

# **Webdesign**

## **dalla teoria alla pratica**

**MODULO\_1: tipografia e HTML**



# programma

- parte prima -

- **Principi di tipografia**
  - Tipografia applicata: i testi e gli oggetti tipografici
  - Il parallelo digitale: gli ipertesti
- **Strutturare i dati: i linguaggi di markup**
  - HTML (HyperText Markup Language)
  - L'accessibilità dei contenuti
  - Commenti, indentazione e leggibilità del codice



# **principi di tipografia**



# Principi di tipografia

L'oggetto di questa parte di lezione sono i LIBRI.

Grossomodo i libri sono fatti di flusso di testi; questi sono composti da alcuni oggetti base che ripercorriamo:

- Font (stili, famiglie e comportamenti)
- Titoli
- Paragrafi
- Liste (ordinate, non ordinate)
- Tabelle, foto o figure

la terminologia usata può non essere ortodossa in tipografia ma ha un senso per il parallelo con gli elementi HTML



# Flusso di testo indistinto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetur. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetur erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra



# Font

## Famiglie

- Serif

Esempio di font serif

- Sans Serif

Esempio di font sans-serif

- Mono-Spaced

Esempio di font sans-serif



# Font

## Stili

- Plain

Esempio di plain

- Bold

**Esempio di bold**

- Italic

*Esempio di italic*

- Bold Italic

***Esempio di bold italic***



# Font

## Stili

- Underline  
Esempio di underline
- Small caps  
ESEMPIO DI SMALL CAPS
- Superscript, Subscript (vd. pdf)

## Spaziature di carattere

## Spaziature di linea



# Titoli

- Annunciano il contenuto del blocco di testo successivo
- Hanno l'utilità di separare un flusso indistinto di contenuti
- Dichiarano una gerarchia innestata di contenuti e blocchi di testo



# Titoli

## Character and Word Spacing

Many times, simply typing in the text and formatting the font, size, and line spacing is enough. However, depending on the program used, extra attention is needed. Larger type sizes need adjustments to the space between characters; and paragraphs need to be adjusted to eliminate “widows” and “orphans.”

### Kerning

Inter-character spacing, known as *kerning*, creates a more pleasing look to the text. Most word processors do not allow kerning adjustments and most page-layout programs apply kerning automatically; however, certain letter combinations may require manual adjustments. Some of these letter combinations include most lowercase letters, and uppercase letters: F, I, K, L, O, P, Q, T, V, W, X, and Y. Kerning also is required when using all uppercase letter combinations such as VA and WA.



# Paragrafi



- Separano i blocchi di testo fra loro
- Aiutano la lettura e guidano l'occhio
- Raggruppano argomenti simili
- Possono avere un interlinea o del margine fra di loro



# Paragrafi

## Lorem ipsum dolor sit amet

consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. **Aliquam erat volutpat.** Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor.

margine di paragrafo

## Lorem ipsum dolor sit amet

consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor.

## Lorem ipsum dolor sit amet

interlinea (line spacing)

consectetuer adipiscing elit. Morbi commodo, ipsum<sup>sed</sup> pharetra<sub>gravida</sub>, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla



# Liste

Una lista serve per ordinare dei punti. In queste slide ne uso a iosa. :)

Liste non ordinate: servono per elencare punti senza particolare enfasi sull'ordine in cui lo si fa

- esempio di
- lista
- non
- ordinata

Liste ordinate: servono per elencare punti con particolare enfasi sull'ordine in cui lo si fa

1. esempio di
2. lista
3. non
4. ordinata



# Tabelle, foto e figure

L'oggetto tabella è necessario solo quando è necessario mostrare dei dati tabellari, appunto.

Employee	Division	Suggestions	Rating
Stephen C. Cox	Marketing	Make discount offers	3/10
Josephin Tan	Advertising	Give bonuses	5/10
Joyce Ming	Marketing	New designs	8/10
James A. Pentel	Marketing	Better Packaging	8/10
Give background color to the table cells to achieve seamless transition			

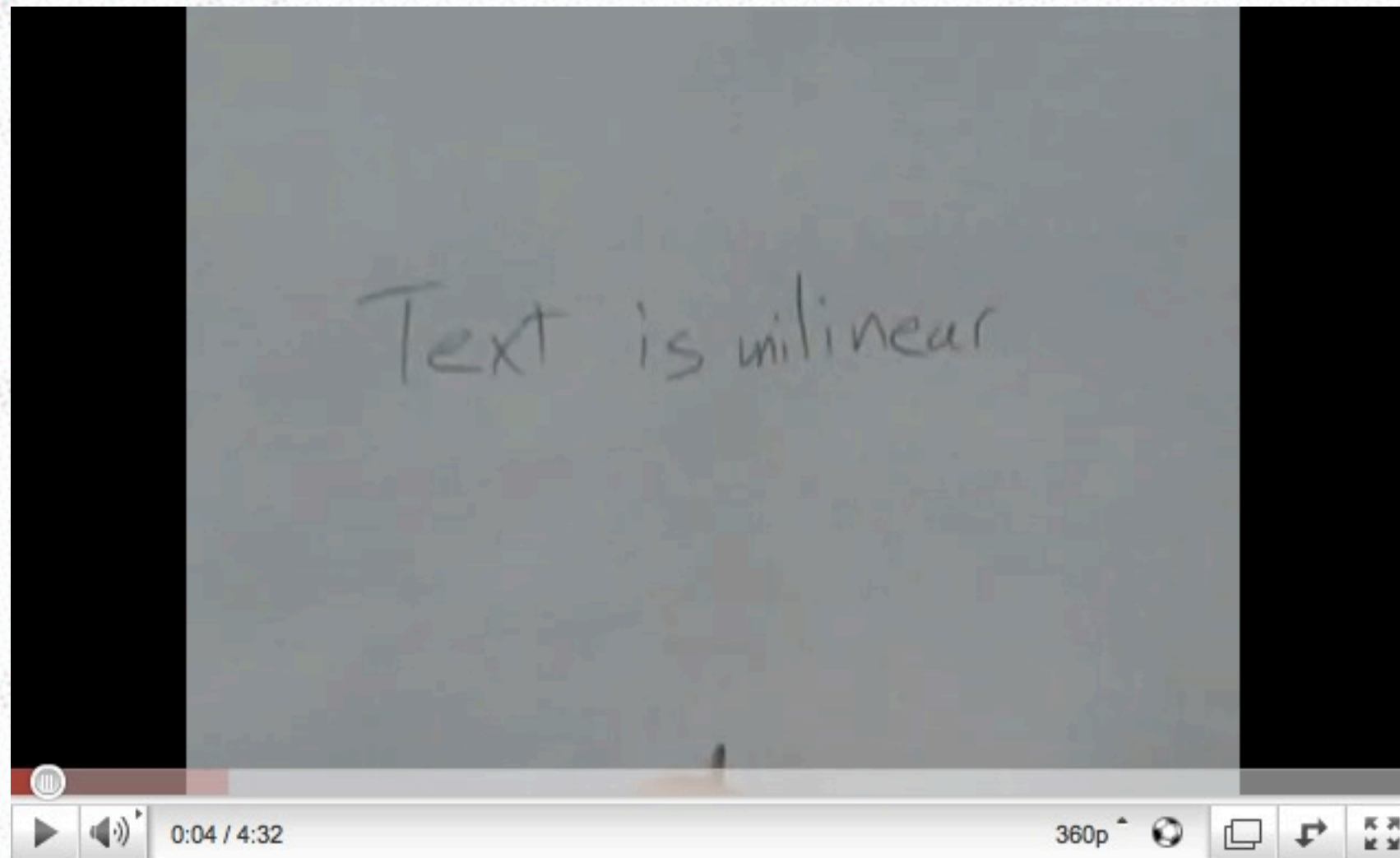
Le immagini o figure possono seguire il flusso del testo o interromperlo. È buona pratica dare sempre “aria” ad esse per lasciare all'utente la sensazione di cura.



# **il parallelo digitale: ipertesti**



# dal testo all'ipertesto



<http://www.youtube.com/watch?v=6gmP4nk0EOE>



# L'ipertesto diventa codice

Ognuno degli oggetti tipografici che abbiamo visto ha un parallelo in HTML.

Ne faremo una brevissima carrellata.



# Paralleli

<b>Font</b>	→	font-family (CSS)
<b>Titoli</b>	→	<h1>, <h2>, ..., <h6>
<b>Paragrafi</b>	→	<p>
<b>Liste</b> (ordinate, non ordinate)	→	<ul>, <ol>, <dl>
<b>Tabelle, foto, figure</b>	→	<table>, <img>



# La novità degli ipertesti: i collegamenti

Un ipertesto è un insieme di documenti messi in relazione tra loro tramite parole chiave. Può essere visto come una rete; i documenti ne costituiscono i nodi. La caratteristica principale di un ipertesto è che la lettura può svolgersi in maniera non lineare: qualsiasi documento della rete può essere "il successivo", in base alla scelta del lettore di quale parola chiave usare come collegamento.

È possibile, infatti, leggere all'interno di un ipertesto tutti i documenti collegati dalla medesima parola chiave. La scelta di una parola chiave diversa porta all'apertura di un documento diverso: all'interno dell'ipertesto sono possibili praticamente infiniti percorsi di lettura. (wikipedia)

A differenza dei testi scritti a mano o stampati, gli ipertesti hanno come caratteristica principale quella di poter richiamare altri ipertesti tramite il meccanismo di collegamento ipertestuale.

Questo meccanismo è il cuore del funzionamento della rete internet e delle reti in generale.



**strutturare i dati:  
i linguaggi di markup**



# Markup Language (wikipedia)

A markup language is a modern system for annotating a text in a way that is syntactically distinguishable from that text. The idea and terminology evolved from the "marking up" of manuscripts, i.e. the revision instructions by editors, traditionally written with a blue pencil on authors' manuscripts.

[...]

Markup is typically omitted from the version of the text which is displayed for end-user consumption. Some markup languages, like HTML have presentation semantics, meaning their specification prescribes how the structured data is to be presented, but other markup languages, like XML, have no predefined semantics. (wikipedia)



# HyperText Markup Language

HTML è un linguaggio semantico e descrittivo delle pagine web, formalizzato a metà anni '80 da Tim Berners-lee e ha visto la sua esplosione a metà degli anni '90.

Non è un linguaggio di programmazione ed è improprio parlare di programmazione di pagine HTML.

Descrive, invece, la struttura e il contenuto di documenti ipertestuali che possono essere visualizzati (parsati) da software specifici (tipicamente browser internet come Mozilla Firefox).

Un parallelo reale di un linguaggio di Markup può essere quello delle correzioni di una maestra su un compito di un alunno.



# Documenti HTML

Gli ipertesti descritti con il linguaggio HTML sono tipicamente dei **file** (di estensione html, htm o shtml), che risiedono su un computer o un server (un computer che è in grado di *servire* una richiesta).

La *chiamata* a un file HTML da un browser restituisce lato utente il contenuto dello stesso con la relativa forma data dalla descrizione HTML.

Es.: demo/1.html



# HTML: tag

In HTML il contenuto è *generalmente* compreso tra una coppia di parole chiave proprie del linguaggio, dette **tag**, che aggiungono le informazioni descrittive necessarie.

ES:

**<title>questo è il titolo di pagina</title>**

in questo esempio il TAG è <title>

**<p>questo è un paragrafo</p>**

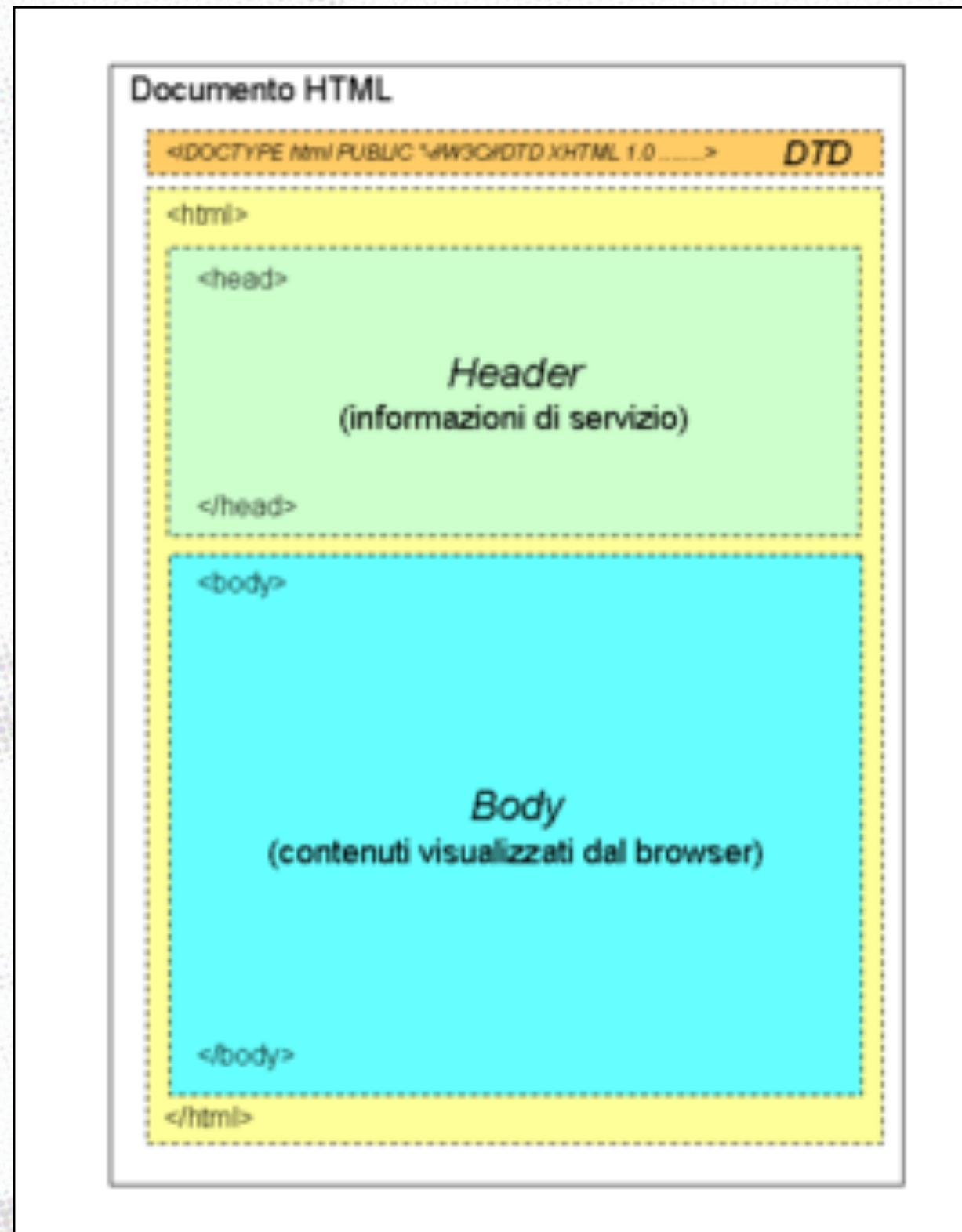
in questo esempio il TAG è <p>

**<a href="questo è un attributo" title="questo è un altro attributo">questo è il contenuto di un ancora</a>**

in questo esempio il TAG <a> con due attributi: title e href.



# HTML: struttura





# HTML: il doctype

Definisce gli elementi leciti all'interno del documento. Non si possono usare altri elementi se non quelli definiti. Una specie di "vocabolario" per i file che lo useranno.

Nei casi di interesse per noi i doctype saranno di tre tipi al massimo:

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`



# HTML: la head

La sezione `<head>` di una pagina HTML contiene informazioni utili dal punto di vista del browser, e della descrizione del documento, ma non vengono visualizzate propriamente dallo stesso a favore dell'utente.

ECCEZIONI:

- `<style>`,  
`<script>`  
style e script, pur non visualizzando direttamente nulla sul browser, possono influire fortemente sulle modalità di visualizzazione delle pagine HTML (lezione di domani)
- `<title>`  
il contenuto del tag `<title>` infatti viene visualizzato dal browser come titolo della pagina



# HTML: <head>

Un altro TAG notevole per la scrittura di pagine HTML nel modo corretto è <meta>.

Esso permette di inserire dati non visibili all'utente ma utili alle macchine che (tramite browser) possono trovarli utili.

<meta>

```
<meta name="description" content="Free Web tutorials" />
```

```
<meta name="keywords" content="HTML,CSS,XML,JavaScript" />
```

```
<meta name="author" content="John Doe" />
```

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



# HTML: <meta> e SEO

I motori di ricerca (Google, Yahoo, e altri) periodicamente scandagliano la rete con dei software molto simili a un browser (spiders vd. Matrix).

Sono software automatizzati qui non sono in grado di riprodurre immagini, video, colori o stili particolari.

Per questo in un buon codice HTML l'attenzione a regole elementari di **accessibilità** ci permettono di avere delle pagine che sono già SEO friendly.



# HTML: il body

Questa è la sezione più interessante per l'utente.

In questa viene incluso tutto il contenuto **e la sua relativa descrizione** che va visualizzato da un browser.

Poiché si tratta di un sistema per la visualizzazione di ipertesti, la sua apparenza di base è molto simile ai principi di tipografia che abbiamo visto prima.



# HTML: i tag principali

`<h1>`, `<h2>`, ..., `<h6>` definiscono i titoli

`<p>` racchiudono un paragrafo

`<br>`, `<br />` è un a capo automatico

`<strong>` rappresenta il grassetto

`<em>` rappresenta il corsivo o enfaticizzato

`<ul>`, `<ol>` definiscono le liste rispettivamente non ordinate e ordinate

`<li>` rappresenta l'elemento di lista

`<a>` è l'elemento di collegamento interno o ipertestuale

`<img>`, `<img />` è necessario per mostrare delle immagini collegate

`<table>` si usa esclusivamente per rendere dati tabellari (approfondimenti successivi)

rif: <http://www.w3schools.com/tags/>



# <h1>, ..., <h6>

Come nei libri e nei testi formattati possono essere utilizzati per annunciare il contenuto successivo (tipicamente un paragrafo o una lista o una tabella).

**Buona norma:** in una pagina HTML l'<h1> riprende il <title> per una migliore indicizzazione da parte dei motori di ricerca.

Inoltre è buona norma usare un solo <h1> per pagina (pro memo: i documenti hanno in genere un solo titolo e alcuni sottotitoli. Qui è lo stesso).

**es:** <h2>questo è un titolo di secondo livello</h2>



# <p>, <br>, <strong>, <em>

Mentre <p> descrive al browser che il contenuto sarà un paragrafo, gli altri avvertono lo stesso, e quindi l'utente dell'interruzione di riga <br>, di un contenuto importante <strong>, o enfaticizzato <em>.

N.B.: è sempre preferibile usare <strong> e <em> rispetto a <b> e <i> (bold e italic) perché questi ultimi due descrivono delle proprietà meramente estetiche del contenuto e non delle proprietà semantiche.

**es:** <p>questo paragrafo contiene una riga<br>interrotta e una parola molto <strong>importante</strong> e una molto <em>enfaticizzata</em></p>



# <ul>, <ol>, <li>

Le liste sono estremamente importanti in HTML.

Descrivono ovviamente delle liste di oggetti, ma sono continuamente utilizzate per descrivere contenuti di menu.

La loro natura permette di leggere come un unico oggetto tutto il contenuto.

**es: (vd. demo/2.html)**

```
<ul><li>questo è un elemento di lista non ordinata</li><li>questo è un altro elemento di lista non ordinata</li></ul>
```

```
<ol><li>questo è un elemento di lista ordinata</li><li>questo è un altro elemento di lista ordinata</li></ol>
```



# <a>

È probabilmente la ragione del successo di internet.

<a> sta per ancora e permette di *ancorare* il contenuto del tag con un elemento all'interno dell'attributo href.

**<a href="/path/to/file.est">parola</a>** da linkare.

*best practices* è buona norma evitare di includere dentro una <a> molte parole. Risulta fuorviante per l'utente.

È ottima norma aggiungere sempre l'attributo **title** nelle ancora. Permette agli utenti di sapere cosa è linkato.

Oltre a file esterni è possibile linkare parti del documento stesso con il tag <a>.

In particolare la sintassi è <a href="#nomeancora">ancora interna</a>, questa farà riferimento a un <a name="nomeancora"></a> come si vede nell'es. 3.html



# <img>

Assieme alle ancora rappresenta un pilastro del webdesign.

È il tag necessario per includere immagini all'interno di un flusso di documento.

Per questa ragione è buona norma utilizzarlo con coscienza dal punto di vista della visualizzazione e delle interfacce.

**es:** ``



# accessibilità dei contenuti

Il flusso di codice HTML può essere visualizzato sia con browser lato utente normodotato (Mozilla Firefox, Google Chrome, Safari, IE), o con altri dispositivi assistivi per utenti ipodotati.

Per questo la scrittura di un documento descrittivo come l'HTML non dovrebbe **mai** prescindere dai vari possibili usi di questo.

L'uso comune e corretto degli attributi principali è **necessario** affinché il codice HTML sia portabile su software più comuni come su software meno comuni.

Ricordati **sempre** di aggiungere un `title=""` alle ancore.

Ricordati **sempre** di aggiungere un `alt=""` alle immagini.



# scrittura del codice, indentazione, commenti

Negli esempi precedenti il codice era spesso scritto senza troppi criteri.

MALE!

Il codice HTML non morirà con te!

Qualcun altro ci metterà le mani o tu lo erediterai da qualcun altro! Mettilo in condizione di riuscire a leggerlo.

Usa i commenti `<!-- questo è un commento -->` con saggezza e in maniera descrittiva per te e per gli altri.



# **Webdesign**

## **dalla teoria alla pratica**

**MODULO\_1: dalla struttura allo stile, laboratorio formativo**



# programma

- parte seconda -

- **I comportamenti di default**
  - ex cursus dei principali comportamenti visivi dei tag
  - Cascading Style Sheet
  - Selettori e proprietà
  - Prima di cominciare
- **Padroneggiare l'apparenza**
  - Cascading e gerarchie
  - Cambiare l'aspetto dei principali tag.
  - Commenti, indentazione e leggibilità del codice



# I comportamenti di default



# ex cursus dei comportamenti dei tag

**titolo 1**

**titolo 2**

**titolo 3**

**titolo 4**

**titolo 5**

**titolo 6**

Nunc auctor bibendum eros. Maecenas porta accumsan mauris. Etiam enim enim, elementum sed, bibendum quis, rhoncus non, metus. Fusce neque dolor, adipiscing sit amet, quam. Suspendisse wisi quam, consectetur in, blandit sed, suscipit eu, eros. Etiam ligula enim, tempor ut, blandit nec, mollis eu, lectus. Nam cursus. Vivamus purus, pharetra in, blandit quis, gravida a, turpis. Donec nisl. Aenean eget mi. Fusce mattis est id diam. Phasellus faucibus interdum sapien. Duis quis nunc. Sed e

[Link](#)

[Link](#)

**questo era grassetto**

*questo era corsivo*

dirigito gli colziao

dirigito gli elszecio

l'uf

l'uf



# Cascading Stylesheet

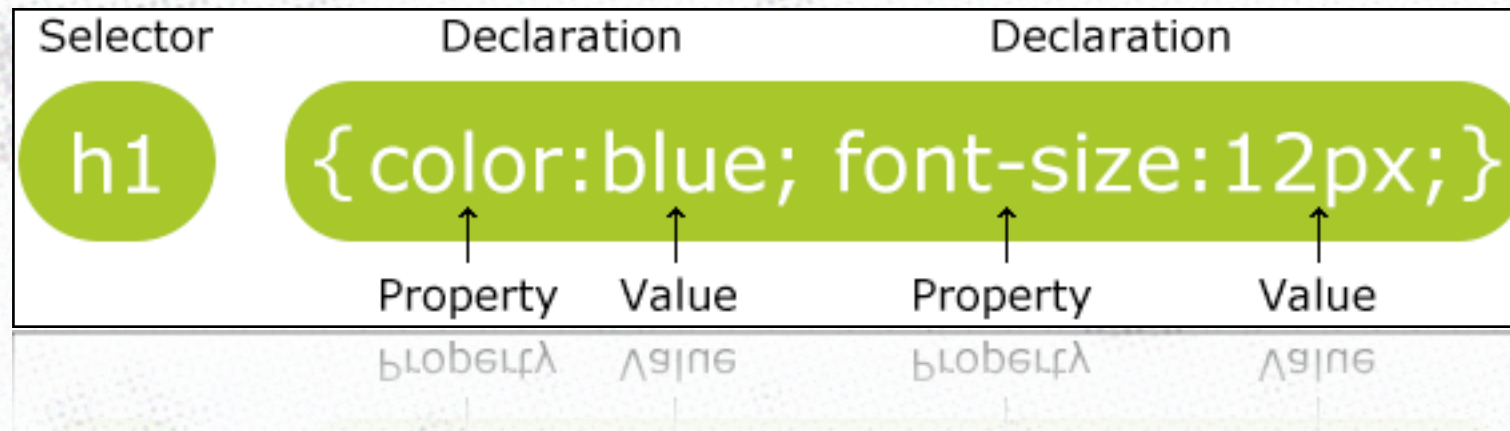
I fogli di stile a cascata, meglio noti con l'acronimo CSS (dall'inglese Cascading Style Sheet) e detti anche semplicemente fogli di stile, vengono usati per definire la rappresentazione di documenti HTML, XHTML e XML. Le regole per comporre i fogli di stile sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C. L'introduzione dei fogli di stile si è resa necessaria per separare i contenuti dalla formattazione e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine HTML che per gli utenti.



# Selettori e proprietà

Sulla falsa riga dell'HTML con TAG e attributi, i CSS lavorano con la triade

- **Selettore**
- **Proprietà**
- **Valore**





# Selettori e proprietà

Il principio del CSS è quello di cambiare l'aspetto delle entità HTML per piegarlo alle esigenze di presentazione senza intaccarne la sostanza e il contenuto.

Prima di continuare introduciamo due altri TAG html che sono fondamentali

`<div>` definisce una **divisione** o sezione di una pagina HTML

`<span>` è un espediente HTML per attribuire un comportamento visuale al testo HTML, senza intaccarne il senso né il contenuto.



# Selettori e proprietà

Le direttive CSS si possono applicare in vari modi

## **TAG**

**h1 { color:white; }**

il colore bianco si applica a TUTTI gli <h1> della pagina web

**p { color: red; }**

il colore rosso si applica a TUTTI gli <p> della pagina web



# Selettori e proprietà

## CLASSI

**.first\_title { color:white; }**

il colore bianco si applica a TUTTI i tag HTML della pagina web che hanno un attributo class="first\_title"

**.second\_title { color: red; }**

il colore rosso si applica a TUTTI i tag HTML della pagina web che hanno un attributo class="second\_title"



# Selettori e proprietà

## IDENTIFICATORI

**#unique\_id\_1 { color:white; }**

il colore bianco si applica a TUTTI i tag HTML della pagina web che hanno un attributo id="unique\_id\_1"

**#unique\_id\_2 { color: red; }**

il colore rosso si applica a TUTTI i tag HTML della pagina web che hanno un attributo id="unique\_id\_2"

**ATTENZIONE:** affinché una pagina sia valida secondo W3C xHTML 1.0 strict è **necessario** che gli id siano chiamati univocamente



# Includere un CSS in una pagina

Per includere un file CSS e poter godere delle direttive degli stessi, si hanno alcune opzioni; la posizione formalmente corretta di questo codice è all'interno di <head></head>

## Link del file esterno

```
<link rel="stylesheet" type="text/css" href="foglio_di_stile.css">
```

## import del file esterno

```
<style type="text/css">
```

```
@import url(foglio_di_stile.css);
```

```
</style>
```

## scrittura del css nella pagina

```
<style type="text/css">
```

```
/* put your css here */
```

```
</style>
```



# Resettiamo tutti i comportamenti default

Questa pratica, comunemente inclusa in un file **reset.css** è molto consigliata per avere da subito il pieno controllo del foglio di stile e per revocare i comportamenti di default di browser bizzarri (IE :))

Per questo nei nostri progetti includeremo sempre un file `reset.css`



**padroneggiare l'apparenza**



# Perché Cascading?

Il principio di funzionamento dei CSS è legato all'ordine in **cascata** con cui le direttive si applicano.

es.: 2.html, 3.html, 4.html



# scrittura del codice, indentazione, commenti

Negli esempi precedenti il codice era spesso scritto senza troppi criteri.

MALE!

Il codice CSS non morirà con te!

Qualcun altro ci metterà le mani o tu lo erediterai da qualcun altro! Mettilo in condizione di riuscire a leggerlo.

Usa i commenti `/* questo è un commento */` con saggezza e in maniera descrittiva per te e per gli altri.

es: 6.css



# Definizione di usabilità

L'usabilità del web o web usability è l'applicazione dei principi dell'usabilità in quegli ambiti in cui la navigazione del web può essere considerata come un paradigma generale (o "metafora") per costruire una interfaccia grafica (GUI).

La web usability è un approccio della progettazione per rendere i siti web facili da usare per l'utente finale, senza richiedere all'utente di sottostare a una formazione specifica. L'utente deve essere in grado di mettere in relazione in modo intuitivo le azioni che ha bisogno di compiere nella pagina web con altre interazioni che svolge fisicamente nella vita quotidiana, come ad esempio premere un pulsante per ottenere un'azione. L'obiettivo principale della web usability è in sintesi:

- Presentare l'informazione all'utente in modo chiaro e conciso
- Offrire all'utente le scelte corrette, in una maniera che risulti ovvia
- Eliminare ogni ambiguità relativa alle conseguenze di un'azione (es. fare clic su cancella/rimuovi/compra)
- Mettere la cosa più importante nella posizione giusta della pagina web o dell'applicazione web.

*fonte wikipedia*



# Usabilità in pratica?

Molto semplicemente è immedesimarsi con spirito critico sulle necessità dell'utente della pagina e del sito che stai progettando.

Due parole d'ordine:

KISS - Keep it simple and stupid

Don't make user think



# L'attenzione sul web

La possibilità di mantenere alta l'attenzione mentre un utente naviga è scarsa.

Gli utenti non leggono.

Scandiscono il testo.

**RITMO VERTICALE -> line-height (CSS)**

**NEGATIVE SPACE -> gli spazi bianchi**



# Esempi da evitare

La maggior parte delle produzioni dei Centri Sociali :)

Volantini politici ecc.

Troppe cose, tutte insieme, implicano confusione.

Confusione implica visita persa.

Visita persa implica sconfitta.



# Regoletta

I buoni designer ridisegnano, i grandi designer riallineano.

La maggior parte degli elementi di una pagina web, essendo un oggetto di interazione con gli utenti, hanno dei paradigmi che vanno rispettati, anche per reinventarli.

<http://patterntap.com>



# Progettazione

- Raccogliete requisiti.
- Raccogliete requisiti.
- Raccogliete requisiti.
- Avete raccolto i requisiti?
  - Scriveteli e sottoponeteli a chi dovete.
  - Discuteteli e raccogliete feedback.
- Fate prototipi... con che software?



# Il software perfetto

LA CARTA.



# Disegnate su carta

Fate dei prototipi rozzi: i wireframe.

Progettate l'occupazione degli spazi.

Progettateli come fosse uno storyboard di un fumetto.

Seguite TUTTO il flusso di navigazione degli utenti.

Ricordatevi che è una pagina in cui si naviga per raccogliere informazioni, rendetela elegante, non illegibile.



# Contatti

Carlo Frinolli

@carl0s\_

[carlo@nois3lab.it](mailto:carlo@nois3lab.it)

<http://nois3lab.it>

<http://blog.nois3lab.it>