

Introduzione a

HTML5, CSS3 & JS



INTRODUZIONE

MARCO TIBALDESCHI

Socio fondatore DBN Communication srl
Professore a contratto presso U.P.O. - Scienze Politiche
Innamorato dello sviluppo del software



... **VOI CHI SIETE?**

Che preparazione avete?
Cosa vi aspettate dal corso?

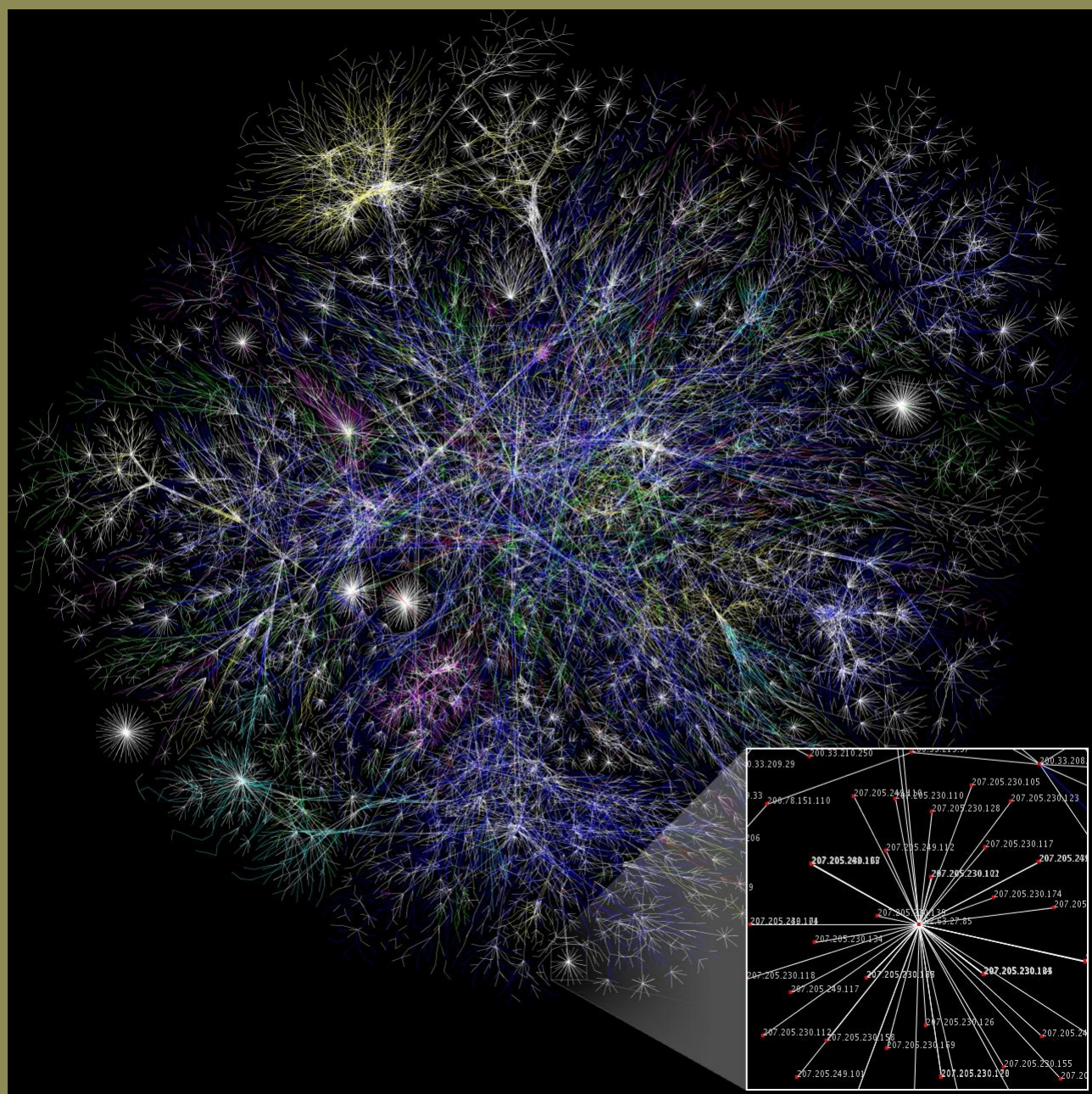
LE BASI DEL WEB

significa "diverso"



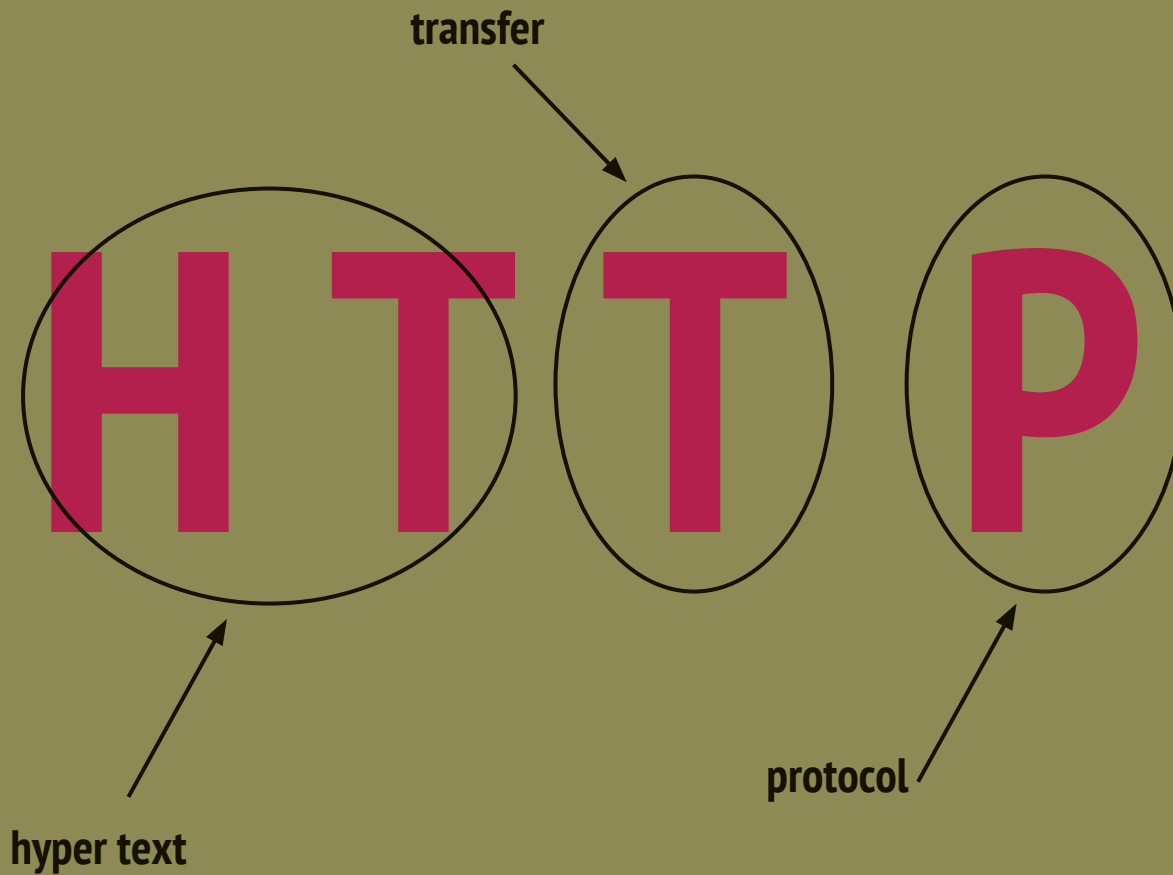
WEB != INTERNET

"Visualization of the various routes through a portion of the Internet." - Matt Britt, Wikipedia.



**IL WORLD WIDE WEB È
FORMATO DA TUTTI I
CONTENUTI MULTIMEDIALI
SERVITI SU INTERNET
TRAMITE HTTP**





HTTP

insieme di regole, formali, specificate e aperte. L'organismo che lo ha in gestione è il W3C (www.w3.org)



protocollo di trasferimento di ipertesti

insieme di contenuti relazionati tra loro attraverso dei collegamenti (link)



HTTP +

HTML +

URL =

Hyper Text Markup Language



Uniform Resource Locator



WEB



FINE ANNI 80



**IL PROTOCOLLO HTTP
DETERMINA LE MODALITÀ
DI COMUNICAZIONE TRA
CLIENT E SERVER
ATTRAVERSO UN MODELLO
RICHIESTA / RISPOSTA**

CLIENT

← BROWSER

SERVER

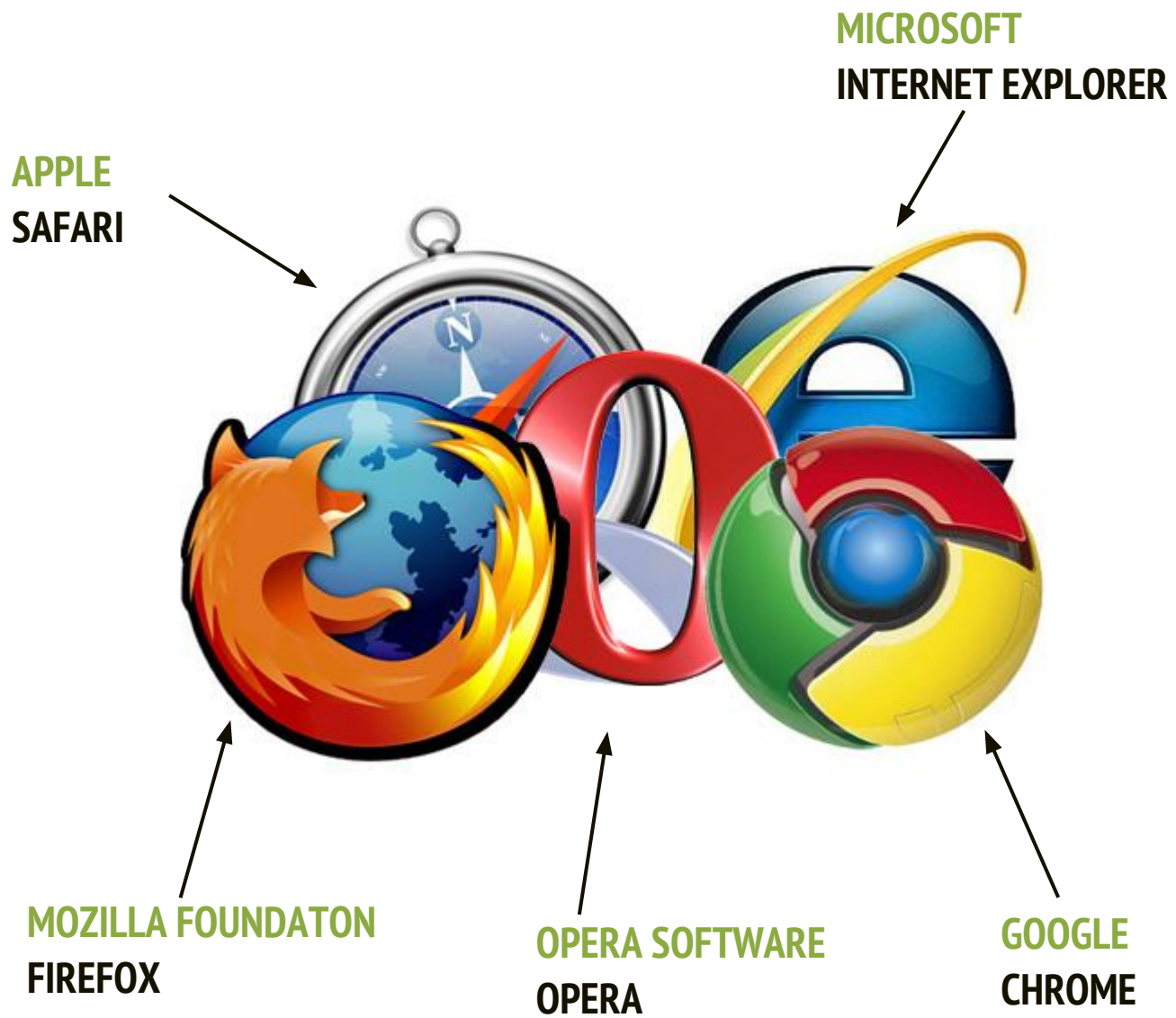
← SITO WEB

RICHIESTA

← DAMMI LA PAGINA, L'IMMAGNE, IL VIDEO, IL FILE MP3...

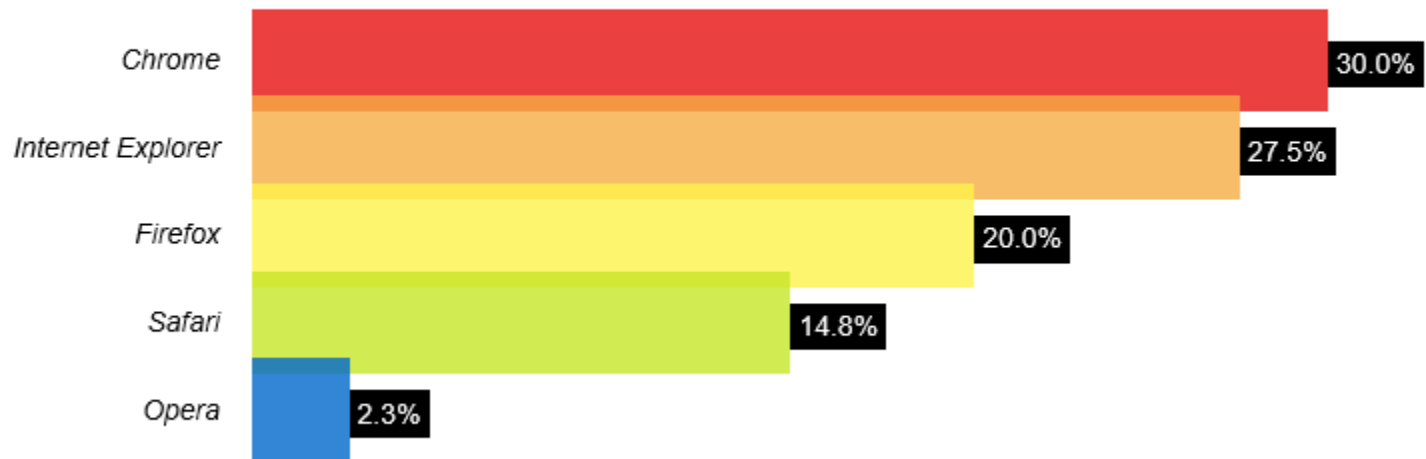
RISPOSTA

← ECCO LA PAGINA, L'IMMAGNE, IL VIDEO, IL FILE MP3...

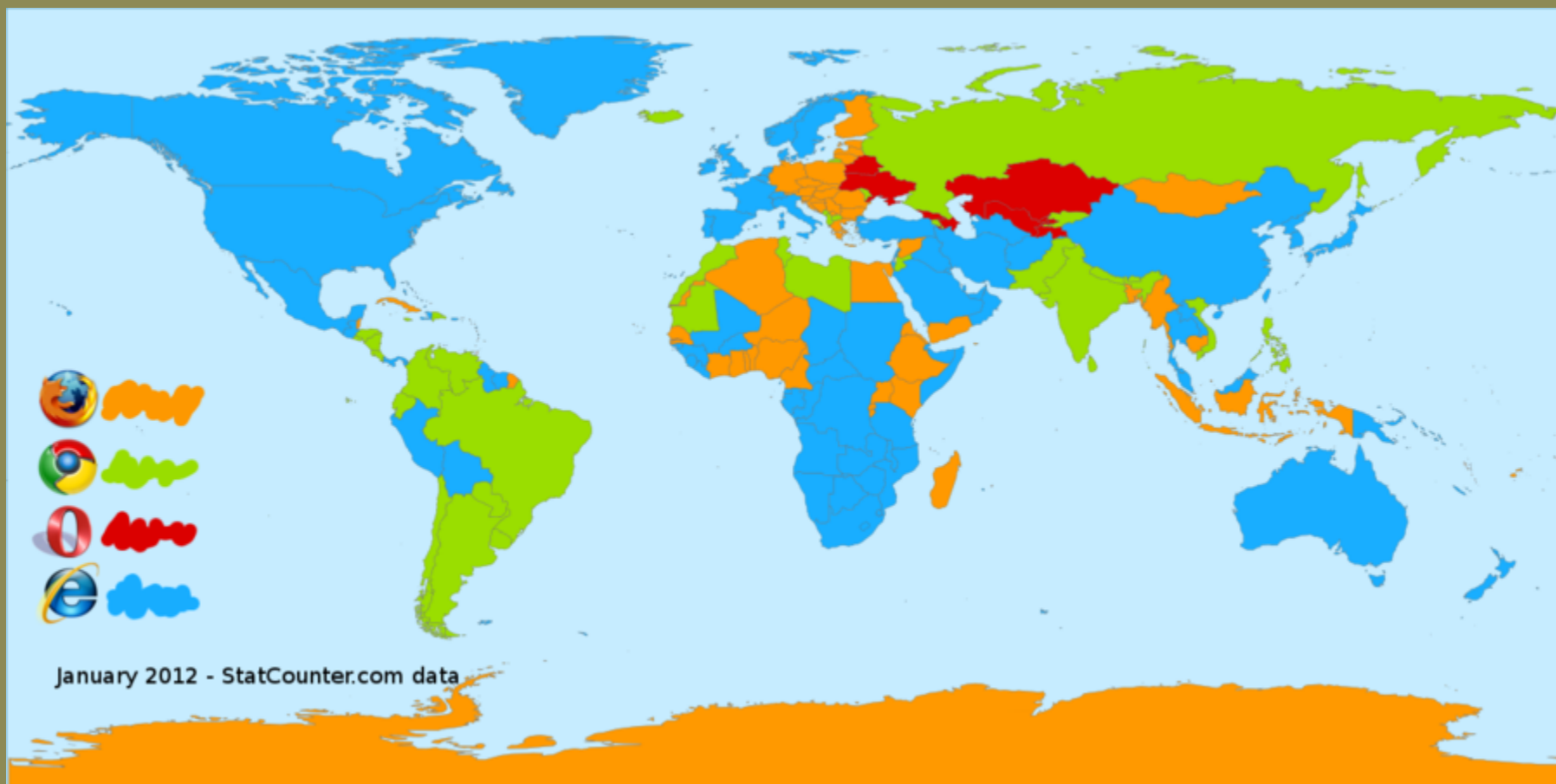


Web Browser Market Share

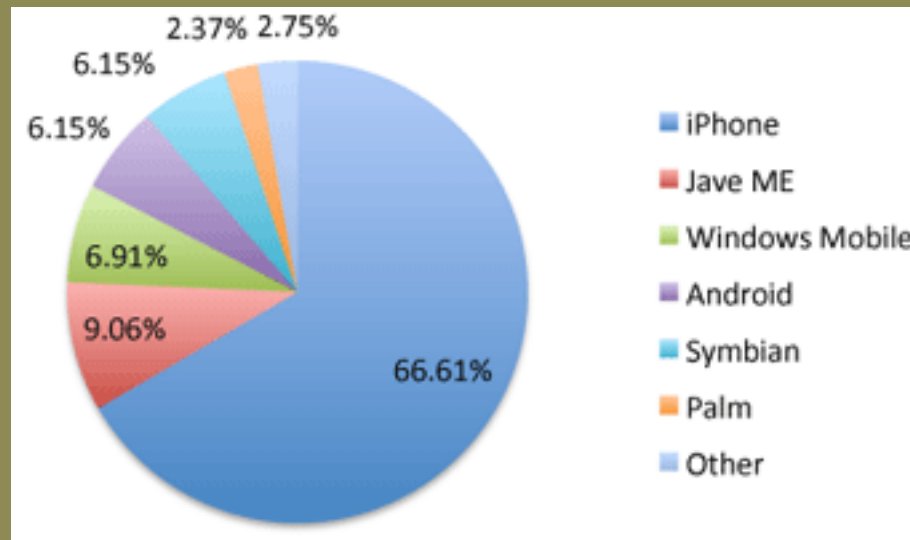
[NEW: View Trends](#)



fonte w3counter.com



LA DIFFUSIONE È DIVERSA NAZIONE PER NAZIONE



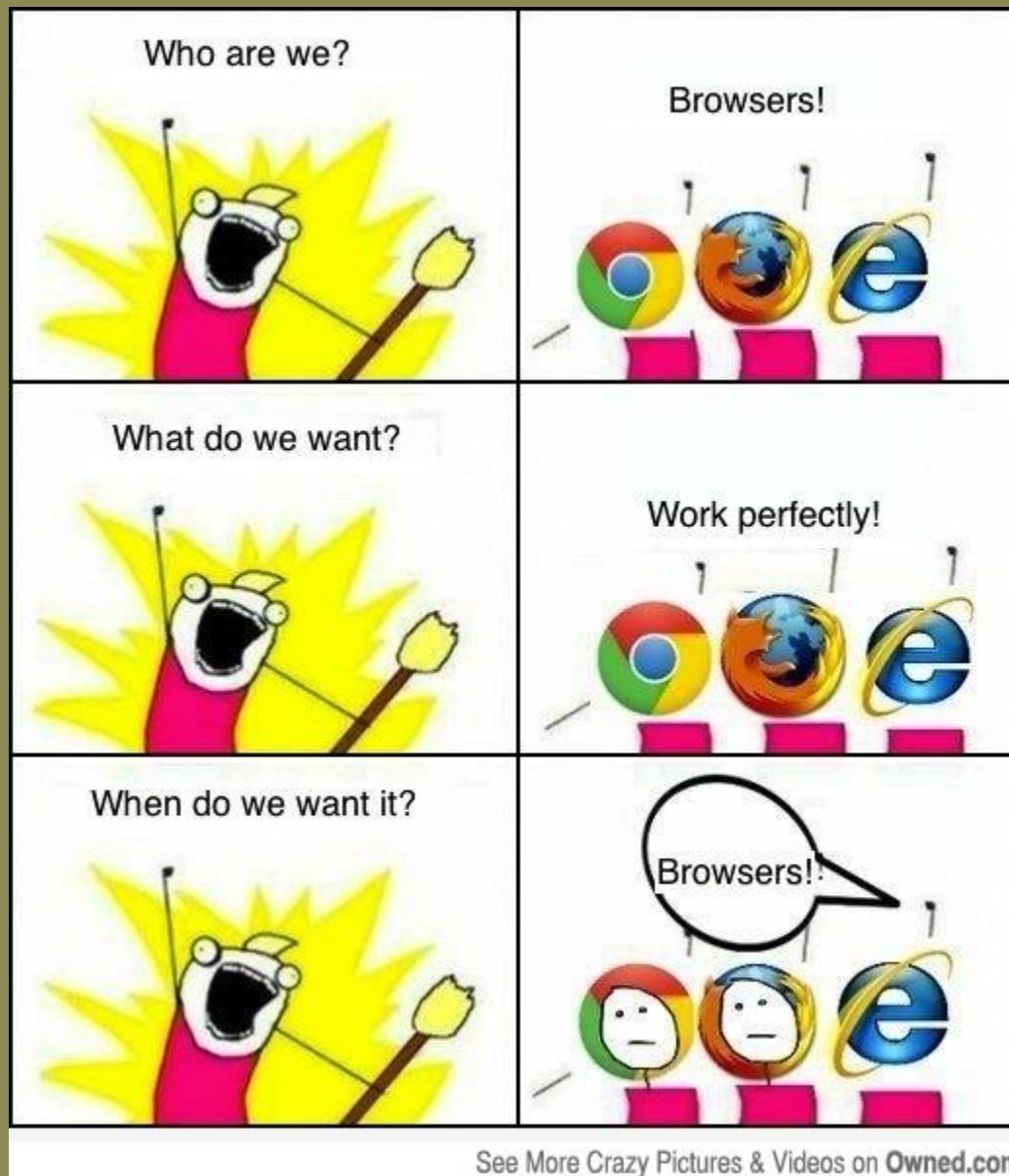
**LA DIFFUSIONE È DIVERSA
TRA COMPUTER DESKTOP / MOBILE**

**COME SVILUPPATORI
DI SITI WEB**

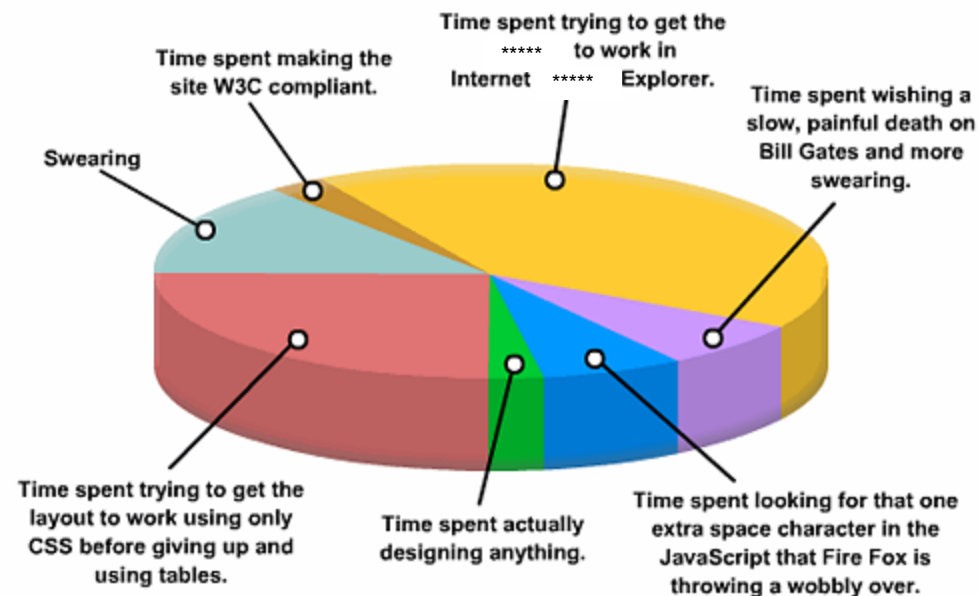
NON

**POSSIAMO IGNORARE
L'ESISTENZA DEI BROWSER**





TIME BREAKDOWN OF MODERN WEB DESIGN



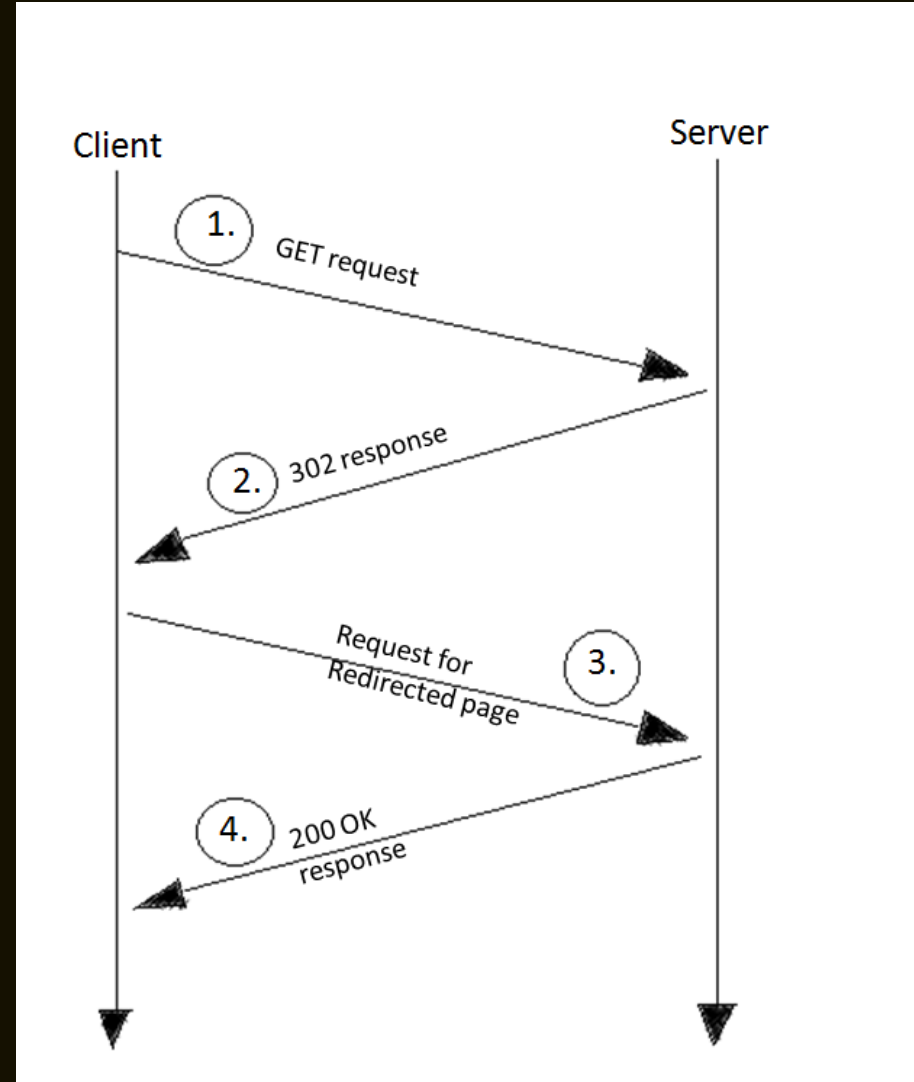
©2006 Alan "IE users must DIE" Foreman poisonedminds.com

**NON C'È
AMORE
TRA CLIENT
E SERVER**



HTTP È SENZA STATO (OVVERO OGNI RICHIESTA È UN MONDO A SÈ)

ESEMPIO DI RELAZIONE TRA CLIENT E SERVER



CON LA VERSIONE 1.1 DI HTTP, SONO STATE INTRODOTTE LE RICHIESTE PARALLELE

MA QUINDI...

**CON HTTP TRASFERIAMO
SOLAMENTE
DEI CONTENUTI?**

175 PAGINE DI SPECIFICA SOLO PER QUESTO?!

NO!

GET

← CHIEDE AL SERVER UNA RISORSA

POST

← CHEDE AL SERVER CHE LA RISORSA
INVIATA SIA AGGIUNTA ALLE RISORSE
GESTITE

PUT

← CHIEDE AL SERVER CHE LA RISORSA
INVIATA SIA MODIFICATA

DELETE

← CHIEDE AL SERVER CHE LA RISORSA
INVIATA SIA RIMOSSA

**COSA SONO
LE RISORSE?**

PRIMA

(FINO AL 1994)

UNA IMMAGINE, UN
DOCUMENTO
IN GENERALE UNA
QUALUNQUE
COSA **DISPONIBILE**
SULLA RETE
E **REPERIBILE**
ATTAVERSO UN
URL (INDIRIZZO).

DOPO

QUALUNQUE COSA,
ANCHE ASTRATTA
CHE POSSA ESSERE
IDENTIFICATA
ATTRAVERSO UN URI.
ESEMPIO:
UNA IMMAGINE, UN
DOCUMENTO, UN
SERVIZIO, **IO STESSO.**

UNIFORM
RESOURCE
IDENTIFIER

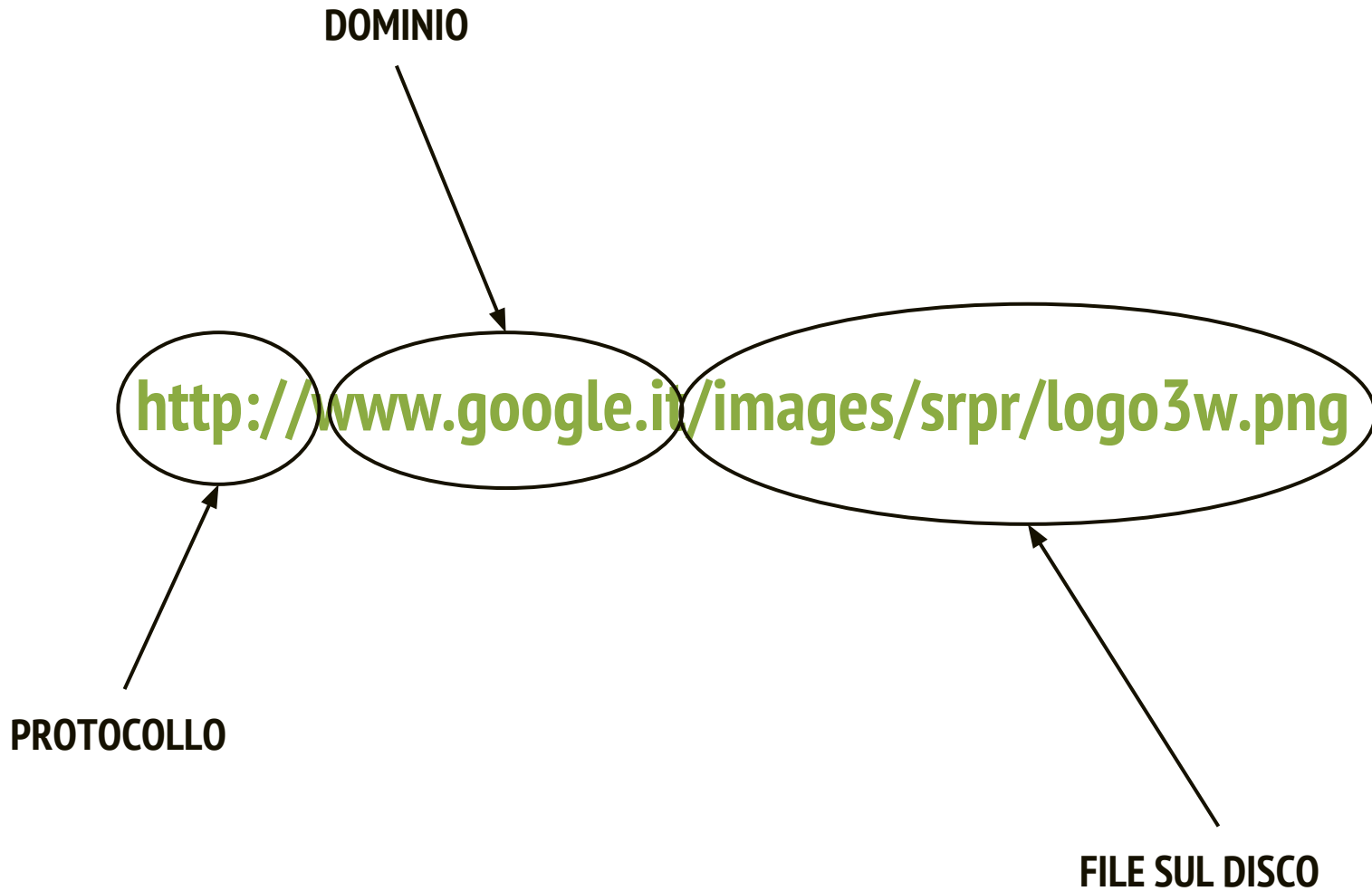
UN UNIFORM
RESOURCE
LOCATOR

UN URL È UN URI

```
graph TD; URI[UNIFORM RESOURCE IDENTIFIER] --> URI_text[URI]; URL[UNIFORM RESOURCE LOCATOR] --> URL_text[URL];
```

LA DIFFERENZA È SQUISITAMENTE TEORICA

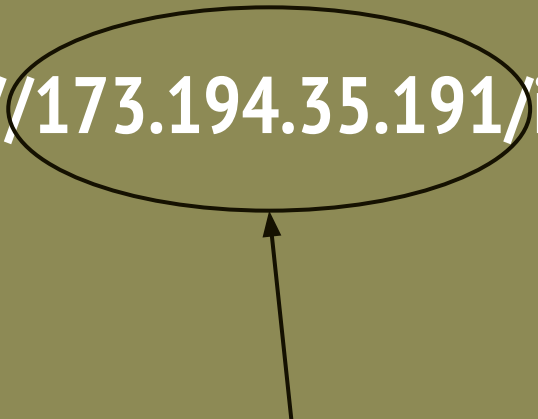
(IN GERGO È POSSIBILE USARE URL E URI INDIFFERENTEMENTE)



DOMINIO

1. UNICO PER TUTTA INTERNET
2. DEVE ESSERE REGISTRATO PRESSO UN REGISTRAR ACCREDITATO
3. DISPONIBILE IN DIVERSI GUSTI (.COM / .IT / .NET / .ORG)
4. NON È NECESSARIO PERCHÈ LE RISORSE SIANO VISIBILI SU INTERNET

<http://173.194.35.191/images/srpr/logo3w.png>



INDIRIZZO IP

IL DNS

TRADUCE IL DOMINIO

IN INDIRIZZO IP

CONCLUDENDO,

SVILUPPARE UN SITO WEB SIGNIFICA

1. CREARE DELLE RISORSE

1A. HTML

1B. CSS

1C. JAVASCRIPT

2. RENDERLE REPERIBILI SUL WEB TRAMITE HTTP

2A. REGISTRARE UN DOMINIO

2B. FARLE SERVIRE DA UN SERVER WEB

3. PREOCCUPARSI DELLA LORO UNIVERSALITÀ

3A. TESTING CROSS-BROWSER

3B. ANALISI PRESTAZIONALI

3C. ACCESSIBILITÀ, USER EXPERIENCE

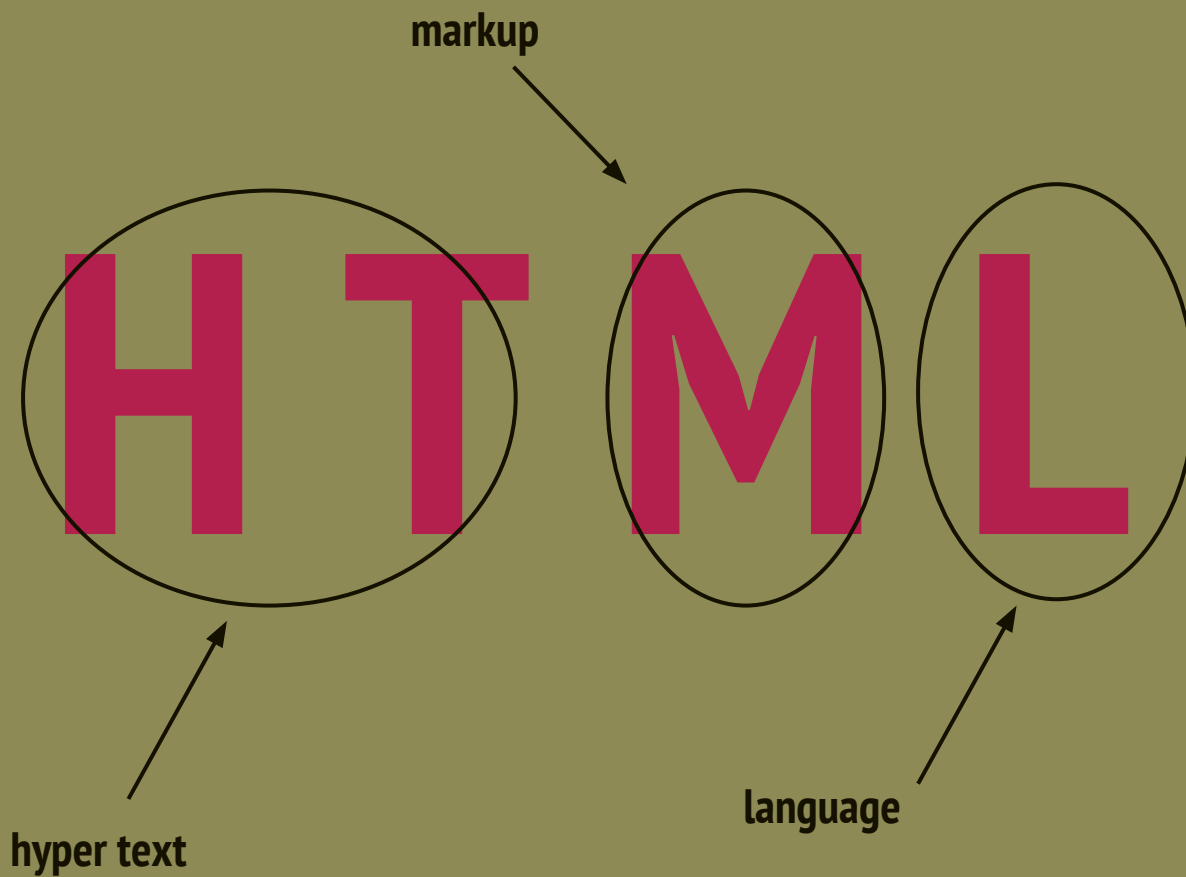
4. GARANTENDO LA LORO DISPONIBILITÀ

4A. RINNOVO DEL DOMINIO

4B. GESTIONE DEI RECORD DNS

4C. MONITORAGGIO SERVER

SEMPLICE, NO?



LINGUAGGO DI MARKUP PER IPERTESTI

← HA DELLE REGOLE!

← ESISTE DEL CONTENUTO CHE DESCRIVE
LA FORMA DEL CONTENUTO STESSO

↑
Questi sono i TAG

LE REGOLE

1. I TAG SONO BEN DEFINITI

(OVVERO ESISTE UN VOCABOLARIO DI TAG UTILIZZABILI)

(MAIUSCOLO / MINUSCOLO NON FA DIFFERENZA, MA È MEGLIO MINUSCOLO)

2. OGNI TAG APERTO DEVE VENIRE CHIUSO

3. I TAG DEVONO ESSERE CORRETTAMENTE ANNIDATI

(OVVERO CHI VIENE APERTO PRIMA, DEVE ESSERE CHIUSO DOPO)

I TAG DI BASE

<HTML> ... </HTML>

LA RADICE DEL DOCUMENTO, QUELLO CHE CONTIENE TUTTI, OBBLIGATORIO, INDIMENTICABILE

<HEAD> ... </HEAD>

CONTENUTO IN <HTML>, CONTIENE ALTRI TAG NON DIRETTAMENTE VISIBILI SUL BROWSER

<BODY> ... </BODY>

CONTENUTO IN <HTML>, CONTIENE ALTRI TAG CHE DETERMINANO STRUTTURA E CONTENUTO

LA TESTA: <HEAD>

<TITLE> ... </TITLE>

DEFINISCE IL TITOLO DELLA PAGINA (QUELLO CHE SI VEDE SULLA TAB)

<LINK> ... </LINK>

DEFINISCE - AD ESEMPIO - UN COLLEGAMENTO AD UN FOGLIO DI STILE

<SCRIPT> ... </SCRIPT>

DEFINISCE - AD ESEMPIO - UN COLLEGAMENTO AD UN FILE JAVASCRIPT

IL CORPO: <BODY>

<H1> ... </H1>, <H2> ... </H2>, <H3> ... </H3>, FINO H6

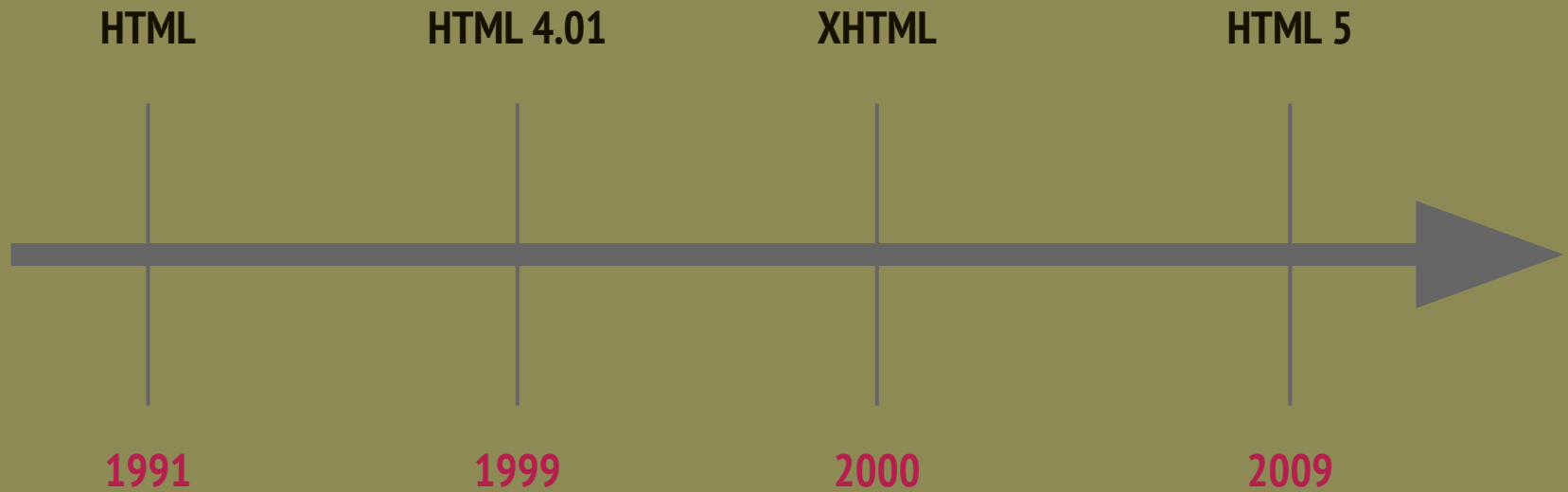
DEFINISCONO TITOLI (DI LIVELLO DIVERSO)

<P> ... </P>

DEFINISCE UN PARAGRAFO

<A> ...

DEFINISCE UN COLLEGAMENTO IPERTESTUALE



MA PRIMA... DOCTYPE

AIUTA IL BROWSER AD INTERPRETARE IL CONTENUTO DELLA PAGINA WEB

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01

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

ESEMPIO

```
<!DOCTYPE html>
<html>
  <head>
    <title>Benvenuti</title>
  </head>

  <body>
    <h1>Titolo</h1>
    <p>Ciao a tutti!</p>
  </body>
</html>
```

ATTRIBUTI: IL SALE DEI TAG!

AGGIUNGONO INFORMAZIONI
(NON DI SECONDARIA IMPORTANZA!)
SUI TAG STESSI.

LE REGOLE

1. I VALORI DEGLI ATTRIBUTI SEMPRE TRA VIRGOLETTE

(SINGOLE O DOPPIE NON FA DIFFERENZA!)

2. GLI ATTRIBUTI VALIDI SONO DIVERSI TRA TAG E TAG

(ESISTONO PROMEMORIA!!!)

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

http://www.w3schools.com/tags/ref_standardattributes.asp

INTESTAZIONI (H1...H6)

1. DEFINISCONO STRUTTURA TITOLI

(USATI DAI MOTORI DI RICERCA PER CAPIRE LA STRUTTURA DELLA PAGINA)

2. NON UTILIZZARE PER DIMENSIONARE IL FONT!

(PER QUESTO ESISTONO I CSS)

PARAGRAFI (P)

1. DEFINISCONO PARAGRAFO

(C'È UNA NUOVA LINEA AL TERMINE DEL PARAGRAFO)

2. NON SI POSSONO APRIRE P DENTRO ALTRI P

(NON SI PUÒ INIZIARE UN PARAGRAFO PRIMA CHE SIA FINITO UN ALTRO)

A CAPO (BR)

1. MANDANO A CAPO LA LINEA

2. NON INTERROMPONO IL PARAGRAFO

(QUINDI ALL'INTERNO DI UN PARAGRAFO SE NE POSSONO METTERE QUANTI SE NE VUOLE)

3. È UN TAG VUOTO

(QUINDI SI PUÒ SCRIVERE `<BR />`, CHE SIGNIFICA APERTO E CHIUSO IMMEDIATAMENTE)

4. NON USARE PER IMPAGINARE

(PER QUESTO ESISTONO I CSS)

TAG DI FORMATTAZIONE

1. **...**

(TESTO IMPORTANTE, SOLITAMENTE RAPPRESENTATO IN GRASSETTO)
(DIVERSO DA ... IN TERMINI SEMANTICI)

2. **...**

(ENFASI, SOLITAMENTE RAPPRESENTATO IN CORSIVO)
(DIVERSO DA <I>...</I> IN TERMINI SEMANTICI)

IMMAGINI (IMG)

1. VISUALIZZA UN'IMMAGINE SULLA PAGINA

2. È UN TAG VUOTO, QUINDI SI PUÒ SCRIVERE `<IMG />`

3. L'ATTRIBUTO SRC INDICA IL PERCORSO
DELL'IMMAGINE

(ATTENZIONE! SE NON È UN URL, SI INTENDE RELATIVO ALLA PAGINA!)

ESEMPIO

```

```

LINK IPERTESTUALI (A)

1. IL CONTENUTO DEL TAG È IL TESTO "LINKATO"
2. L'ATTRIBUTO HREF CONTIENE L'URL DA LINKARE

ESEMPIO

```
<a href="http://www.google.com/">Visita Google!</a>
```

3. L'ATTRIBUTO TARGET DICE DOVE APRIRE IL LINK

(ESEMPIO TARGET="_BLANK" APRE IN UNA NUOVA FINESTRA)

ANCORE (A)

SONO COLLEGAMENTI IPERTESTUALI ALL'INTERNO DELLA PAGINA

SEGUONO LE STESSE REGOLE DEI LINK, MA COME VALORE DELL'ATTRIBUTO HREF SI INSERISCE IL VALORE DELL'ATTRIBUTO "ID" DELL'ELEMENTO DA ANCORARE.

ESEMPIO

```
<h1 id="iniziopagina">Benvenuti!</h1>
```

....

```
<a href="#iniziopagina">Torna in cima</a>
```

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Creare un documento HTML5 che contenga un'intestazione, un paragrafo e un link a www.google.it.

INTERMEDIO

Creare un documento HTML5 che contenga un'intestazione, un sottotitolo, un paragrafo, un link a www.google.it e un'immagine.

NERD

Creare un documento HTML5 che contenga un'intestazione, un sottotitolo, un paragrafo, un link a www.google.it su un'immagine che si apra in una nuova finestra.

BLOCCO (DIV)

1. RACCHIUDE PIÙ TAG DIVERSI

2. NON HA UNA RAPPRESENTAZIONE GRAFICA

3. MOLTO GENERICO!

ALCUNI DETTAGLI

(NON PROPRIO DI SECONDARIA IMPORTANZA)

TIPI DI ELEMENTO

DI BLOCCO

SPEZZANO IL FLUSSO CON UNA INTERRUZIONE DI LINEA

ESEMPI: DIV, P, BR...

IN LINEA

NON SPEZZANO IL FLUSSO

ESEMPI: IMG, A, SPAN...

ATTRIBUTO LANG

ATTRIBUTO DEL TAG HTML,
DETERMINA LA LINGUA DELLA PAGINA

ESEMPIO

```
<html lang="it">
```

<http://people.w3.org/rishida/utils/subtags/index.php?list=4&submit=List>

CHARSET (ATTR. DI META)

TAG HTML DA INSERIRE IN HEAD
STABILISCE IN CHE MODO DEBBANO ESSERE
RAPPRESENTATI I BIT TRASMESSI
IMPORTANTE!

SE NON SPECIFICATO LASCIA IL VOSTRO SITO APERTO AD ATTACCHI!!!

ESEMPIO

(da inserire all'interno di head)

```
<meta charset="utf-8" />
```

<http://www.iana.org/assignments/character-sets/character-sets.xml>

LINK

TAG HTML DA INSERIRE IN HEAD
STABILISCE UNA RELAZIONE TRA LA PAGINA HTML
E ALTRE RISORSE.

IL TIPO DI RELAZIONE È DETERMINATO DALL'ATTRIBUTO REL

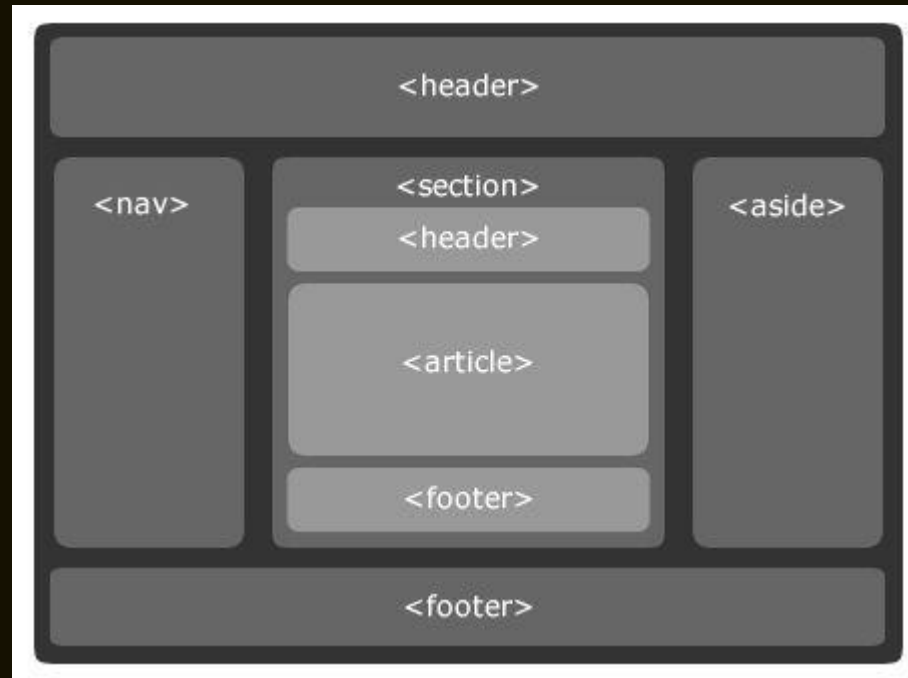
rel="stylesheet" (predefinito)

rel="alternate" (versione differente della pagina)

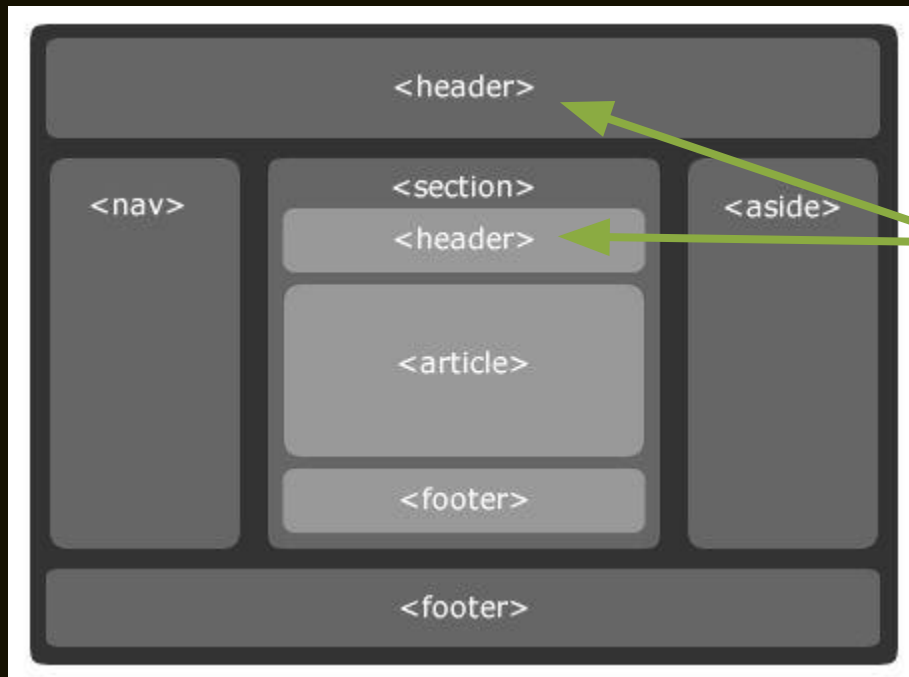
rel="icon" (icona)

TAG SEMANTICI

(SOLO HTML5)



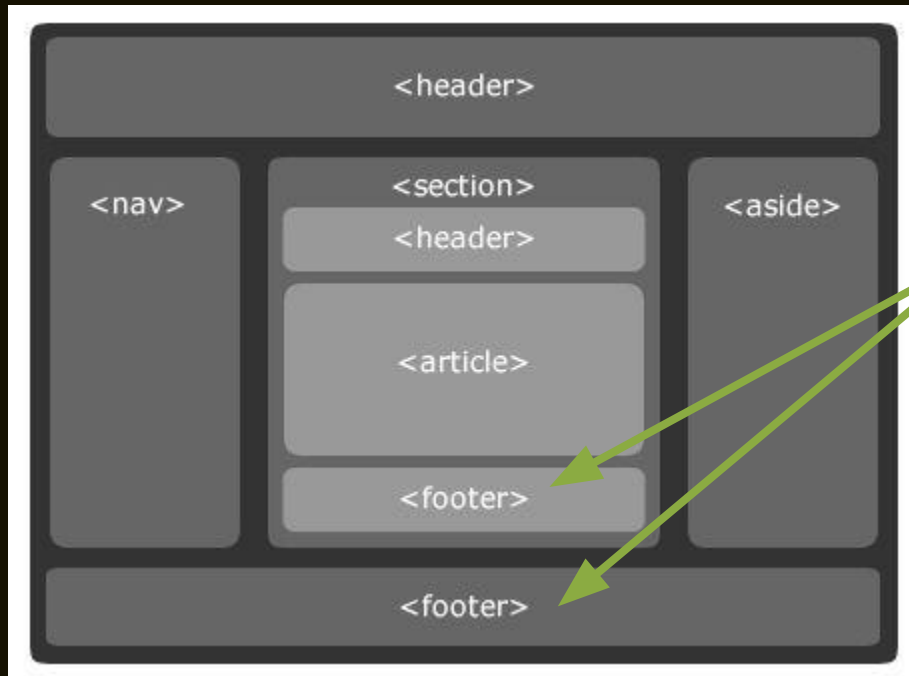
INTESTAZIONE (HEADER)



**IDENTIFICA
L'INTESTAZIONE DI
UNA SEZIONE O DI UN
DOCUMENTO**

(POSSONO ESSERCENE PIÙ D'UNO IN
UNA PAGINA)

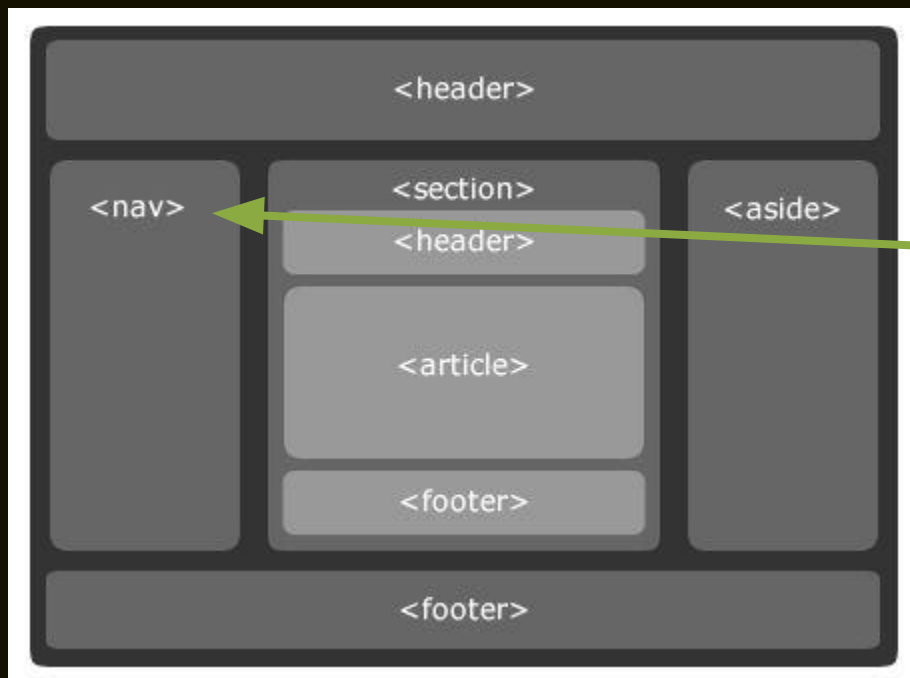
PIÈ DI PAGINA (FOOTER)



**IDENTIFICA IL PIÈ DI
PAGINA DEL
DOCUMENTO (O DI UNA
SEZIONE SPECIFICA)**

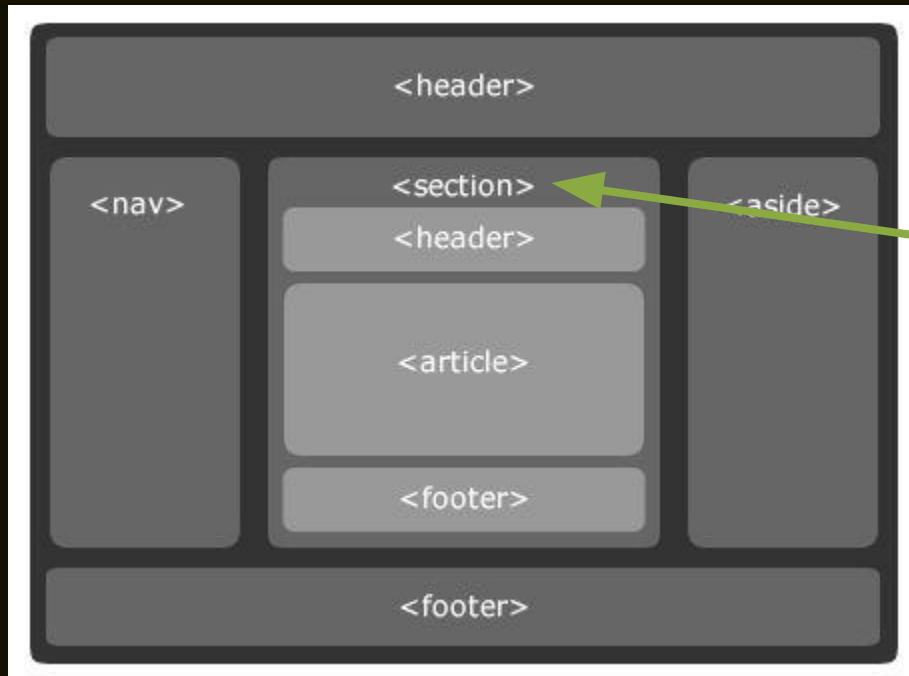
(POSSONO ESSERCENE PIÙ D'UNO
ALL'INTERNO DELLA PAGINA)

NAVIGAZIONE (NAV)



**IDENTIFICA UNA
SEZIONE CONTENENTE
I LINK DI NAVIGAZIONE
PRINCIPALI
ALL'INTERNO DEL SITO
(UTILE PER I NON VEDENTI)**

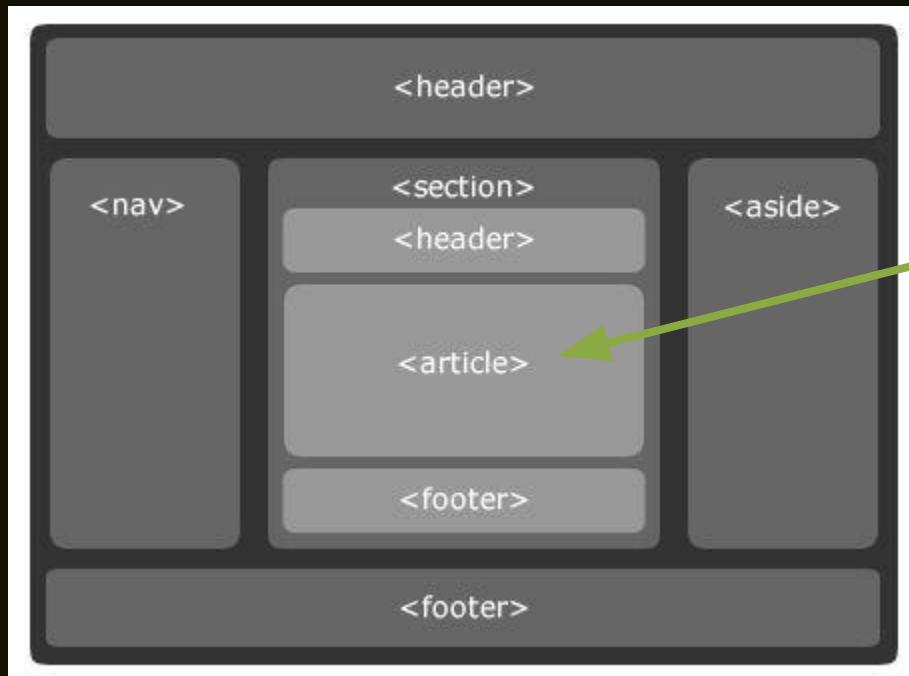
SEZIONE (SECTION)



IDENTIFICA UNA SEZIONE ALL'INTERNO DEL SITO. A SUA VOLTA UNA SECTION PUÒ CONTENERE ALTRI HEADER E FOOTER.

(POSSONO ESSERCENE PIÙ D'UNO ALL'INTERNO DELLA PAGINA E POSSONO ANCHE ESSERE ANNIDATE)

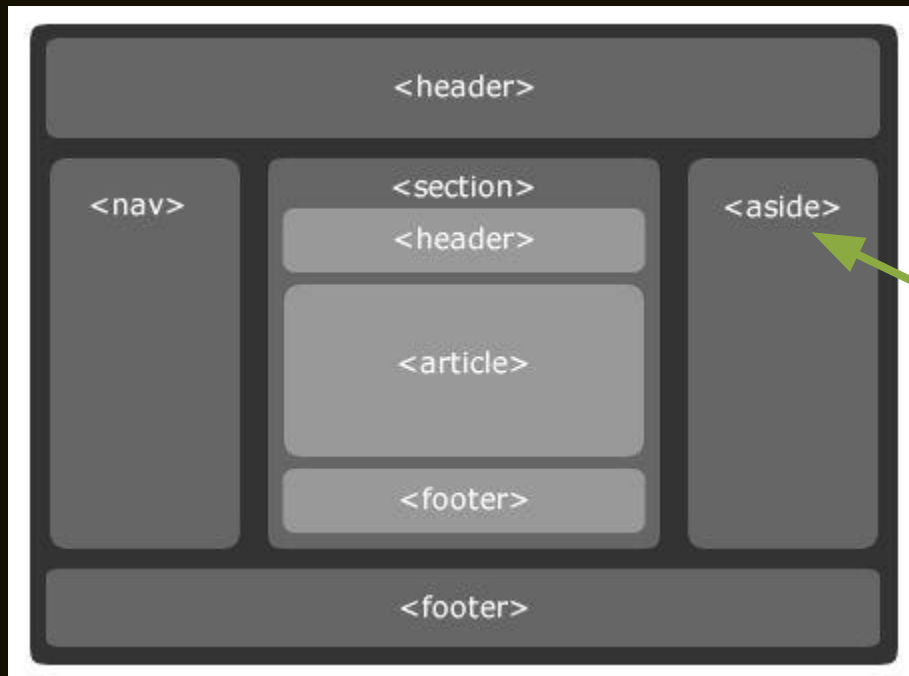
ARTICOLO (ARTICLE)



**IDENTIFICA UN
ARTICOLO CHE
CONCETTUALMENTE
POTREBBE ESSERE
ESTRATTO DAL SITO E
NON PERDERE
SIGNIFICATO.**

(POSSONO ESSERCENE PIÙ D'UNO
ALL'INTERNO DELLA PAGINA)

INFO DI CONTORNO (ASIDE)



**IDENTIFICA UNA
SEZIONE DI
CONTENUTO NON
PRINCIPALE, MA
CORRELATO.**

(POSSONO ESSERCENE PIÙ D'UNO
ALL'INTERNO DELLA PAGINA)

RIASSUMENDO

Una **section** è un raggruppamento tematico, tipicamente preceduto da un tag **header** e chiuso da un tag **footer**.

Una sezione può essere annidata ad un'altra e può contenere ogni tipo di tag.

Una **header** di una sezione contiene tipicamente il titolo, ma può anche contenere ulteriori info sulla sezione.

Un **footer** di una sezione contiene tipicamente informazioni sulla sezione stessa (chi l'ha scritta, link a documenti correlati...)

Il tag **nav** contiene un elenco di link, generalmente i link principali di un sito.

Infine **article** e **aside** vengono usati rispettivamente per contenuti indipendenti e contenuti correlati.

COMPORTAMENTO DEI BROWSER

**SE IL BROWSER NON RICONOSCE UN TAG PARTICOLARE
LO TRATTERÀ COME IN LINEA!**

**È POSSIBILE CREARE UN FOGLIO DI STILE CHE IMPONGA,
INVECE, CHE GLI ELEMENTI NUOVI HTML5 VENGANO
CONSIDERATI COME DI BLOCCO!**

<http://html5doctor.com/html-5-reset-stylesheet/>

...MA...

C'È SEMPRE

INTERNET EXPLORER

A STUPIRCI!

COMPORTAMENTO DEI BROWSER

SE INTERNET EXPLORER (VER < 9) NON RICONOSCE UN TAG, SEMPLICEMENTE NON APPLICA EVENTUALI REGOLE CSS DA NOI DEFINITE!!!

È POSSIBILE, USANDO UN PO' DI JAVASCRIPT, RIMUOVERE QUESTO LIMITE!

<http://remysharp.com/2009/01/07/html5-enabling-script/>

```
<script type="text/javascript" src="http://html5shiv.googlecode.com/svn/trunk/html5.js" />
```

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Modificare il vostro documento HTML5 in modo tale che contenga un'intestazione del sito (header) e un articolo (article).

INTERMEDIO

Modificare il vostro documento HTML5 in modo tale che contenga un'intestazione del sito (header), una sezione di link (nav) e un articolo (article) con una sua intestazione (header).

NERD

Modificare il vostro documento HTML5 in modo tale che contenga un'intestazione (header) e un footer (footer) del sito, una sezione di link (nav), un articolo (article) con una sua intestazione (header) e un suo footer (footer).

**QUALI BROWSER
SUPPORTANO
HTML5?**

DIPENDE!

HTML5 NON È MONOLITICO!

LA SPECIFICA HTML5 DEFINISCE TANTISSIME
FUNZIONALITÀ DIVERSE ALCUNE DELLE QUALI
POSSONO ESSERE SUPPORTATE
E ALTRE NO!



**QUALI
FUNZIONALITÀ
DI HTML5 SONO
SUPPORTATE?**

<http://www.html5test.com/>

MODERNIZR

<http://www.modernizr.com>

CANVAS

DISEGNARE CON HTML5

- 1. È UN RETTANGOLO SULLA PAGINA**
- 2. LA MATITA SI MUOVE TRAMITE JS (CANVAS API)**
- 3. È POSSIBILE REALIZZARE DI TUTTO CON I CANVAS**
- 4. NON SONO MOLTO SEMPLICI DA UTILIZZARE**

VIDEO

FILMATI CON HTML5

- 1. PERMETTE DI INSERIRE VIDEO DENTRO LA PAGINA**
- 2. NON NECESSITA DI PLUGIN PARTICOLARI**
- 3. SUPPORTA DIVERSI FORMATI**

LOCAL STORAGE

MEMORIZZARE SUL CLIENT CON HTML5

1. PERMETTE DI SALVARE INFORMAZIONI SUL CLIENT
2. SUPERA I LIMITI DEI COOKIE (NUMERO E DIMENSIONI)
3. IL CONTENUTO NON È ACCESSIBILE DA ALTRI SITI, MA CHI HA ACCESSO AL CLIENT PUÒ VEDERE TUTTO.

WEB WORKERS

PARALLELISMO CON HTML5

- 1. PERMETTE DI ESEGUIRE, IN PARALLELO, DIVERSI SCRIPT JAVASCRIPT.**
- 2. IN QUESTO MODO L'INTERFACCIA UTENTE NON RISULTA "BLOCCATA" IN CASO DI CALCOLI COMPLESSI**

OFFLINE APP

UTILIZZARE IL SITO, ANCHE OFFLINE, CON HTML5

1. PERMETTE DI ESEGUIRE IL SITO ANCHE QUANDO NON SI È CONNESSI AD INTERNET
2. IL CLIENT, LA PRIMA VOLTA CHE VISITA IL SITO, SCARICA TUTTE LE RISORSE NECESSARIE PER VISUALIZZARLO ANCHE OFFLINE
3. LE MODIFICHE EVENTUALMENTE FATTE, POTRANNO ESSERE INVIATE AL SERVER QUANDO SI TORNA ONLINE

GEOLOCATION

LA NOSTRA POSIZIONE NEL MONDO CON HTML5

- 1. PERMETTE DI CONOSCERE LA NOSTRA POSIZIONE SULLA TERRA E (VOLENDO) DI CONDIVIDERLA CON CHI TI FIDI**
- 2. UTILIZZA METODI DIVERSI: INDIRIZZO IP, INFORMAZIONI SULLA RETE WIFI, CELLA DELLA RETE CELLULARE, GPS...**

INPUT TYPES

NUOVI TAG PER MODULI CON HTML5

1. DEFINISCE NUOVI TIPI DI CAMPO PER I MODULI
COME: BOX DI RICERCA, NUMERI DI TELEFONO,
INDIRIZZI EMAIL, URL, DATE...
2. PERMETTE DI DEFINIRE UN PLACEHOLDER TEXT
3. NUOVE FUNZIONALITÀ: AUTOFOCUS E VALIDAZIONE

MICRODATA

METAINFORMAZIONI SEMANTICHE CON HTML5

- 1. NON VISIBILI DIRETTAMENTE SULLA PAGINA**
- 2. PERMETTONO FUNZIONALITÀ AGGIUNTIVE
(ES. INFORMAZIONI DI LICENZA SULLE FOTOGRAFIE)**
- 3. I MOTORI DI RICERCA UTILIZZANO QUESTE
INFORMAZIONI, SE LI CONOSCONO. IN CASO
CONTRARIO VENGONO IGNORATI.**

HISTORY API

OPERARE SULLA CRONOLOGIA CON HTML5

- 1. PERMETTE DI OPERARE SULLA CRONOLOGIA INSERENDO DEGLI ELEMENTI OPPURE REAGIRE QUANDO QUESTI VENGONO RIMOSSI**
- 2. PERMETTONO, IN ALTRE PAROLE, DELLE FUNZIONALITÀ DI CRONOLOGIA ANCHE IN APPLICAZIONI CHE FANNO USO DI UNA SOLA PAGINA**

VIDEO

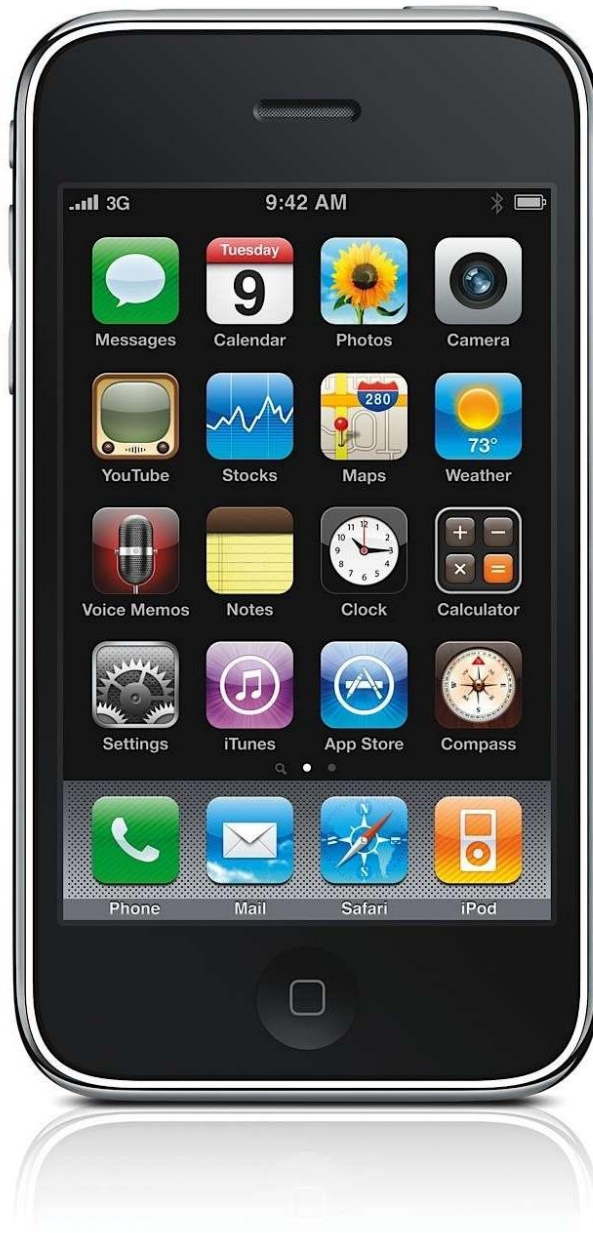
AUDIO

http://www.w3schools.com/html/html5_video.asp

http://www.w3schools.com/html/html5_audio.asp

PERCHÈ VIDEO E AUDIO?

**FINO A PRIMA DI HTML5 NON ESISTEVA LA POSSIBILITÀ
DI INSERIRE UN VIDEO ALL'INTERNO DI UNA PAGINA WEB
SENZA OBBLIGARE L'UTENTE AD AVERE INSTALLATO UN
PLUGIN (FLASH, SILVERLIGHT...)**



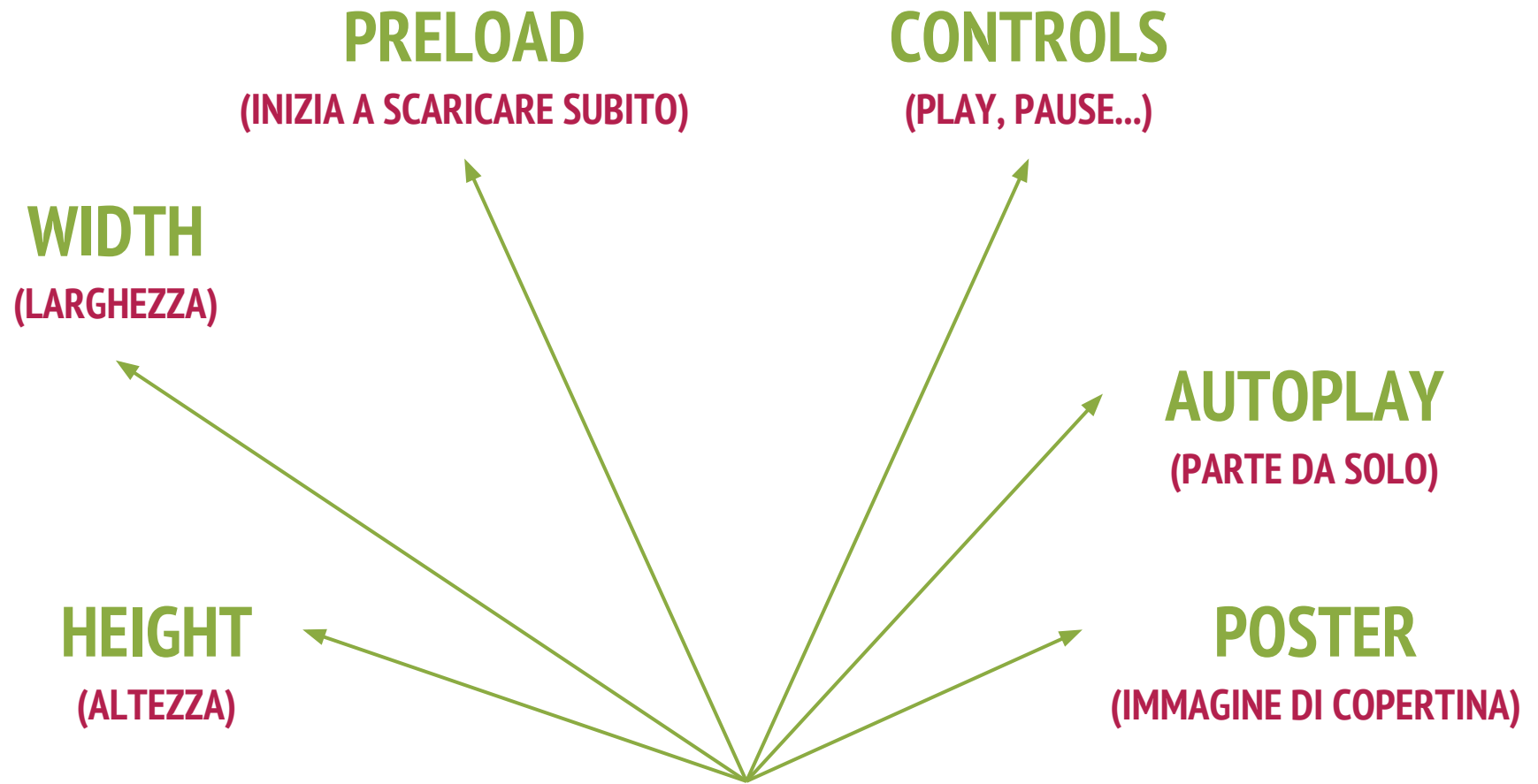


iPhone



<VIDEO>

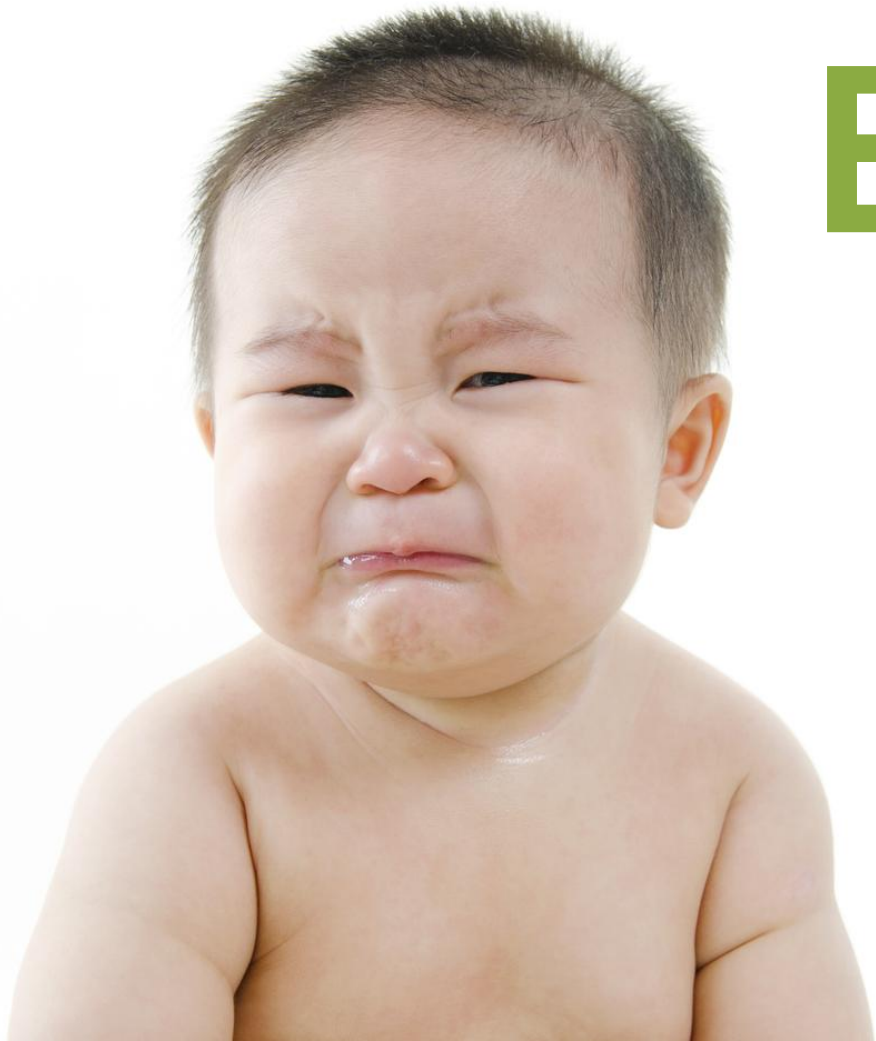
DEFINITO DA OPERA NEL 2007 PER RENDERE I VIDEO
CITTADINI DI SERIE A DEL WEB (COME LE IMMAGINI)



<VIDEO>

QUALI FORMATI?

TUTTI...
E NESSUNO!



**LA SPECIFICA HTML5
ATTUALE **NON** DEFINISCE
UN FORMATO STANDARD**

**OGNI BROWSER È
ATTUALMENTE LIBERO DI
SUPPORTARE I FORMATI
CHE PREFERISCE!**

È POSSIBILE CHE QUESTA COSA CAMBI NEL FUTURO PROSSIMO

I FORMATI PIÙ DIFFUSI

- APERTO
- (FORSE) SENZA COPYRIGHT



OGG



- CHIUSO
- COPERTO DA COPYRIGHT



H.264



- APERTO
- SENZA COPYRIGHT



WEBM



H.264

**UN PO' MENO
DEL 50%**



WEBM / OGG

**UN PO' PIÙ
DEL 50%**



E QUINDI?!

CASO SEMPLICE

(MA NON UNIVERSALE)

USANDO
L'ATTRIBUTO
SRC
SPECIFICHIAMO
UN SOLO FILE.

ESEMPIO

```
<video  
  src="festa.ogv"  
  width="800" height="600"></video>
```

CASO COMPLESSO

USANDO I TAG SOURCE DENTRO AL TAG VIDEO SPECIFICHIAMO PIÙ FILE.

ESEMPIO

```
<video width="800" height="600">  
  <source src="festa.ogv" type="video/ogg; codecs=theora,vorbis" />  
  <source src="festa.webm" type="video/webm; codecs=vp8,vorbis" />  
  <source src="festa.mp4" type="video/mp4; codecs=avc1.42E01E,mp4a.40.2"/>  
</video>
```

http://wiki.whatwg.org/wiki/Video_type_parameters

Two green arrows originate from the code examples. One arrow points from the 'codecs=theora,vorbis' parameter in the first source tag to the URL. The other arrow points from the 'codecs=vp8,vorbis' parameter in the second source tag to the same URL.

IL BROWSER SCEGLIERÀ, ANALIZZANDOLI
NELL'ORDINE, IL VIDEO CON IL FORMATO CHE
PUÒ ESSERE APERTO!

**E PER CHI NON
SUPPORTA IL TAG
VIDEO?**

RETROCOMPATIBILITÀ

CHI NON SUPPORTA HTML5, IGNORERÀ IL TAG VIDEO, PROCESSANDO, COMUNQUE GLI ELEMENTI CONTENUTI IN ESSO.

ESEMPIO

```
<video width="800" height="600">
  <source src="festa.ogv" type="video/ogg; codecs=theora,vorbis" />
  <source src="festa.webm" type="video/webm; codecs=vp8,vorbis" />
  <source src="festa.mp4" type="video/mp4; codecs=avc1.42E01E,mp4a.40.2"/>
  <p>Spiacente, solo per chi usa un browser che supporta HTML5!</p>
</video>
```

**SOLUZIONI
AFFIDABILI?**

VIDEO FOR EVERYBODY!

http://camendesign.com/code/video_for_everybody

1. Se video HTML5 è supportato, fa uso di quello
2. Se video HTML5 non è supportato, fa uso di un player Flash
3. Se Flash non è installato, viene visualizzata una immagine con un link al file video.

CONTROLS

(PLAY, PAUSE...)

PRELOAD

(INIZIA A SCARICARE SUBITO)

AUTOPLAY

(PARTE DA SOLO)

<AUDIO>

I FORMATI PIÙ DIFFUSI

MP3



WAV



OGG



**COME PER
<VIDEO>
DUE MODALITÀ**

CASO SEMPLICE

(MA NON UNIVERSALE)

USANDO
L'ATTRIBUTO
SRC
SPECIFICHIAMO
UN SOLO FILE.

ESEMPIO

```
<audio  
  src="festa.ogg">  
</audio>
```

PUÒ ESSERE ANCHE UN URL



CASO COMPLESSO

USANDO I TAG SOURCE DENTRO AL TAG AUDIO
SPECIFICHIAMO PIÙ FILE.

ESEMPIO

```
<audio>  
  <source src="festa.ogg" />  
  <source src="festa.mp3" />  
  <source src="festa.wav" />  
</audio>
```

POSSONO ESSERE ANCHE URL



IL BROWSER SCEGLIERÀ, ANALIZZANDOLI
NELL'ORDINE, L'AUDIO CON IL FORMATO CHE
PUÒ ESSERE APERTO!

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

All'interno della pagina che avete sviluppato sino ad ora, inserire un video che abbia come sorgente uno degli URL video indicati.

INTERMEDIO

All'interno della pagina che avete sviluppato sino ad ora, inserire un video (con controls) che abbia due sorgenti.

NERD

All'interno della pagina che avete sviluppato sino ad ora, inserire un video (con autoplay e controls) che abbia due sorgenti e un audio .

URL VIDEO

<http://html5demos.com/assets/dizzy.mp4>
<http://html5demos.com/assets/dizzy.webm>
<http://html5demos.com/assets/dizzy.ogv>

URL AUDIO

<http://www.dontbenegative.it/download/html5/bunny.mp3>

MODULI

http://www.w3schools.com/html/html_forms.asp

PERCHÈ I MODULI?

PERCHÈ VOGLIAMO DEI FEEDBACK DA PARTE DEI NOSTRI UTENTI.

I MODULI ERANO GIÀ POSSIBILI CON HTML4, MA CON HTML5 SONO DEFINITI NUOVI TIPI DI CAMPO.

METHOD

(VERBO HTTP DA
USARE, TIPICAMENTE
POST)

AUTOCOMPLETE

(ON OPPURE OFF
SOLO CON HTML5, STABILISCE
SE DEVE ESSERE SUPPORTATO
L'AUTOCOMPLETAMENTO)

ACTION

(URL A CUI SARANNO
MANDATE LE
INFORMAZIONI DEL
MODULO - TIPICAMENTE
PAGINA PHP, ASP.NET...)

NOVALIDATE

(SOLO HTML5, DISABILITA LA
VALIDAZIONE)



<FORM>

CONTROLLI DI INPUT

DENTRO AL NOSTRO TAG FORM DOBBIAMO SPECIFICARE UN INSIEME DI CONTROLLI DI INPUT (CAMPI DI TESTO, BOTTONI, CHECKBOX...) CHE PERMETTANO ALL'UTENTE DI COMPILARE IL MODULO.

CONTROLLI DI INPUT

INPUT

(CAMPO DI INPUT GENERICO, IL TIPO È DEFINITO DALL'ATTRIBUTO TYPE)

TEXTAREA

(CAMPO DI TESTO GENERICO, LINEE MULTIPLE)

SELECT (ALL'INTERNO, OPTION)

(TENDINA, CON SCELTA SINGOLA OPPURE MULTIPLA)

CHECKBOX

(SPUNTA, PERMETTE LA SCELTA
DI PIÙ VALORI IN UN INSIEME
RISTRETTO DEFINITO
DALL'ATTRIBUTO NAME)

SUBMIT / RESET

(BOTTONI DI INVIO / PULIZIA
DEL MODULO)

RADIO

(RADIOBUTTON, PERMETTE LA
SCELTA DI UN SINGOLO
VALORE, IN UN INSIEME
RISTRETTO DEFINITO
DALL'ATTRIBUTO NAME)

PASSWORD

(CAMPO DI TESTO,
SINGOLA LINEA,
TESTO OSCURATO)

TEXT

(CAMPO DI TESTO,
SINGOLA LINEA)

<INPUT TYPE="...">

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Creare un form con tre campi di testo (nome, cognome ed email) e un bottone per inviare.

INTERMEDIO

Creare un form con tre campi di testo (nome, cognome ed email), un radiobutton per il sesso (maschio / femmina) e un bottone per inviare.

NERD

Creare un form con tre campi di testo (nome, cognome ed email), un radiobutton per il sesso (maschio / femmina), un gruppo di checkbox per le conoscenze (html5 / css / javascript) e un bottone per inviare.

TUTTO QUI?

NUMERI? EMAIL? URL? DATE?

EMAIL

(EMAIL, FACILITA
L'INSERIMENTO PER I
VISITATORI CHE USANO
DISPOSITIVI MOBILI)



NUMBER / RANGE

(NUMERI. È POSSIBILE STABILIRE UN
MINIMO (MIN), UN MASSIMO (MAX),
UNO STEP (STEP) E UN VALORE
PREDEFINITO (VALUE))



SEARCH

(CAMPO DI RICERCA, AGGIUNGE
QUALCHE ELEMENTO DI UI)



URL

(URL, FACILITA L'INSERIMENTO
PER I VISITATORI CHE USANO
DISPOSITIVI MOBILI)



DATE / DATETIME / TIME

(DATA / DATA E ORA / ORA)



COLOR

(SCELTA DI UN COLORE)



<INPUT TYPE="...">

CHE CONFUSIONE!

TUTTI I BROWSER, ANCHE
QUELLI ANTICHI (IE6) SE NON
CONOSCONO IL **TYPE**, LO
INTERPRETANO COME **TEXT**!

ESEMPIO

INOLTRE ALCUNI

ATTRIBUTI

INTERESSANTI

ATTRIBUTI INTERESSANTI

PLACEHOLDER="..." 

(TESTO PREDEFINITO DI UN CAMPO DI TESTO, VIENE RIMOSSO NON APPENA L'UTENTE CLICCA DENTRO ALLO STESSO)

AUTOFOCUS="AUTOFOCUS" 

(IL CAMPO DI TESTO, IL BOTTONE, LA CHECKBOX VENGONO SELEZIONATI APPENA LA PAGINA È PRONTA)

COME
CONTROLLARE
L'INPUT INSERITO
DALL'UTENTE?

PRE-HTML5

VALIDAZIONE LATO CLIENT CON
JAVASCRIPT

+

VALIDAZIONE LATO SERVER

HTML5

VALIDAZIONE LATO CLIENT **BUILT-IN**

+

VALIDAZIONE LATO SERVER

ATTRIBUTI INTERESSANTI

REQUIRED="REQUIRED"

(INDICA CHE IL PARTICOLARE CONTROLLO DI INPUT DEVE CONTENERE UN VALORE PRIMA DI POTER PERMETTERE L'INVIO DEL FORM)

NOVALIDATE="NOVALIDATE"

FORMNOVALIDATE="FORMNOVALIDATE"

(POSSIBILITÀ DI INVIARE IL FORM SENZA VALIDAZIONE)

MAGGIORI INFORMAZIONI:

<http://www.html5rocks.com/en/tutorials/forms/constraintvalidation/>

**COME
COMPORTARSI
CON I BROWSER
ANTICHI?**

WEBSHIMS

<http://afarkas.github.com/webshim/demos/index.html>

NECESSITA DI JQUERY

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Modificare il form con specificando che il campo "email" debba essere di tipo email. Aggiungere un campo numerico per specificare l'età (compresa tra 10 e 100 anni).

INTERMEDIO

Come novizio + aggiunta di placeholder opportuni su ogni campo.

NERD

Come intermedio + aggiunta di un bottone che permetta di inviare il modulo senza validare (es. per salvare il modulo)

ESERCIZIO

FINALE

SU HTML5

ESERCIZIO

Creare tre pagine HTML5, che condividano la stessa struttura e che useremo per i prossimi esercizi sui CSS.

Le tre pagine devono essere: **homepage**, **prodotti** e **contattaci**.

Le tre pagine devono avere **una immagine che serve da logo**, un **menu** contenenti tre link e devono fare uso dei **tag semantici** di HTML5 (deve esserci una header, un footer e un nav, oltre a quelli che userete per le pagine)

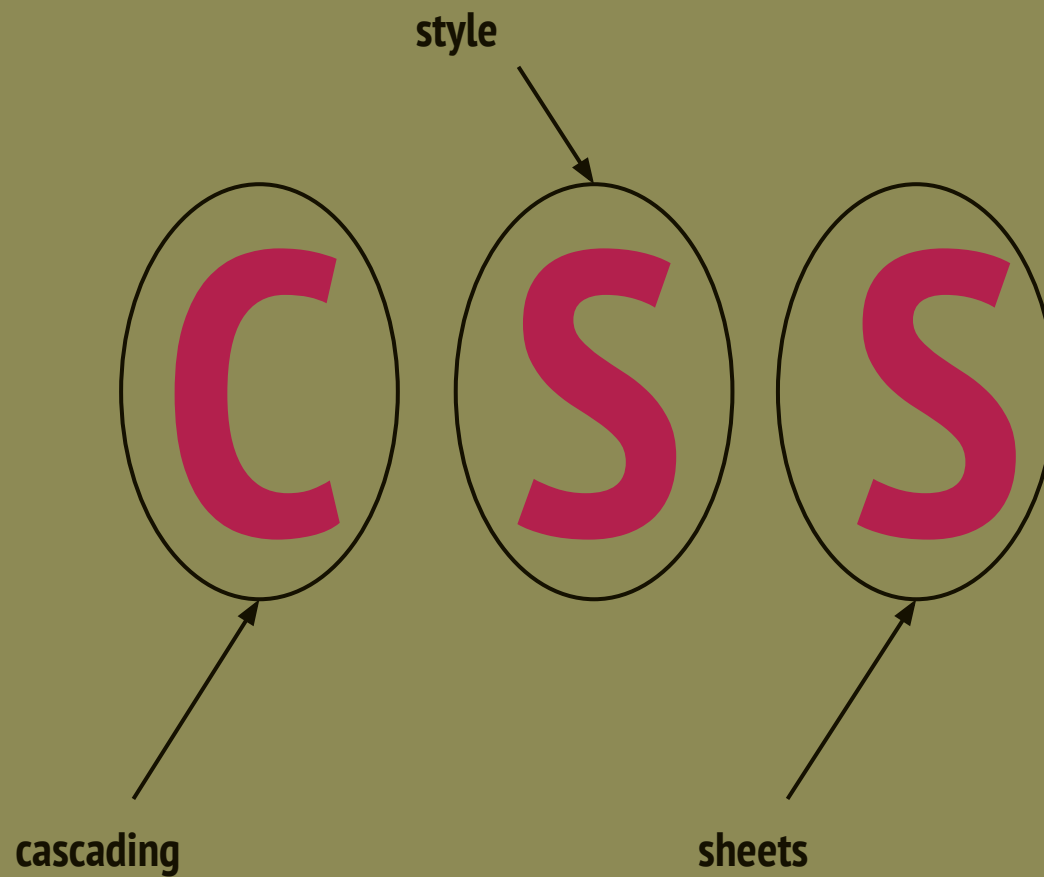
La **homepage** deve contenere un messaggio di benvenuto.

La **pagina di prodotti** deve contenere tre box di tre prodotti (foto, nome, descrizione e link (per ora disattivato) con testo dettaglio)

La **pagina contatti** deve contenere un modulo di contatto (campi decisi da voi, ma almeno nome, cognome, email ed un bottone per inviare).

css





I CSS sono un insieme di regole il cui compito è quello di istruire il browser su come rappresentare i contenuti del documento HTML associato.

Purtroppo queste regole non vengono interpretate correttamente da tutti motori di rendering (specialmente quello di Internet Explorer).

**COME ASSOCIARE UN
FILE *CSS* AD UN
DOCUMENTO *HTML*?**

TYPE="TEXT/CSS"

(DEFINISCE IL MIME-TYPE)

HREF="..."

(PERCORSO AL FILE CSS)

REL="STYLE SHEET"

(STABILISCE IL TIPO DI LINK)

MEDIA = "..."

(STABILISCE A QUALE MEDIA SI
APPLICA IL FOGLIO DI STILE.
MAGGIORI INFORMAZIONI PIÙ
AVANTI)

<LINK>

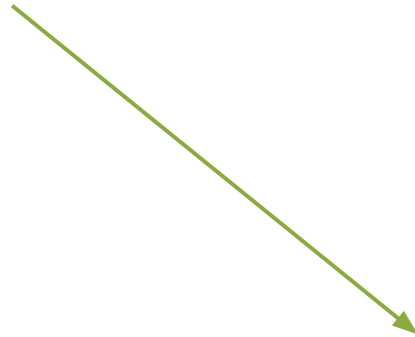
ESEMPIO

```
<!DOCTYPE html>
<html>
  <head>
    <title>Benvenuti</title>
    <link rel="stylesheet" type="text/css" href="css/stile.css" />
  </head>

  <body>
    <h1>Titolo</h1>
    <p>Ciao a tutti!</p>
  </body>
</html>
```

**COSA CONTIENE
UN FILE CSS?**

QUESTA È UNA REGOLA CSS



Selettore

h1

Dichiarazione

Dichiarazione

{color:blue; font-size:12px;}

Proprietà / Valore

Proprietà / Valore

SELETTORI

Selettore

h1

Dichiarazione

{color:blue; font-size:12px;}

Dichiarazione

↑
Proprietà / Valore

↑
Proprietà / Valore

SELETTORE UNIVERSALE

SI APPLICA A TUTTI GLI ELEMENTI DEL DOCUMENTO HTML



ESEMPIO

```
*  
{  
  margin: 0;  
  padding: 0;  
}
```

SELETTORE DI TIPO

SI APPLICA A TUTTI GLI ELEMENTI DI QUEL TIPO
(I SEGUENTI SONO ESEMPI!)

div { ... }

article { ... }

p { ... }

header { ... }

SELETTORE DI CLASSE

SI APPLICA A TUTTI GLI ELEMENTI CHE HANNO UN ATTRIBUTO
CLASS CON QUEL PARTICOLARE VALORE

IL PUNTO INDICA CHE È UN
SELETTORE DI CLASSE

QUESTA REGOLA SI
APPLICA A TUTTI GLI
ELEMENTI CHE HANNO
CLASS="BELLO"

.bello { ... }

QUESTA REGOLA SI
APPLICA A TUTTI GLI
ELEMENTI CHE HANNO
CLASS="BRUTTO"

.brutto { ... }

SELETTORE DI ID

SI APPLICA ALL'ELEMENTO CHE HA UN ATTRIBUTO
ID CON QUEL PARTICOLARE VALORE

IL CANCELLETTO INDICA CHE È
UN SELETTORE DI ID

QUESTA REGOLA SI
APPLICA ALL'ELEMENTO
CHE HA ID="BELLO"

#bello { ... }

QUESTA REGOLA SI
APPLICA ALL'ELEMENTO
CHE HA ID="BRUTTO"

#brutto { ... }

PERCHÈ CASCADING?

CASCATA

CASCADING SIGNIFICA CHE SE DUE O PIÙ REGOLE SI APPLICANO INDIFFERENTEMENTE ALLO STESSO ELEMENTO VIENE EFFETTUATO UN CALCOLO CHE DETERMINA LA SPECIFICITÀ DI OGNI REGOLA.

LA REGOLA CON SPECIFICITÀ PIÙ ALTA, VINCE.
SE PIÙ REGOLE HANNO UGUALE SPECIFICITÀ, VINCE QUELLA CHE È SCRITTA DOPO (CASCATA).

CALCOLO DELLA SPECIFICITÀ

<http://specificity.keegan.st/>

VALORI

Selettore

h1

Dichiarazione

{color:blue; font-size:12px;}

Dichiarazione

Proprietà / Valore

Proprietà / Valore

NUMERI

ASSOLUTI



L'UNITÀ DI MISURA È ASSOLUTA, NON
DIPENDE DA ALTRI FATTORI. PUÒ
ESSERE RIFERITA AL METRO.
ESEMPI: 2CM, 1MM, 20IN, 10PT...

RELATIVI



L'UNITÀ DI MISURA È RELATIVA,
OVVERO DIPENDE DA ALTRI FATTORI.
ESEMPI: 1EM, 1EX, 12PX ...

PERCENTUALI



IL VALORE È DIPENDENTE DA UN
VALORE DI RIFERIMENTO, TIPICAMENTE
L'ELEMENTO PADRE.
ESEMPIO: 12%, 20% ...

URI

si scrive url('...')

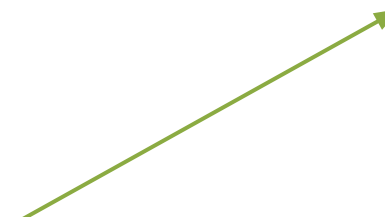
ESEMPIO:

url('http://www.sito.it/img/logo.jpg')

ASSOLUTI



RELATIVI AL PERCORSO DEL FOGLIO DI STILE



ESEMPIO:

url('../img/sfondo.jpg')

COLORI

ESADECIMALE

#FF0000, #FFFF00 ...

http://www.w3schools.com/tags/ref_colorpicker.asp

PAROLA CHIAVE

red, green, blue, ...

http://www.w3schools.com/cssref/css_colornames.asp

RGB(...,...,...)

RGB(X,Y,Z)

0 <= X,Y,Z <= 255

RGBA(...,...,...,...)

RGBA(X,Y,Z,A)

0.0 <= A <= 1.0

SOLO CSS3

0 = comp. trasparente / 1 opaco

PAROLE CHIAVE

**È POSSIBILE CHE UNA PROPRIETÀ
ABBA UN DETERMINATO INSIEME DI
VALORI POSSIBILI, COSTITUITI
DALLE SUE PAROLE CHIAVE.**

PROPRIETÀ

Selettore

h1

Dichiarazione

{color:blue; font-size:12px;}

Dichiarazione

Proprietà / Valore

Proprietà / Valore

BOX MODEL

OGNI ELEMENTO HTML PUÒ ESSERE RIGUARDATO COME UN BOX CON LE SEGUENTI PROPRIETÀ:



BOX MODEL

WIDTH

NUMERO

Indica la larghezza

```
div { width: 200px; }
```

HEIGHT

NUMERO

Indica l'altezza

```
div { height: 300px; }
```

QUESTE DUE PROPRIETÀ POSSONO ESSERE IMPOSTATE SOLO PER ELEMENTI DI BLOCCO (ES. DIV, P, ...)

BOX MODEL

PADDING

NUMERO / NUMERI

Se specificato un solo numero indica la grandezza del padding su tutti e quattro i lati.
Altrimenti su ogni singolo lato (top / right / bottom / left)

```
div { padding: 2px; }  
div { padding: 2px 2px 3px 4px; }
```

MARGIN

NUMERO / NUMERI

Se specificato un solo numero indica la grandezza del margine su tutti e quattro i lati.
Altrimenti su ogni singolo lato (top / right / bottom / left)

```
div { margin: 20px; }  
div { margin: 20px 10px 5px 10px; }
```

GLI ELEMENTI IN LINEA POSSONO SPECIFICARE SOLO IL MARGINE A SINISTRA O A DESTRA (NON TOP NÈ BOTTOM)

BORDI

BORDER-STYLE:

KEYWORD

none / solid / dotted / dashed ...

```
div { border-style: solid; }
```

Indica lo stile del bordo (nessuno / solido / punteggiato / tratteggiato)

BORDER-WIDTH:

NUMERO

Indica la larghezza del bordo

```
div { border-width: 2px; }
```

BORDER-COLOR:

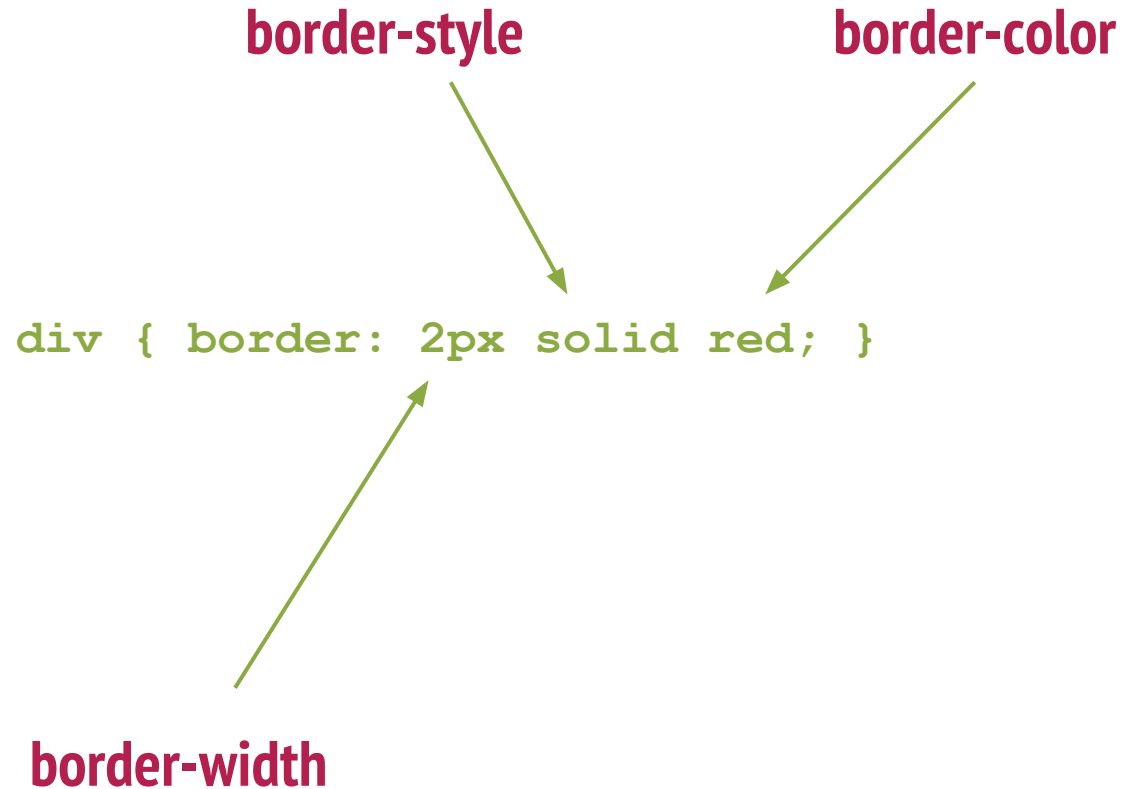
COLORE

Indica il colore del bordo

```
div { border-color: #CC0000; }
```

BORDI

FORMA RIDOTTA - l'ordine delle proprietà è importante



BORDI ARROTONDATI (CSS 3)

BORDER-RADIUS:
NUMERO / NUMERI

```
div { border-radius: 25px; }
```



No rounded corner



Rounded using an
arc of circle



Rounded using an
arc of ellipse

```
div { border-radius: 25px / 10px; }
```

IMMAGINI COME BORDI (CSS 3)

BORDER-IMAGE

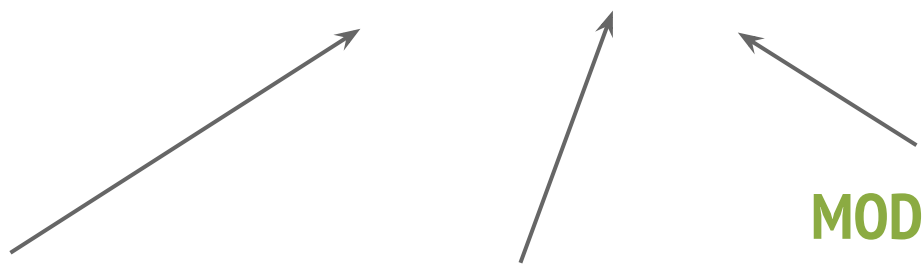
NON È SUPPORTATO DA INTERNET EXPLORER, MENTRE:

OPERA: **-O-BORDER-IMAGE**

WEBKIT: **-WEBKIT-BORDER-IMAGE**

```
div { border-image: url('dot.png') 25% repeat;  
}
```

IMMAGINE DA USARE



PUNTO DI TAGLIO
DELL'IMMAGINE
(SLICE)

MODALITÀ DI
RIPETIZIONE DEI
TRATTI NON DI
ANGOLO

IMMAGINI COME BORDI (CSS 3)

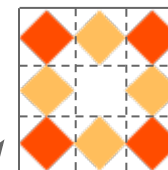


SUPPONIAMO CHE QUESTA SIA LA NOSTRA DOT.PNG
dimensioni: 81px x 81px (ovvero ogni blocchetto è 27px x 27px)

LA NOSTRA PROPRIETÀ SLICE PUÒ AVERE DA 1
A 4 VALORI CHE RAPPRESENTANO I PUNTI DI
TAGLIO DELL'IMMAGINE.

27 27 27 27

TOP RIGHT BOTTOM LEFT



LA NOSTRA IMMAGINE VIENE VIRTUALMENTE
TAGLIATA DAL BROWSER IN 9 PUNTI

IMMAGINI COME BORDI (CSS 3)

L'ULTIMO PARAMETRO DETERMINA COME IL BROWSER DEBBA COMPORTARSI NELLA PARTI NON D'ANGOLO DEL BORDO.

REPEAT: RIPETE LA FETTA PER LA LUNGHEZZA, EVENTUALMENTE TAGLIANDOLA



STRETCH: STIRA UNA SINGOLA FETTA PER TUTTA LA LUNGHEZZA



ROUND: COME REPEAT SE IL NUMERO DI FETTE UTILIZZABILI È INTERO, ALTRIMENTI VENGONO RIDOTTE IN MODO TALE CHE CI STIANO.

BOX MODEL

CON I CSS2 LE PROPRIETÀ WIDTH E HEIGHT AGISCONO SOLO SULLA ZONA CONTENT.

PER CONOSCERE IL REALE INGOMBRO DI OGNI BOX OCCORRE QUINDI SOMMARE AL VALORE LA DIMENSIONE DEL MARGINE + DIMENSIONE DEL BORDO + DIMENSIONE DEL PADDING.

POCO PRATICO! DEVO SEMPRE CALCOLARE E NON HO CONTROLLO COMODO SUL REALE INGOMBRO DEL BOX (BORDO + PADDING + CONTENUTO)

BOX MODEL (CSS 3)

BOX-SIZING

KEYWORD

CONTENT-BOX / BORDER-BOX

Indica da dove debbano essere considerate le proprietà height e width.

```
div
{
    box-sizing: content-box;

    width: 200px;
    padding: 5px;
    border: 2px;
    margin: 5px;
}
```

ZONA DI CONTENUTO: 200PX

PADDING: 5PX + 5PX

BORDER: 2PX + 2PX

MARGIN: 5PX + 5PX

TOTALE: 224PX

```
div
{
    box-sizing: border-box;

    width: 200px;
    padding: 5px;
    border: 2px;
    margin: 5px;
}
```

PADDING: 5PX + 5PX

BORDER: 2PX + 2PX

ZONA DI CONTENUTO: 200PX - PADDING - BORDER = 186PX

MARGIN: 5PX + 5PX

TOTALE: 210PX

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Impostare uno stile sull'intestazione e sul footer del sito, utilizzando le proprietà del box model (width / height / padding e margin).

INTERMEDIO

Come novizio + impostare un bordo con stile e colore a piacere all'intestazione e al footer.

NERD

Come intermedio + bordi arrotondati sugli article (stile e colore a piacere)

SFONDI

BACKGROUND-COLOR:

COLORE

Indica il colore di sfondo

```
div { background-color: red; }
```

BACKGROUND-IMAGE:

URI

Indica l'immagine da usare come sfondo

```
div { background-image: url('image.jpg');  
}
```

BACKGROUND-REPEAT:

KEYWORD

no-repeat / repeat / repeat-x / repeat-y

Indica in che modo deve essere ripetuta l'immagine di sfondo (non ripetuta / su x e y / solo su x / solo su y)

```
div { background-repeat: no-repeat; }
```

BACKGROUND-POSITION:

NUMERO/KEYWORD

Indica il punto di partenza dell'immagine.

```
div { background-position: top right; }
```

SFONDI MULTIPLI (CSS3)

BACKGROUND-IMAGE:

URI, URI, URI, ...

È possibile, separandoli da virgola, specificare più di un url, in modo tale da specificare più immagini di sfondo contemporaneamente.

Le immagini appaiono su livelli diversi, ovvero la prima immagine specificata è quella che sta più di fronte, via via fino all'ultima che risulta, quindi, essere sullo sfondo.

Le proprietà aggiuntive (background-repeat / background-position) possono quindi presentare valori diversi, separati da virgola, per ogni immagine usata come sfondo.

```
div
{
    background-image: url('uomo.jpg'), url('sfondo.jpg');
    background-position: bottom left, top left;
}
```

SFONDI

FORMA RIDOTTA - l'ordine delle proprietà è importante

background-image

background-repeat

```
div { background: #FFCC34 url('...') no-repeat top left; }
```

background-color

background-position

SFONDI (CSS 3)

BACKGROUND-SIZE:

```
div { background-size: 100px 120px; }
```

NUMERO / KEYWORD (COVER oppure CONTAIN)

Permette di specificare la dimensione dell'immagine di sfondo. Con i CSS 2 la dimensione era necessariamente quella dell'immagine per cui se si voleva utilizzare l'immagine in diversi punti con diverse dimensioni occorre duplicarla.

Con le parole chiave COVER e CONTAIN è possibile indicare che l'immagine debba essere utilizzata (con stretch in entrambe le direzioni) per occupare tutto il contenitore, oppure adattata (con eventuale stretch in una sola direzione).

BACKGROUND-ORIGIN:

```
div { background-origin: padding-box; }
```

KEYWORD

padding-box / border-box / content-box

Definisce il sistema di riferimento della proprietà background-position. Padding-box (predefinito) significa che parte dal padding, border-box significa che parte dal bordo, content-box significa che parte dal contenuto.

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Creare un foglio di stile per il vostro sito in modo tale che venga definito uno sfondo a tinta unita.

INTERMEDIO

Come novizio + uso di una immagine di sfondo replicata.

NERD

Come intermedio + uso di più immagini di sfondo.

OPERARE SUL TESTO

COLOR:

COLORE

Specifica il colore del font

```
a { color: blue }
```

TEXT-ALIGN:

KEYWORD

left / right / center / justify

Specifica l'allineamento del testo all'interno di un elemento di blocco

```
div { text-align: right; }
```

TEXT-DECORATION:

KEYWORD

none / overline / underline / line-through

Specifica la decorazione sul testo (nessuna / linea sopra / sottolineato / barrato)

```
a { text-decoration: none }
```

TEXT-TRANSFORM:

KEYWORD

uppercase / lowercase / capitalize

Specifica come trasformare il testo (maiuscole / minuscole / iniziale di ogni parola maiuscola)

```
a{ text-transform: capitalize }
```

OPERARE SUL TESTO

FONT-SIZE:

NUMERO

Definisce la dimensione del font

FONT-WEIGHT:

KEYWORD / NUMERO (SENZA UM)

normal / bold / numero può essere 100, 200, ..., 900 (400 = normal, 700 = bold)

Definisce lo spessore del testo (usato per il bold)

FONT-STYLE:

KEYWORD

normal / italic / oblique

Definisce l'inclinazione del testo (usato per il corsivo)

OPERARE SULLE LISTE PUNTATE

LIST-STYLE-TYPE:

KEYWORD

none / circle / disc / square / ...

Specifica il colore del font

```
li { list-style-type: none }
```

FONT

http://www.w3schools.com/cssref/css_websafe_fonts.asp

FONT-FAMILY:

STRINGHE, SEPARATE DA VIRGOLA

Permette di specificare una famiglia di font. I font **devono** essere installati sul computer dell'utente. È quindi consigliabile sempre specificare, come ultima, la famiglia generica (**serif / sans-serif**) in modo tale che nel caso peggiore (ovvero nessun font è installato sul pc dell'utente) venga usato il font di famiglia predefinito.

```
div { font-family: Verdana, Arial, sans-serif; }
```

Se l'utente ha Verdana,
viene usato quello.

Se l'utente non ha Verdana,
ma ha Arial, viene usato quello.

Se l'utente non ha nè Verdana,
nè Arial, viene usato il font sans-
serif predefinito sul computer
dell'utente.

@FONT-FACE (CSS3)

Con i CSS3 è finalmente possibile specificare dei font che non necessariamente sono installati sui computer dell'utente. Per poterlo fare è necessario utilizzare la direttiva @font-face.

```
@font-face
{
    font-family: ilmiofont;
    src:        url('Sansation_Norm.ttf'),
               url('Sansation_Norm.eot');
}
```

NOME DEL FONT (DEFINITO DA NOI)

FILE DEL FONT:

.EOT - SOLO IE9+

.TTF - TUTTI GLI ALTRI

```
div { font-family: ilmiofont; }
```

ATTENZIONE ALLE VERSIONI GRASSETTO E CORSIVO!
CONVIENE SEMPRE DEFINIRE DELLE ALTRE SPECIFICHE
@FONT-FACE CON LE RELATIVE SORGENTI GRASSETTO
E CORSIVO + DICHIARAZIONE:

FONT-WEIGHT: BOLD PER IL GRASSETTO

FONT-STYLE: ITALIC, OBLIQUE PER IL CORSIVO

```
@font-face
{
    font-family: ilmiofont;
    src:        url('Sansation_Bold.ttf'),
               url('Sansation_Bold.eot');
    font-weight: bold;
}
```

```
@font-face
{
    ...
    font-style: italic, oblique;
}
```

ATTENZIONE ALLA LICENZA!

I FONT SONO TUTELATI DAL DIRITTO D'AUTORE. ATTENZIONE ALLA LICENZA CHE ANCHE SE PERMETTE L'USO GRATUITO POTREBBE PROIBIRE L'USO IN UNA DIRETTIVA @FONT-FACE!

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

È UNA RACCOLTA DI FONT GRATUITI, ANCHE PER USO COMMERCIALE, MOLTI DEI QUALI UTILIZZABILI IN UNA DIRETTIVA @FONT-FACE.

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Creare un foglio di stile per il vostro sito in modo tale che venga definito uno sfondo a tinta unita e che il contenuto della header sia maiuscolo. Cambiare inoltre il font usando la direttiva font-family.

INTERMEDIO

Come novizio + rimozione sottolineatura ai link e cambio del colore in uno a scelta.

NERD

Come intermedio + header e footer con uno sfondo di colore a piacere. Definire, inoltre, l'uso di un font particolare

FLOAT

FLOAT:

KEYWORD

none / left / right

```
div { float: left; }
```

Permette ad un elemento di essere spinto a sinistra o a destra e di essere avvolto dagli elementi **che si trovano successivamente nel flusso**. Ciò significa che tutti gli elementi che si trovano **prima** nel flusso, non saranno per nulla toccati da questa proprietà.

Attenzione: è importante ricordarsi che l'elemento flottante viene rimosso dal flusso e viene spinto il più possibile a sinistra (se il valore è left) o a destra (se il valore è right) fin quando i suoi confini più esterni toccano i confini del contenitore oppure i confini esterni di un altro elemento flottante.

Se non c'è spazio in orizzontale per ospitare il blocco flottante, lo stesso scenderà di linea.



A floating image

Iste quidem veteres inter ponetur honeste, qui
vel mense brevi vel toto est iunior anno. Utor
permisso, caudaeque pilos ut equinae paulatim
vello unum, demo etiam unum, dum cadat elusus
Interdum volgus rectum videt, est ubi peccat.
Si veteres ita miratur laudatque poetas, ut
nihil anteferat, nihil illis comparet, errat.
Si quaedam nimis antequae, si peraque dure

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque
poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antequae
si peraque dure

FLOAT

RIASSUNTO E AVVERTIMENTI

1. IMPOSTARE SEMPRE UNA PROPRIETÀ WIDTH PER GLI ELEMENTI FLOTTANTI
2. GLI ELEMENTI CHE SI TROVANO PRIMA NEL FLUSSO NON VENGONO TOCCATI
3. GLI ELEMENTI CHE SI TROVANO DOPO NEL FLUSSO, AVRANNO IL CONTENUTO POSIZIONATO A SINISTRA O A DESTRA
4. GLI STESSI ELEMENTI, PERÒ, AVRANNO BORDO E SFONDO SOTTO ALL'ELEMENTO FLOTTANTE
5. SE UN CONTENITORE CONTIENE UN SOLO ELEMENTO FLOTTANTE, QUESTO STESSO CONTENITORE NON AVRÀ ALTEZZA (A MENO CHE NON GLI VENGA IMPOSTATA FORZOSAMENTE) POICHÈ L'ELEMENTO FLOTTANTE È STATO TOLTO DAL FLUSSO!

<http://css.maxdesign.com.au/floatutorial/introduction.htm>

RESETTARE IL FLOAT

CLEAR:

KEYWORD

none / left / right / both

```
div { clear: left; }
```

Permette all'elemento di essere posto nella linea successiva a qualunque elemento flottante a sinistra (left), oppure a destra (right) oppure in una qualunque delle due direzioni (both) che si trova nel precedentemente nel flusso.

ESEMPI DI LAYOUT CON FLOAT

1 COLONNA

X

= ORDINE NEL FLUSSO HTML

1

`<header>` : nessun float, nè clear

2

`<nav>` - nessun float, tranne che sui `<li>` della `<ul>` (in modo tale da farli venire uno di fianco all'altro)

3

`<section>`: nessun float, né clear,

4

`<footer>`

ESEMPI DI LAYOUT CON FLOAT

2 COLONNE

X

= ORDINE NEL FLUSSO HTML

1

`<header>` : nessun float, nè clear

2

`<nav>`
float: left
width: 200px

In questo modo, `<nav>`
si posizionerà a sinistra
e gli elementi
successivi si
posizioneranno alla sua
destra.

3

`<section>`: nessun float, né clear, margin-left:200px

Il margine serve perché così se vogliamo mettere
uno sfondo o un bordo questi non vadano sotto alla colonna.

Inoltre evitiamo che se il contenuto fosse più lungo della colonna,
questo vada sotto la stessa.

4

`<footer>` - clear: left

In questo modo, `<footer>` si posizionerà successivamente al termine verticale di `<nav>`

ESEMPI DI LAYOUT CON FLOAT

3 COLONNE

X

= ORDINE NEL FLUSSO HTML

1

`<header>` : nessun float, nè clear

2

`<nav>`
float: left
width: 150px

In questo modo, `<nav>` si posizionerà a sinistra e gli elementi successivi si posizioneranno alla sua destra.

4

`<section>`: nessun float, né clear,
margin-left: 150px, margin-right: 150px

3

`<aside>`
float: right
width: 150px

In questo modo, `<aside>` si posizionerà a destra e gli elementi successivi si posizioneranno alla sua sinistra.

5

`<footer>` - clear: both

In questo modo, `<footer>` si posizionerà successivamente al termine verticale di `<nav>` e `<aside>`

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Impostare il layout del sito a una, due o tre colonne

INTERMEDIO / NERD

Impostare il layout del sito a una, due o tre colonne, con bordi arrotondati

SPOSTARE LIBERAMENTE GLI ELEMENTI IN TUTTO IL SITO

POSITION:

KEYWORD

`static / fixed / relative / absolute`

Permette ad un elemento di essere posizionato in un qualunque punto del documento, secondo un sistema di riferimento di coordinate da **noi** determinato, estraendolo o meno dal flusso (e quindi intaccando o meno gli elementi successivi).

TOP / BOTTOM - RIGHT / LEFT:

NUMERO

Muove l'elemento, **rispetto al sistema di riferimento determinato dal valore di POSITION**, nelle direzioni indicate.

```
div { position: relative; top: 10px; left:15px; }
```

POSIZIONAMENTO ELEMENTI

STATIC

Valore predefinito, l'elemento si comporta come siamo abituati normalmente. L'elemento **non** è, quindi, estratto dal flusso.

FIXED

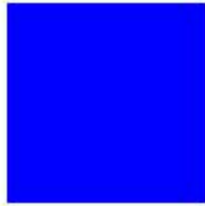
L'elemento viene posizionato rispetto alla finestra del browser e non si muoverà neanche quando la finestra scorre in orizzontale o in verticale. L'elemento viene estratto dal flusso.

ABSOLUTE

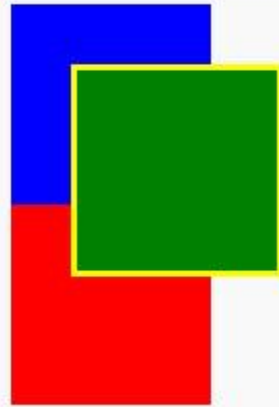
L'elemento viene posizionato rispetto al primo contenitore con un posizionamento **non** static. Se non ce ne sono, viene posizionato rispetto alla finestra del browser. L'elemento viene estratto dal flusso e può sovrapporsi ad altri elementi.

RELATIVE

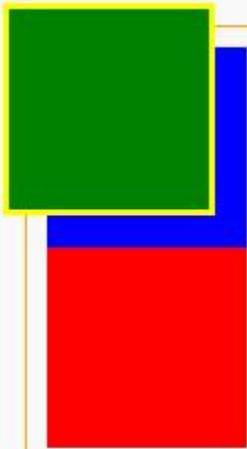
L'elemento viene posizionato rispetto a se stesso. L'elemento **non** è estratto dal flusso, ma l'ingombro occupato, però, è quello dell'elemento originale, non spostato. L'elemento può sovrapporsi ad altri elementi. Viene spesso usato per definire un sistema di riferimento da usare per elementi contenuti che sono in POSITION: ABSOLUTE.



← The second box is
set to “position: relative”
and is offset using
top and left values.



The second box is set to “position: absolute” and is offset in relation to the “.container” element



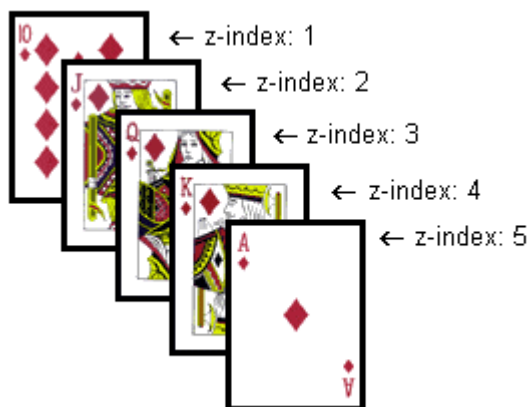
← The green box is now positioned in relation to the viewport

SOVRAPPOSIZIONE ELEMENTI

Z-INDEX:

NUMERO - SENZA UNITÀ DI MISURA

Definisce il livello di un elemento, quando questo si sovrappone ad altri. A numeri più grandi, corrispondono livelli più alti (più in fronte). Il valore può essere anche negativo. Se non specificato, sta in fronte chi viene dopo nel flusso.



**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Posizionare il logo del sito in un punto a piacere dell'intestazione, facendo uso della proprietà `position`.

INTERMEDIO

Come novizio + inserimento di una immagine di decorazione in fondo a sinistra con `position: static`.

NERD

Come intermedio + per ogni prodotto/servizio specificare due foto che verranno posizionate una sopra l'altra (con l'uso di `z-index`)

OMBRE SUL TESTO (CSS3)

NON SUPPORTATE DA IE

TEXT-SHADOW:

NUMERO, NUMERO, (COLORE), (NUMERO)

Definisce una ombreggiatura sul testo. I parametri obbligatori sono i primi due numeri che determinano lo scostamento orizzontale e verticale dell'ombreggiatura. Un valore positivo per il primo numero, indica uno scostamento verso destra, negativo verso sinistra. Un valore positivo per il secondo numero, indica uno scostamento verso il basso, negativo verso l'alto.

Se viene definito un terzo parametro, questo deve essere il colore. Se non è definito, viene usato il colore del testo.

Il quarto parametro è chiamato **blur** e determina la sfumatura dell'ombreggiatura.

Infine è possibile specificare più ombre ad un testo, separando valori di ombre diverse con la virgola.

OMBRE SUL TESTO (CSS3)

ESEMPI

Css3 Text Shadow

```
p { text-shadow: 5px 5px; }
```

Css3 Text Shadow

```
p { text-shadow: 5px 5px #666666; }
```

Css3 Text Shadow

```
p { text-shadow: 5px 5px #666666 6px; }
```

Css3 Text Shadow

```
p { text-shadow: #666666 4px 4px 8px, #00FDFD -2px -2px 4px; }
```

OMBRE SUI BOX (CSS3)

SUPPORTATA PIÙ O MENO DA TUTTI, CON PREFISSI

BOX-SHADOW:

(KEYWORD INSET), NUMERO, NUMERO, (COLORE), (NUMERO), (NUMERO)

Definisce una ombreggiatura di un box. I parametri sono simili a quelli di text-shadow, con l'aggiunta di due parametri:

il primo parametro - sempre opzionale - può assumere il solo valore **inset** per determinare una ombreggiatura interna al box. L'ultimo parametro - sempre opzionale - determina lo **spread** dell'ombreggiatura. Se il valore è positivo, determina che l'ombra si espanderà in tutte le direzioni di quel valore, se è negativo, invece, si ritirerà sempre in tutte le direzioni.

Attenzione perché sia Firefox che browser basati su Webkit (Opera e Chrome) necessitano del relativo prefisso, rispettivamente -moz- e -webkit-. Attenzione inoltre se non viene specificato il colore: il comportamento cambia da browser a browser.

Come per text-shadow, è possibile associare ombre multiple ai box, impostando valori diversi separati da virgola.

OMBRE SUI BOX (CSS3)

ESEMPI



```
.a { box-shadow: -5px -5px #888888; }
```

```
.b { box-shadow: -5px -5px 5px #888888; }
```

```
.c { box-shadow: -5px -5px 0px 5px #888888; }
```

```
.d { box-shadow: -5px -5px 5px 5px #888888; }
```

```
.e { box-shadow: 0px 0px 5px #888888; }
```

```
.f { box-shadow: 0px 0px 5px 5px #888888; }
```

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Applicare una ombreggiatura singola ai titoli di tutte le intestazioni.

INTERMEDIO

Come novizio + applicare una ombreggiatura al logo del sito.

NERD

Come intermedio + applicare una ombreggiatura al container di tutto il sito.

IMPOSTARE IL METODO DI VISUALIZZAZIONE DI UN ELEMENTO

DISPLAY:

KEYWORD

NONE: L'ELEMENTO NON VIENE VISUALIZZATO

BLOCK: L'ELEMENTO VIENE VISUALIZZATO COME ELEMENTO DI BLOCCO

INLINE: L'ELEMENTO VIENE VISUALIZZATO COME IN LINEA

INLINE-BLOCK: L'ELEMENTO VIENE VISUALIZZATO COME IN LINEA, MA SI COMPORTA COME BLOCCO (È POSSIBILE DARGLI UNA LARGHEZZA E UN'ALTEZZA, AD ESEMPIO)

... TANTI ALTRI

PSEUDO-CLASSI

Le pseudoclassi permettono di applicare una regola su un particolare **stato** di un elemento. Ciò significa che ad ogni selettore di una regola possiamo applicare una pseudoclasse e, quindi, quella particolare regola verrà applicata quando quel selettore si troverà in quello stato.

:FIRST-CHILD

INDIVIDUA IL PRIMO FIGLIO DIRETTO DI UN ELEMENTO PADRE. ATTENZIONE SE PERÒ L'ELEMENTO NON È IL PRIMO (OVVERO È PRECEDUTO DA UN ALTRO ELEMENTO, ANCHE DI TIPO DIVERSO) QUESTA PSEUDOCLASSE NON SARÀ APPLICATA!!!



È importante il due punti.

I nomi delle pseudoclassi sono prefissati, ovvero non possiamo definire nostre pseudoclassi.

PSEUDO-CLASSI - ESEMPIO 1

```
<div>  
  <p>Paragrafo esterno</p>  
</div>  
<div>  
  <p>1° paragrafo</p>  
  <p>2° paragrafo</p>  
  <p>3° paragrafo</p>  
</div>
```

NESSUNA PSEUDOCLASSE.

LA REGOLA SI APPLICA A **TUTTI** I <P>

```
p { color: blue; }
```



PSEUDO-CLASSI - ESEMPIO 2

```
<div>
  <p>Paragrafo esterno</p>
</div>
<div>
  <p>1° paragrafo</p>
  <p>2° paragrafo</p>
  <p>3° paragrafo</p>
</div>
<div>
  <a href="#">Vai</a>
  <p>Non selezionato</p>
</div>
```

PSEUDOCLASSE ASSOCIATA DIRETTAMENTE A P

LA REGOLA SI APPLICA A **TUTTI** I <P> CHE SONO PRIMI FIGLI (IN SENSO ASSOLUTO, OVVERO ANCHE IN PRESENZA DI ALTRI ELEMENTI HTML) DI UN QUALUNQUE ELEMENTO HTML

```
p:first-child { color: blue;
}
```

QUESTO <P> NON È UN PRIMO FIGLIO! C'È IL TAG <A> PRIMA!!!

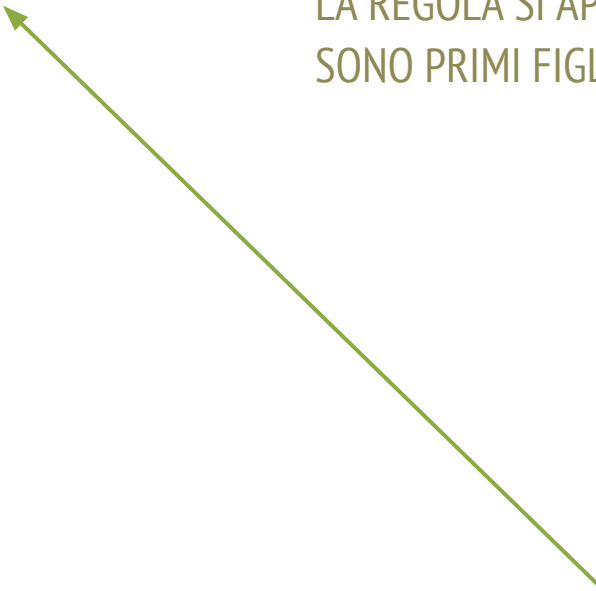
PSEUDO-CLASSI - ESEMPIO 3

```
<div>
  <p>Paragrafo esterno</p>
</div>
<div id="test">
  <p>1° paragrafo</p>
  <p>2° paragrafo</p>
  <p>3° paragrafo</p>
</div>
```

PSEUDOCLASSE ASSOCIATA DIRETTAMENTE A P

LA REGOLA SI APPLICA A **TUTTI** I <P> CHE
SONO PRIMI FIGLI DI UN ELEMENTO DIV

```
#test p:first-child { color: blue; }
```



PSEUDO-CLASSI

Se le inserite tutte, possono esserci conflitti sui link! Inseritele in questo ordine per essere sicuri che vengano applicate correttamente!

:LINK

È VERIFICATA DA TUTTI I LINK CHE NON SONO MAI STATI VISITATI
SI PUÒ APPLICARE SOLO AGLI ELEMENTI <A>

```
a:link { color: blue; }
```

:VISITED

È VERIFICATA DA TUTTI I LINK CHE SONO STATI VISITATI
SI PUÒ APPLICARE SOLO AGLI ELEMENTI <A>

```
a:visited { color: red; }
```

:HOVER

SI VERIFICA QUANDO IL PUNTATORE DEL MOUSE SI POSIZIONA SOPRA L'ELEMENTO
SI PUÒ APPLICARE AD OGNI ELEMENTO

```
a:hover { color: red; }
```

:FOCUS

SI VERIFICA QUANDO L'ELEMENTO DETIENE IL FOCUS
SI PUÒ APPLICARE AD OGNI ELEMENTO

```
input:focus { color: red; }
```

PSEUDO-CLASSI (CSS 3)

<http://reference.sitepoint.com/css/css3psuedoclasses>

:FIRST-OF-TYPE

INDIVIDUA IL PRIMO FIGLIO DIRETTO DI UN ELEMENTO PADRE. IN QUESTO CASO, VIENE PRESO IL PRIMO ANCHE SE PRECEDENTEMENTE ESISTONO ALTRI ELEMENTI DI ALTRI TIPI.

```
p:first-of-type { color: blue; }
```

:LAST-CHILD

INDIVIDUA L'ULTIMO FIGLIO DIRETTO DI UN ELEMENTO PADRE. COME PER FIRST-CHILD, NON VIENE APPLICATO SE NON È VERAMENTE L'ULTIMO CONTANDO ANCHE ELEMENTI DI ALTRO TIPO!

```
p:last-child { color: blue; }
```

:LAST-OF-TYPE

INDIVIDUA L'ULTIMO FIGLIO DIRETTO DI UN ELEMENTO PADRE. IN QUESTO CASO, VIENE PRESO L'ULTIMO ANCHE SE SUCCESSIVAMENTE ESISTONO ALTRI ELEMENTI DI ALTRI TIPI.

```
p:last-of-type { color: blue; }
```

PSEUDO-CLASSI (CSS 3)

p:first-child { ... } - NESSUNO!

p:last-child { ... } - NESSUNO!

```
<div>
  
  <p>1° paragrafo</p>
  
  <p>2° paragrafo</p>
  <p>3° paragrafo</p>
  
</div>
```

p:first-of-type { ... }

p:last-of-type { ... }

PSEUDO-CLASSI (CSS 3)

:NTH-CHILD(*n*)

È VERIFICATA DA TUTTI GLI ELEMENTI CHE SONO N-ESIMI FIGLI DI UN ELEMENTO. POSSONO ESSERE SCRITTE ESPRESSIONI NUMERICHE!

INDISPENSABILE QUANDO VOLETE RAPPRESENTARE IN MODO DIVERSO ELEMENTI PARI E DISPARI (ESEMPIO IN UNA LISTA)

SELEZIONA OGNI SECONDO <P> FIGLIO DI QUALUNQUE ELEMENTO HTML

```
p:nth-child(2) { color: blue; }
```

SELEZIONA OGNI <P> FIGLIO PARI DI QUALUNQUE ELEMENTO HTML

```
p:nth-child(2n) { color: black; }  
p:nth-child(even) { color: black; }
```

SELEZIONA OGNI <P> FIGLIO DISPARI DI QUALUNQUE ELEMENTO HTML

```
p:nth-child(n) { color: red; }  
p:nth-child(odd) { color: red; }
```

ATTENZIONE! NEL CONTEGGIO DELLA PARITÀ / DISPARITÀ DEI <P> RIENTRANO ANCHE ELEMENTI DI TIPO DIVERSO CHE SI TROVANO ALLO STESSO LIVELLO!

:NTH-OF-TYPE(*n*)

COME IL PRECEDENTE, CON LA DIFFERENZA CHE NEL CONTEGGIO DI PARITÀ / DISPARITÀ NON RIENTRANO ELEMENTI ALLO STESSO LIVELLO MA DI TIPO DIVERSO

PSEUDO-CLASSI (CSS 3)

```
<div>
  <p>1° paragrafo</p>
  
  <p>2° paragrafo</p>
  <p>3° paragrafo</p>
</div>
```

p:nth-child(odd) { ... }

p:nth-of-type(odd) { ... }

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Cambiare lo stile dei link quando il mouse si posiziona sopra (ad esempio, rimuovere sottolineatura e cambiare colore).

INTERMEDIO

Come novizio + applicare una immagine di sfondo agli input quando hanno il focus.

NERD

Come intermedio + applicare uno stile diverso agli elementi pari e dispari del menu, facendo uso delle pseudoclassi.

PSEUDO-ELEMENTI

Gli pseudo-elementi permettono di identificare elementi che **non** esistono esplicitamente nel documento HTML.

:FIRST-LETTER

INDIVIDUA LA PRIMA LETTERA ALL'INTERNO DI UN TESTO. ALCUNI CARATTERI DI INTERPUNZIONE VENGONO COMUNQUE INTESI COME PARTE DELLA PRIMA LETTERA.

```
p:first-letter { font-size: 200% }
```

:FIRST-LINE

INDIVIDUA LA PRIMA LINEA ALL'INTERNO DI UN TESTO

```
p:first-line { font-size: 150%; }
```

PSEUDO-ELEMENTI

:BEFORE

INSERISCE DEL CONTENUTO (TRAMITE LA PROPRIETÀ CONTENT)
PRIMA DI UN DETERMINATO ELEMENTO

```
#briciole:before { content: "Sei qui:" }
```

:AFTER

INSERISCE DEL CONTENUTO (TRAMITE LA PROPRIETÀ CONTENT) DOPO UN
DETERMINATO ELEMENTO

```
p:after { content: "Finito" }
```

OVERFLOW (-X E -Y)

Con la proprietà overflow possiamo decidere il comportamento del contenuto quando questo stesso esce dai confini del box che lo contiene (e che, quindi, deve avere dimensioni fissate).

L'overflow può essere settato quando esce in verticale (**overflow-y**) oppure in orizzontale (**overflow-x**), oppure su entrambi i lati (**overflow**).

OVERFLOW:

KEYWORD

visible / hidden / scroll / auto

VISIBLE = valore predefinito, il testo esce dal box e rimane visibile

HIDDEN - il testo che esce dai bordi del contenitore viene nascosto

SCROLL - il contenitore fa comparire delle barre di scorrimento che vengono abilitate quando necessario

AUTO - il contenitore fa comparire delle barre di scorrimento solo quando necessario

TEXT-OVERFLOW (CSS 3)

Supera un limite dell'overflow, ovvero quello che determina il clipping (ritaglio) del testo quando questo esce dal proprio contenitore. Con questa proprietà è possibile impostare un carattere di rimpiazzamento per il testo che non viene visualizzato, ad esempio "...".

OVVIAMENTE QUESTA PROPRIETÀ È DA USARE IN COMBINAZIONE CON LA PROPRIETÀ OVERFLOW!

TEXT-OVERFLOW:

KEYWORD

clip / ellipsis

CLIP = il testo viene tagliato

ELLIPSIS - viene inserito un carattere "ellipsis", ovvero tre puntini prima di tagliare il testo

OPACITY (CSS 3)

Permette di settare l'opacità (ovvero la trasparenza) di un elemento

OPACITY:

NUMERO (SENZA UM) COMPRESO TRA 0 E 1

COMPLETAMENTE
TRASPARENTE

COMPLETAMENTE
OPACO

```
div { opacity: 0.5 }
```

GRADIENTI (CSS3)

Permette di impostare sfondi che cambino di colore. Esistono sostanzialmente due tipi di gradienti:
LINEARI, RADIALI

```
div { BACKGROUND: LINEAR-GRADIENT(...) }
```

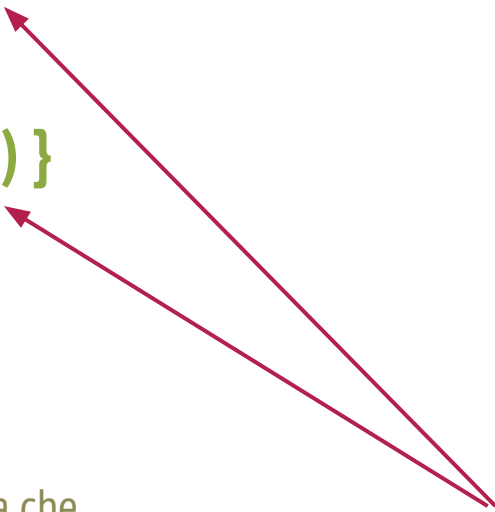
PER I GRADIENTI LINEARI

```
div { BACKGROUND: RADIAL-GRADIENT(...) }
```

PER I GRADIENTI RADIALI

ATTENZIONE!

I gradienti sono ancora in forte sviluppo, per cui ogni volta che vengono utilizzati **dovete** fare uso della specifica del browser relativa, ovvero: `-webkit-linear-gradient(...)`, `-moz-linear-gradient(...)`, `-ms-linear-gradient(...)`, `-o-linear-gradient(...)` e così via



**DENTRO LA PARENTESI CI SONO
DEI PARAMETRI!**

GRADIENTI LINEARI (CSS 3)

```
div { background: linear-gradient(from-stop, to-stop); }  
div { background: linear-gradient(#FFF, #000); }
```

PARAMETRI RICHIESTI, AL MINIMO, SONO DUE COLORI DI INIZIO E FINE DEL GRADIENTE



```
div { background: linear-gradient(point, from-stop, to-stop);  
}
```

```
div { background: linear-gradient(left, #FFF, #000); }
```

POINT STABILISCE IL PUNTO DI PARTENZA DEL GRADIENTE. PREVEDE DUE PAROLE CHIAVE A SCELTA TRA (TOP/BOTTOM/CENTER E LEFT/RIGHT/CENTER). SE UNA NON VIENE SPECIFICATA È INTESA COME CENTER. SE ENTRAMBE NON VENGONO SPECIFICATE VIENE INTESO COME CENTER TOP.



point = left oppure left center

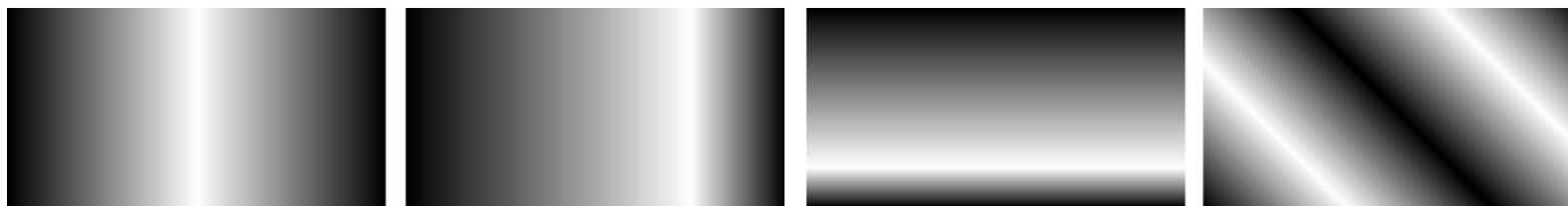
point = right oppure right center

point = right top

GRADIENTI LINEARI (CSS 3)

```
div { background: linear-gradient(point, from-stop, stop-1, ..., stop-n, to-stop); }
```

È POSSIBILE STABILIRE PIÙ COLORI DI STOP. IL COLORE DI STOP SI TROVERÀ ESATTAMENTE A METÀ TRA I DUE COLORI FROM-STOP E TO-STOP.



```
.a { background: linear-gradient(left, #000, #FFF, #000); }
```

```
.b { background: linear-gradient(left, #000, #FFF 75%, #000); }
```

```
.c { background: linear-gradient(bottom, #000, #FFF 20px, #000); }
```

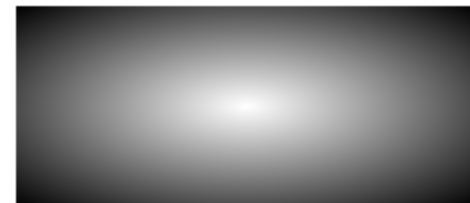
SENZA SEPARARLO CON VIRGOLE, È POSSIBILE SPECIFICARE UN NUMERO (PERCENTUALE O PX) CHE STABILISCA A QUALE PUNTO DEL GRADIENTE VENGA INSERITO IL COLORE DI STOP. ATTENZIONE, QUINDI, SE STABILITE PIÙ DI 3 COLORI DI STOP, PERCHÈ POTREBBERO SOVRAPPORSI!

```
.d { background: linear-gradient(left bottom, #000, #FFF, #000, #FFF, #000); }
```

GRADIENTI RADIALI (CSS 3)

```
div { background: radial-gradient(from-stop, to-stop); }  
div { background: radial-gradient(#FFF, #000); }
```

PARAMETRI RICHIESTI, AL MINIMO, SONO DUE COLORI DI INIZIO E FINE DEL GRADIENTE

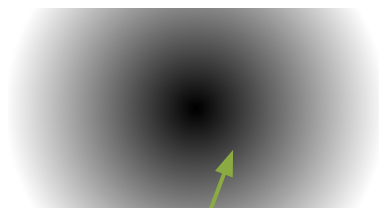


```
div { background: linear-gradient(point, shape size, from-stop, to-stop); }  
div { background: linear-gradient(left, circle cover, #FFF, #000); }
```

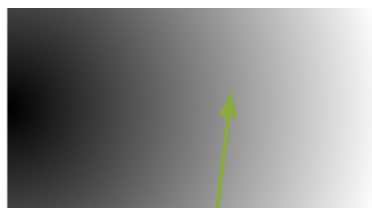
POINT STABILISCE IL PUNTO DI PARTENZA DEL GRADIENTE (come per i linear)

SHAPE STABILISCE SE IL GRADIENTE DEBBA ESSERE CIRCOLARE (CIRCLE) O ELLITTICO (ELLIPSE, PREDEFINITO)

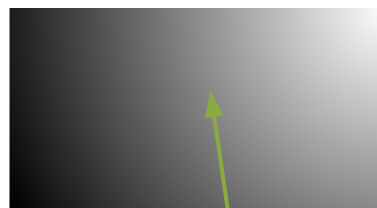
SIZE STABILISCE LA DIMENSIONE DEL GRADIENTE, TRAMITE UNA PAROLA CHIAVE A SCELTA FRA 4



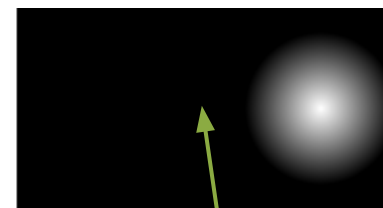
**SHAPE: CIRCLE
SIZE: FARTEST-SIDE**



**POINT: LEFT CENTER
SHAPE: CIRCLE
SIZE: FARTEST-SIDE**



**POINT: RIGHT TOP
SHAPE: CIRCLE
SIZE: COVER**



**POINT: 80% 50%
SHAPE: CIRCLE
SIZE: CONTAIN**

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Modificare lo sfondo dell'intestazione in modo tale che faccia uso di un riempimento gradiente (lineare o radiale a scelta).

INTERMEDIO

Come novizio + modificare l'opacità di un qualche box all'interno della sezione di contenuto.

NERD

Come intermedio + impostare una dimensione fissa sui box dei prodotti / servizi e tagliare il testo che esce dal box usando overflow e text-overflow.

TRANSIZIONI (CSS 3)

Le transizioni sono effetti che permettono ad un elemento di passare gradualmente da uno stile associato ad un altro. È prerequisite, quindi, avere nel foglio di stile due regole che si applicano in due stati diversi (di cui uno, tipicamente, è **:hover**) e che agiscono su proprietà che vogliamo rendere transienti.

TRANSITION:

CARATTERISTICA DA MODIFICARE, TEMPO TRANSIZIONE, FUNZIONE DI TIMING, RITARDO

es. background, opacity, width...

NON LE VEDIAMO

ESPRESSO IN SECONDI, NELLA
FORMA XS, DOVE X È UN
NUMERO E S RIMANE TALE.
INDICA LA DURATA DELLA
TRANSIZIONE

ESPRESSO IN SECONDI, NELLA
FORMA XS, DOVE X È UN
NUMERO E S RIMANE TALE.
INDICA QUANTO TEMPO SARÀ
FATTO PASSARE PRIMA DI
SCATENARE LA TRANSIZIONE

TRANSIZIONI (CSS 3)

Anche in questo caso **utilizzate sempre i prefissi dei browser:**

-webkit-transition, -moz-transition, -o-transition. Internet Explorer (<10) non supporta attualmente le transizioni.

```
<div class="animato">  
    Questo box è animato.  
</div>
```

```
.animato {  
    width: 100px; height: 100px;  
    border: 1px solid black;  
    background: red;  
    transition: background 2s;  
    -webkit-transition: background 2s;  
}  
  
.animato:hover { background: yellow; }
```

TRANSIZIONI MULTIPLE (CSS 3)

Se si vuole variare più proprietà contemporaneamente, la proprietà transition **non** è sufficiente. Bisogna fare uso della versione "esplosa" che è composta da 4 proprietà diverse:

TRANSITION-PROPERTY:

ELENCO DI PROPRIETÀ DA VARIARE, SEPARATE DA VIGOLA

TRANSITION-DURATION:

ELENCO DI TEMPI RAPPRESENTANTI LA DURATA DI OGNI PROPRIETÀ DA VARIARE, SEPARATI DA VIRGOLE

TRANSITION-DELAY:

ELENCO DI TEMPI RAPPRESENTANTI LA DURATA DEL RITARDO DI OGNI PROPRIETÀ DA VARIARE, SEPARATI DA VIRGOLE

TRANSITION-TIMING-FUNCTION:

NON LA VEDIAMO

TRANSIZIONI MULTIPLE (CSS 3)

Anche in questo caso **utilizzate sempre i prefissi dei browser:**

-webkit-transition, -moz-transition, -o-transition. Internet Explorer (<10) non supporta attualmente le transizioni.

```
<div class="animato">  
    Questo box è animato.  
</div>
```

```
.animato {  
    width: 100px; height: 100px;  
    border: 1px solid black;  
    background: red;  
    transition-property: background, width;  
    transition-duration: 2s, 1s;  
    -webkit-transition-property: background, width;  
    -webkit-transition-duration: 2s, 1s;  
}  
  
.animato:hover { background: yellow; width: 200px }
```

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Applicare una transizione singola all'hover sul logo (es. rimuovere la box-shadow)

INTERMEDIO

Come novizio + applicare una transizione all'hover sulle immagini dei prodotti, in modo tale che cambi lo spessore del bordo .

NERD

Come intermedio + applicare un'altra transizione all'hover sulle immagini dei prodotti, in modo tale che cambi la box-shadow (o altro).

TRASFORMAZIONI 2D (CSS 3)

Con le trasformazioni è possibile, appunto, trasformare un qualunque elemento, ruotandolo, traslandolo, scalandolo e così via. Le trasformazioni possono essere a 2D oppure 3D. Vedremo solo le trasformazioni 2D.

TRANSFORM:

FUNZIONE(X,Y)

LE FUNZIONI UTILIZZABILI SONO:

1. **TRANSLATE** - si comporta come `position: relative`. I due valori sono lo scostamento rispetto alla sua posizione. L'ingombro è quello che era prima della trasformazione.

Esempio: `transform: translate(20px, 20px);`

2. **SCALE** - applica uno zoom o comprime un elemento. I due valori rappresentano lo zoom (se > 1) o la compressione (se compresi tra 0 e 1) sull'asse x e sull'asse y. Non deve essere specificata l'unità di misura. L'ingombro è sempre quello dell'elemento non trasformato.

Esempio: `transform: scale(3,3);`

TRASFORMAZIONI 2D (CSS 3)

TRANSFORM:

FUNZIONE(X,Y)

LE FUNZIONI UTILIZZABILI SONO:

3. **ROTATE** - Ruota l'elemento in senso orario (angolo positivo) o in senso antiorario (angolo negativo). Il valore è uno solo e deve essere seguito dall'"unità di misura" deg (senza spazi). L'ingombro è sempre quello dell'elemento non trasformato.

Esempio: transform: rotate(45deg);

4. **SKEW** - Inclina e distorce l'elemento. I parametri sono due e sono gli angoli di distorsione sull'asse x e sull'asse y. L'ingombro è sempre quello dell'elemento non trasformato.

Esempio: transform: skew(10deg,30deg);

PER CONCATENARE PI Û TRASFORMAZIONI, SEPARARLE DA UNO SPAZIO.

ESEMPIO: TRANSFORM: ROTATE(20DEG) SCALE(50PX)

TRASFORMAZIONI 2D (CSS 3)

Le trasformazioni hanno come punto di applicazione, se non diversamente definito, il punto centrale del box. Se vogliamo cambiare questo comportamento facciamo uso della proprietà transform-origin.

TRANSFORM-ORIGIN:

TOP / BOTTOM , LEFT RIGHT

<http://www.sitepoint.com/a-primer-on-css3-transforms/>

**SE VE LO STATE
CHIEDENDO...**

Sì!

**POTETE APPLICARE
TRANSIZIONI SULLE
TRASFORMAZIONI!**

ESEMPI

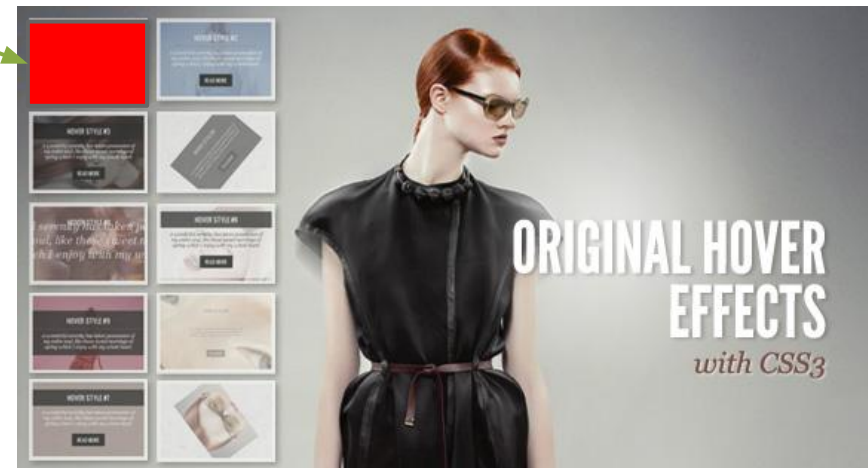
BOX INFORMATIVI ANIMATI

MARKUP

```
<div class="view">
  
  <div class="mask">
    <h2>Titolo</h2>
    <p>Testo descrittivo</p>
    <a href="#" class="info">Leggi...</a>
  </div>
</div>
```

Ognuno di quei box, sulla sinistra, ha il markup descritto sopra:

1. immagine
2. titolo + testo + link



BOX INFORMATIVI ANIMATI

STILE DI BASE DEL BOX

```
.view
{
    width: 300px;
    height: 200px;
    border: 10px solid #fff;
    overflow: hidden;
    position: relative;
    text-align: center;
    box-shadow: 1px 1px 2px #e6e6e6;
    background: #FFFFFF;
}
```

BOX INFORMATIVI ANIMATI

STILE DI BASE DEL CONTENUTO

```
.view .mask
{
  width: 300px;
  height: 200px;
  position: absolute;
  top: 0;
  left: 0;
  overflow: hidden;
}
```

BOX INFORMATIVI ANIMATI

STILE DI BASE DEL TESTO

```
.view h2
{
  text-transform: uppercase;
  color: #FFFFFFF;
  text-align: center;
  background: rgba(0,0,0,0.8);
}
```

```
.view p
{
  font-style: italic;
  font-size: 12px;
  color: #FFFFFFF;
  text-align: center;
}
```

```
.view .info
{
  display: inline-block;
  text-decoration: none;
  background: #000000;
  color: #FFFFFFF;
  text-transform: uppercase;
  box-shadow: 0 0 1px
#000000;
  padding: 7px 14px;
}
```

BOX INFORMATIVI ANIMATI

DOPO AVER DEFINITO LO STATO "FINALE", MODIFICHIAMO LO STILE IN MODO TALE DA PORTARCI NELLO STATO "INIZIALE".

```
.view .mask
{
    ...
    opacity: 0;
    background-color: rgba(219,127,8, 0.7);
}
```

In questo modo la maschera diventa di uno sfondo color arancione, ma completamente trasparente. Avendo usato opacity, tutto il contenuto diventa trasparente e, quindi, non si vede.

```
.view:hover .mask
{
    opacity: 1;
}
```

BOX INFORMATIVI ANIMATI

A QUESTO PUNTO GLI EFFETTI!

```
.view:hover img { transform: scale(1.1); }
```

```
.view h2 { transform: translateY(-100px); }  
.view:hover h2 { transform: translateY(0px); }
```

```
.view p { transform: translateY(100px); }  
.view:hover p { transform: translateY(0px); }
```

```
.view .info { transform: translate(250px, -100px) rotate(90deg)  
}  
.view:hover .info { transform: translate(0) rotate(0) }
```

BOX INFORMATIVI ANIMATI

MANCANO SOLO PIÙ LE TRANSIZIONI!

```
.view img { transition: all 1s; }
```

```
.view h2{ transition: all 0.5s; }
```

```
.view p{ transition: all 0.5s; }
```

```
.view .info{ transition: all 0.5s; }
```

```
.view .mask{ transition: all 0.5s; }
```

ESEMPIO DIDATTICO COMPLETO

CSS BASE + POSITION + FLOAT + TRANSITION + TRANSFORM + FONT-FACE
+ PSEUDOCASSI

<http://net.tutsplus.com/tutorials/html-css-techniques/create-a-sticky-note-effect-in-5-easy-steps-with-css3-and-html5/>

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

**Definire una trasformazione sul logo (es. rotazione) quando si è in hover.
Trasformare questa proprietà con una transizione.**

INTERMEDIO

Come novizio + applicare una trasformazione (e transizione) sulle immagini dei prodotti quando si è in hover.

NERD

Come intermedio + applicare una serie di trasformazioni (es. scale + rotate) sulle immagini dei prodotti, da resettare (o applicare) quando si è in hover.

ANIMAZIONI (CSS 3)

Con i CSS3 abbiamo visto che è possibile far passare un elemento da una classe ad un'altra in modo smorzato. Tutto questo è molto bello, ma possiamo ottenere ancora di più! Come? Facendo uso delle animazioni! Anche se non esplicitato, ricordate **sempre** i prefissi -webkit-, -moz-, ecc. ecc.

@keyframes *nomeanimazione* { ... }

Con la regola @keyframes (@-webkit-keyframes, ...) possiamo definire un'animazione, facendo uso dei concetti di keyframe (chi conosce un po' di Flash probabilmente sa di cosa si parla).

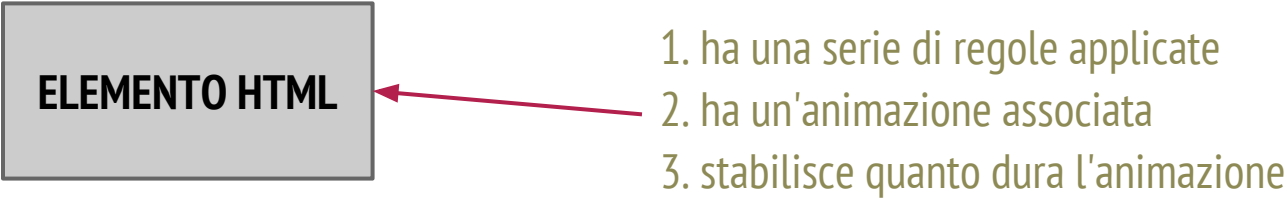
In altre parole, all'interno della parentesi graffa vengono definiti i cosiddetti **keyframe** (ovvero frame chiave) che non sono altro che punti chiave della nostra animazione. L'animazione, quindi, si comporta come un susseguirsi di transizioni tra un keyframe ed un altro.

Inoltre è importante notare che l'animazione è referenziata da un nome (**nomeanimazione**) che sarà poi usato, come vedremo, all'interno del foglio di stile.

ANIMAZIONI (CSS 3)

```
@keyframe sunrise
{
    0% { bottom: 0; left: 340px; background: #FF0000; }
    33% { bottom: 340px; left: 340px; background: #FFD630; }
    66% { bottom: 340px; left: 40px; background: #FFD630; }
    100% { bottom: 0; left: 40px; background: #FF0000; }
}
```

0%, 33%, 66% e 100% rappresentano i **keyframe** ovvero delle "fotografie" dell'elemento, le cui proprietà sono quelle definite dal keyframe stesso. Ogni singola fotografia si posiziona in un punto **relativo** dell'animazione, la cui durata è stabilita dall'elemento stesso! I periodi "intermedi" tra un keyframe e un altro si comportano come una normale transizione.



ELEMENTO HTML

1. ha una serie di regole applicate
2. ha un'animazione associata
3. stabilisce quanto dura l'animazione

ANIMAZIONI (CSS 3)

Per associare un'animazione ad un elemento, si fa uso della proprietà **animation-name** che prende come valore il nome di un'animazione definita nel foglio di stile.

esempio:

```
.sun { animation-name: sunrise; }
```

animation-duration

NUMERO (CON UNITÀ DI MISURA S)

Stabilisce la durata (in secondi) dell'animazione. Esempio: animation-duration: 10s;

animation-iteration-count

NUMERO (SENZA UNITÀ DI MISURA), OPPURE INFINITE

Stabilisce il numero di volte che l'animazione dovrà essere eseguita.

animation-direction

KEYWORD (NORMAL / REVERSE / ALTERNATE)

Quando l'animazione finisce, stabilisce se deve essere eseguita di nuovo dall'inizio (normale) oppure se deve essere eseguita al contrario (reverse).

MEDIA

PRE-CSS 3

**DIVERSI TIPI DI
MEDIA, MA
PREFISSATI E
NON MODIFICABILI.**

POSSONO ESSERE SPECIFICATI PIÙ DI UN MEDIA, MA DEVONO ESSERE
SEPARATI DA VIRGOLA

ALL: tutti

SCREEN: schermi computer

PRINT: stampanti

PROJECTION: proiettori

AURAL: sintetizzatori vocali

BRaille: barre braille

HANDHELD: palmari

...

```
<link rel="stylesheet" media="screen" type="text/css" href="css/stile.css" />  
<link rel="stylesheet" media="print" type="text/css" href="css/stile_stampa.css"  
>
```

CSS 3

MEDIA QUERIES

Permette di specificare che:

a) un intero foglio di stile, oppure

b) un insieme di regole

vengano applicate solo a certe condizioni che il media deve rispettare.

WIDTH / HEIGHT

DEVICE-WIDTH / DEVICE-HEIGHT

ORIENTATION

COLOR

RESOLUTION

...

MEDIA QUERIES (CSS 3)

WIDTH / MAX-WIDTH / MIN-WIDTH

NUMERO IN PX

Stabilisce quale debba essere la larghezza / massima / minima della finestra.

DEVICE-WIDTH / MAX-DEVICE-WIDTH / MIN-DEVICE-WIDTH

NUMERO IN PX

Stabilisce quale debba essere la larghezza / massima / minima dello schermo.

HEIGHT / MAX-HEIGHT / MIN-HEIGHT

NUMERO IN PX

Stabilisce quale debba essere l'altezza / massima / minima della finestra.

DEVICE-HEIGHT / MIN-DEVICE-HEIGHT / MAX-DEVICE-HEIGHT

NUMERO IN PX

Stabilisce quale debba essere l'altezza / minima / massima dello schermo.

ORIENTATION

LANDSCAPE (ORIZZONTALE) / PORTRAIT (VERTICALE)

Stabilisce come debba essere orientato lo schermo.

MEDIA QUERIES (CSS 3)

È possibile impostare che il foglio di stile debba essere interamente applicato se una certa media query è verificata:

```
<link rel="stylesheet" media="min-width: 400px" type="text/css" href="stile.css" />
```

Oppure è possibile definire delle sottosezioni all'interno del foglio di stile che si applicano quando una certa media query è verificata:

```
.miaclasse
{
    background: #CCC;
    width: 200px;
    height: 200px;
}

@media (max-width: 700px)
{
    .miaclasse
    {
        width: 100px;
        height: 100px;
        background: red;
    }
}
```

MEDIA QUERIES (CSS 3)

Inoltre è possibile combinare più proprietà richieste dal nostro particolare media separando le proprietà dalla parola chiave "AND".

```
<link rel="stylesheet" media="(min-width: 400px) and (max-width: 800px)" type="text/css" href="stile.css" />
```

Vale sempre il discorso sulle regole all'interno del foglio di stile.

```
.miaclasse
{
    background: #CCC;
    width: 200px;
    height: 200px;
}

@media (max-width: 700px) and (min-width: 200px)
{
    .miaclasse
    {
        width: 100px;
        height: 100px;
        background: red;
    }
}
```



COS'È JQUERY

JQUERY È UN FRAMEWORK JAVASCRIPT, OVVERO UNA LIBRERIA DI CODICE SCRITTO IN JAVASCRIPT IL CUI SCOPO È QUELLO DI PERMETTERE ALL'UTENTE DI SCRIVERE CODICE CHE COMPIE OPERAZIONI COMPLESSE IN MANIERA SEMPLICE.

CARATTERISTICA FONDAMENTALE DI JQUERY È QUELLA DI PERMETTERCI DI TRASCURARE LE DIVERSITÀ DI INTERPRETAZIONE DEL CODICE JAVASCRIPT DA PARTE DI OGNI BROWSER.

INTEGRARE JQUERY

SEMPLICE:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

ALL'INTERNO DEL TAG HEAD!

IN QUESTO CASO STIAMO FACENDO USO DELLA LIBRERIA JQUERY OSPITATA SUI SERVER DI GOOGLE. IN ALTERNATIVA POSSIAMO SCARICARE LA LIBRERIA DAL SITO DI JQUERY E COLLEGARLA IN MANIERA SIMILARE:

```
<script src="js/jquery.min.js"></script>
```

USARE JQUERY

COME PER I FOGLI DI STILE, ANCHE PER JAVASCRIPT POSSIAMO SCRIVERE IL CODICE DIRETTAMENTE NELLA PAGINA HTML, OPPURE

PER MAGGIORE PULIZIA

POSSIAMO COLLEGARE UN NOSTRO DOCUMENTO JAVASCRIPT.

```
<script src="js/scripts.js"></script>
```

ALL'INTERNO DEL TAG HEAD, DOPO IL TAG SCRIPT DI JQUERY.

USARE JQUERY

ALL'INTERNO DEL NOSTRO CODICE POSSIAMO FARE USO DELLA FUNZIONE \$, DEFINITA DA JQUERY.

CON LA FUNZIONE \$ POSSIAMO ACCEDERE A QUALUNQUE ELEMENTO HTML (O QUALUNQUE INSIEME DI ELEMENTI) DEFINITO NELLA NOSTRA PAGINA, INSERENDO AL SUO INTERNO IL SELETTORE CHE LO IDENTIFICA.
ESEMPIO:

ESEMPIO 1 - TUTTI I DIV

```
$ ("div")
```

ESEMPIO 2 - ELEMENTO CON ID 'WRAPPER'

```
$ ("#wrapper")
```

ESEMPIO 3 - ELEMENTI CON CLASSE 'MYCLASS'

```
$ (".myclass")
```

ESEMPIO 4 - IMG FIGLIE DI ID 'GALLERY'

```
$ ("#gallery img")
```

USARE JQUERY

UNA VOLTA SELEZIONATO UNO O PIÙ ELEMENTI, UTILIZZANDO LA FUNZIONE \$, È POSSIBILE CONCATENARE ALTRE FUNZIONI. NE VEDIAMO ALCUNE:

- CLICK
- ADDCLASS / REMOVECLASS / TOGGLECLASS
- FADEIN / FADEOUT / FADETOGGLE / SHOW / HIDE

ESEMPI:

```
$ (".myclass") .show () ;  
$ ("#wrapper") .addClass ('myclass') ;
```

LA PAGINA DEVE ESSERE PRONTA!

UNA COSA IMPORTANTE DA RICORDARE È CHE IL CODICE JAVASCRIPT (PER CUI ANCHE QUELLO JQUERY) PUÒ ESSERE ESEGUITO ANCOR PRIMA CHE LA PAGINA SIA STATA EFFETTIVAMENTE CARICATA.

PER PREVENIRE QUESTO, OCCORRE INSERIRE IL NOSTRO CODICE ALL'INTERNO DI UN BLOCCO CHE VERIFICA CHE LA PAGINA SIA STATA CARICATA INTERAMENTE:

```
$(document).ready(function()  
{  
    //Qui andremo a mettere il nostro codice  
});
```

FUNZIONE CLICK

LA FUNZIONE CLICK CI PERMETTE DI ASSOCIARE ALL'ELEMENTO A CUI VIENE CONCATENATA UNA FUNZIONE CHE DEVE ESSERE ESEGUITA QUANDO QUEL PARTICOLARE ELEMENTO VIENE CLICCATO.

ESEMPIO:

```
$(document).ready(function()  
{  
    $(".gallery img").click(function()  
    {  
        //Qui mettiamo il codice che deve essere eseguito quando  
        //una qualunque img in .gallery viene cliccata  
    });  
});
```

FUNZIONE ADDCLASS

LA FUNZIONE ADDCLASS CI PERMETTE DI ASSOCIARE ALL'ELEMENTO A CUI VIENE CONCATENATA UNA CLASSE.

ESEMPIO:

```
$(document).ready(function()  
{  
    $(".gallery img").click(function()  
    {  
        $(this).addClass('myclass');  
    });  
});
```

NOTA BENE: SE LA CLASSE È GIÀ ASSOCIATA, QUESTA NON VIENE AGGIUNTA.

FUNZIONE REMOVECLASS

LA FUNZIONE REMOVECLASS CI PERMETTE DI RIMUOVERE ALL'ELEMENTO A CUI VIENE CONCATENATA UNA CLASSE.

ESEMPIO:

```
$(document).ready(function()  
{  
    $(".gallery img").click(function()  
    {  
        $(this).removeClass('myclass');  
    });  
});
```

NOTA BENE: SE LA CLASSE NON È ASSOCIATA, NON SUCCEDE NULLA.

FUNZIONE TOGGLECLASS

LA FUNZIONE TOGGLECLASS CI PERMETTE DI AGGIUNGERE SE NON C'È UNA CLASSE O DI TOGLIERLA SE INVECE ERA GIÀ ASSOCIATA.

ESEMPIO:

```
$(document).ready(function()  
{  
    $(".gallery img").click(function()  
    {  
        $(this).toggleClass('myclass');  
    });  
});
```

FUNZIONI DI VISIBILITÀ

LE FUNZIONI SHOW, HIDE, FADEIN E FADEOUT CI PERMETTONO DI CAMBIARE LA VISIBILITÀ DI UN ELEMENTO. LA COSA INTERESSANTE È CHE QUESTE FUNZIONI POSSONO ESSERE APPLICATE A SEGUITO DEL CLIC SU QUALCHE ALTRO ELEMENTO.

SHOW = VISUALIZZA UN ELEMENTO CHE ERA NASCOSTO

HIDE = NASCONDE UN ELEMENTO CHE ERA VISIBILE

FADEIN = FA COMPARE UN ELEMENTO CHE ERA NASCOSTO, SFUMANDO IL SUO INGRESSO

FADEOUT = FA SCOMPARE UN ELEMENTO CHE ERA VISIBILE, SFUMANDO LA SUA USCITA.

FADETOGGLE = FA COMPARE / SCOMPARE UN ELEMENTO.

IN ALTRE PAROLE

SENZA L'USO DI JQUERY POTEVAMO CAMBIARE LO STILE ASSOCIATO AD UN ELEMENTO UTILIZZANDO SOLAMENTE LE PSEUDOCCLASSI :HOVER, :FOCUS E POCHE ALTRE.

CON JAVASCRIPT E JQUERY, INVECE, POSSIAMO SCRIVERE CODICE INTERATTIVO, ASSOCIANDO AD UN PARTICOLARE ELEMENTO, AD ESEMPIO, UNA CLASSE QUANDO QUESTO VIENE CLICCATO.

TUTTO CIÒ CHE ABBIAMO VISTO, RIMANE VALIDO! CIÒ SIGNIFICA CHE NON APPENA UN ELEMENTO VIENE CLICCATO, GLI VIENE ASSOCIATA LA CLASSE E IL MOTORE DI RENDERING DEL BROWSER ASSOCERÀ LO STILE (CON EVENTUALMENTE LE TRANSIZIONI IMPOSTATE!!!)

**PRONTI PER UN
PO' DI ESERCIZI?**

NOVIZIO

Installare jQuery e associare una classe (e uno stile) quando l'utente clicca sulle immagini dei prodotti / servizi.

INTERMEDIO

Come novizio + rimuovere una classe quando l'utente clicca sulle immagini dei prodotti / servizi.

NERD

Come intermedio + far aprire un "finto popup" al clic delle immagini con un testo descrittivo. Il finto popup altro non è che un `<div>` che occupa quasi tutto lo schermo, posizionato assolutamente e con lo sfondo semiopaco.