il lavoro da fare è ancora tanto. E poi va aggiunto che il modo in cui molte persone usano la rete non è il massimo, non è esattamente quello strumento per collaborare che avevo immaginato 25 anni fa".

! *Siamo in una fase in cui ogni giorno qualcuno dice che la rete è pericolosa e che il web è "rotto", non funziona come dovrebbe. È diventato uno strumento di censura e sorveglianza di massa o per vendere i nostri dati.*

TBL: "Il web non è diverso dall'umanità, che è fatta di cose orribili e altre meravigliose. Chi accusa il web di avere un lato oscuro, dovrebbe riflettere sul fatto che quel lato oscuro è nell'umanità stessa. Ciò detto io sono ottimista e resto convinto che il saldo finale, il bilancio di una umanità più connessa resta positivo. In ogni caso è un buon segno della maturità di Internet il fatto che la gente si faccia delle domande sugli effetti del web".

NOME COGNOME

DATA DI CONSEGNA: gg mmmm aaaa - ORA DI CONSEGNA: hh:mm

Immagini di sfondo:

<http://www.borga.it/main/download.aspx?file=176>

<http://www.borga.it/main/download.aspx?file=177>