



HTML



Università degli Studi di Milano
Insegnamento di Applicazioni Web e Cloud

Linguaggio di markup

- ▶ Linguaggio di markup modellato per rendere esplicita una particolare interpretazione di un testo
 - ▶ Per esempio, tutte quelle aggiunte al testo scritto che permettono di renderlo più fruibile
 - ▶ Oltre a rendere il testo più leggibile, il markup permette anche di specificare ulteriori usi del testo
- ▶ Con il markup per sistemi informatici, specifichiamo le modalità esatte di utilizzo del testo nel sistema stesso
- ▶ I linguaggi di markup sono i linguaggi più opportuni per strutturare e marcare i documenti in maniera indipendente dall'applicazione, favorendo la riusabilità, la flessibilità e la apertura ad applicazioni complesse



HyperText Markup Language (HTML)

- ▶ HTML linguaggio per la creazione di pagine Web
 - ▶ Standard per la rappresentazione
 - ▶ Descrive la struttura della pagina
 - ▶ Non è un linguaggio di programmazione, è un linguaggio di markup
 - ▶ Descrive dati e regole su come mostrarli
 - ▶ Poche e semplici regole sintattiche
- ▶ File di testo



Caratteristiche

- ▶ Iper testo
- ▶ Multimedia
- ▶ Ipermedia



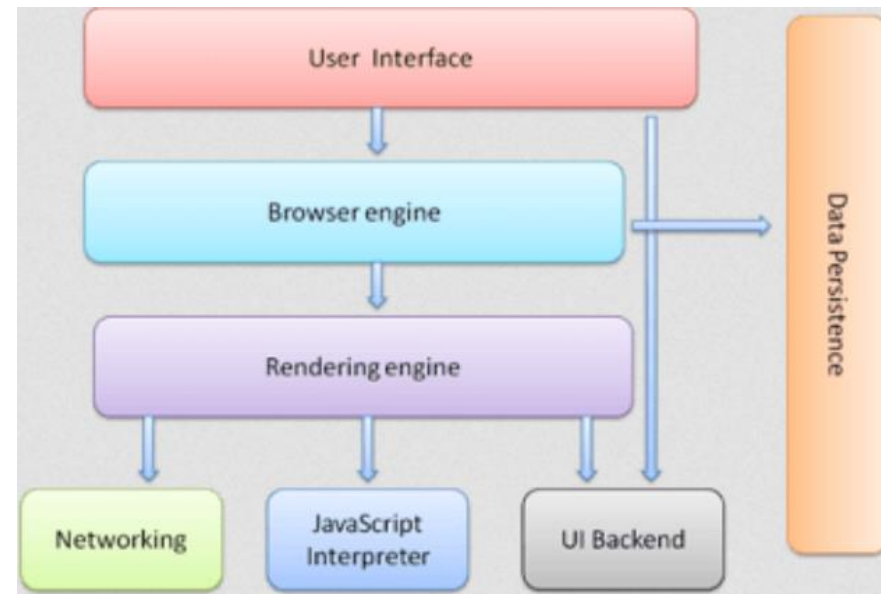
Browser - 1

- ▶ Programma per la navigazione nel Web
- ▶ Richiede risorse attraverso il web (all'indirizzo richiesto) e le mostra nella finestra
- ▶ Legge e interpreta i documenti HTML, CSS, JavaScript, immagini sulla base di specifiche e standard
- ▶ Browser: Internet Explorer, Mozilla Firefox, Chrome, Opera, Safari



Browser - 2

- ▶ **Layout Engine** – riceve input dal browser (URL bar, search box, mouse clicks and key presses) e li passa al rendering engine
- ▶ **Rendering Engine** – riceve il codice HTML e lo interpreta visualizzandolo a video. Ad esempio, testo in bold
- ▶ **User Interface** – controlli del browser, ad esempio, bottoni per andare avanti e indietro, bookmark
- ▶ **JavaScript Engine** – engine che si prende cura di parsare ed eseguire codice JavaScript per poi ritornare il risultato dell'esecuzione
- ▶ **Network Layer** – gestisce funzioni di rete, ad esempio, crittazione, richieste http e tutte le configurazioni di rete
- ▶ **Storage** – porzione di memoria dove il browser memorizza cached file, cookie e oggetti e dati create con JavaScript
- ▶ **Operating System Interface** – componente che interagisce con il sistema operativo per disegnare diversi elementi e per gestire la finestra (chiudi, massimizza, minimizza)



Browser - 3

- ▶ Mentalità multi-browser
- ▶ Obiettivo: rendere visibile la propria pagina su tutti i browser
- ▶ Rendering

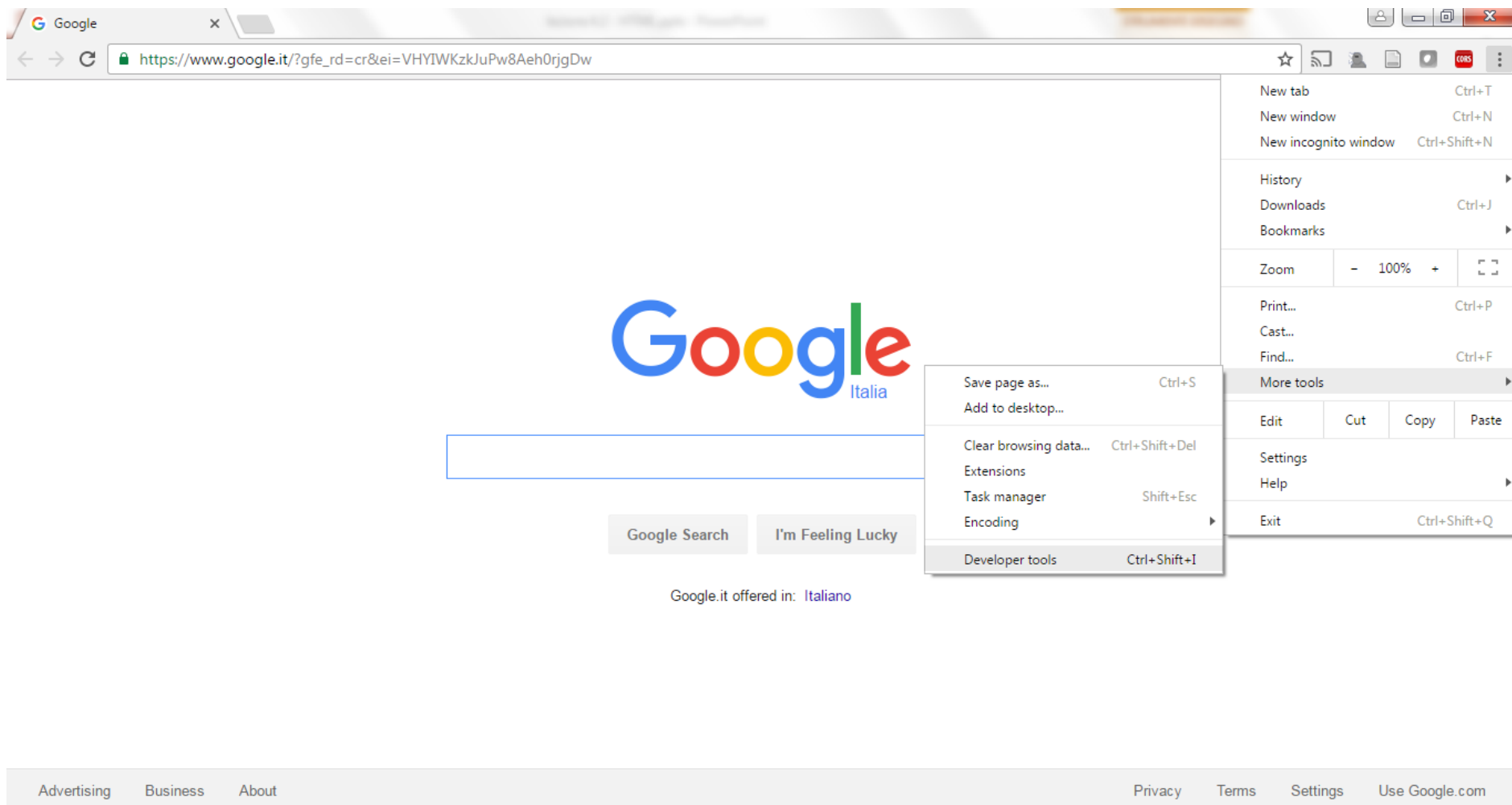


Browser - 4

- ▶ Contiene tool per gli sviluppatori
 - ▶ Debug pagina
 - ▶ Analisi stili e sorgenti dei file
 - ▶ Visualizzazione HTTP request e response
 - ▶ Accesso a cookie e storage
 - ▶ Molto altro...



Browser - 5



Browser - 6

The screenshot displays a web browser's developer tools interface. The top pane shows the HTML document structure, and the bottom pane shows the CSS styles applied to the selected element.

HTML Structure:

```
<!DOCTYPE html>
<html itemscope itemtype="http://schema.org/WebPage" lang="en-IT"> == $0
  <head>...</head>
  <body class="hp vasq" onload="try{if(!google.j.b)
    {document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}}
    catch(e){}if(document.images)new Image().src='/images/nav_logo242.png'" id=
    "gsr">
    <div class="ctr-p" id="viewport">
      <div data-jiis="cc" id="doc-info"></div>
      <div data-jiis="cc" id="cst">...</div>
      <a href="/setprefs?suggon=2&prev=https://www.google.it/?
        gfe_rd%3Dcr%26ei%3DVHYIWKzkJuPw8Aeh0rjgDw&sig=0_L9P-
        dmbNmbHQDybSheSkGYJRr18%3D" style="left:-1000em;position:
        absolute">Screen-reader users, click here to turn off Google Instant.</a>
      <textarea name="csi" id="csi" style="display:none"></textarea>
      <script>if(google.j.b)document.body.style.visibility='hidden';</script>
      <div class="jhp" id="searchform">
        <script>...</script>
        <style>...</style>
        <div id="oh" class="oh I"></div>
    </div>
```

Styles:

Filter: :hov .cls +

element.style { }

body, html {
font-size: small;
}

html, body {
height: 100%;
margin: 0;
}

html[Attributes Style] {
-webkit-locale: "en-IT";
}

html {
display: block;
}

user agent stylesheet

Diagram:

A diagram illustrating the box model for the selected element. It shows a central content area (791 x 645) surrounded by padding, a border, and a margin.

Properties:

Property	Value
display	block
font-size	13px
height	645px
margin-bottom	0px
margin-left	0px

HTML

- ▶ HTML non è un linguaggio di programmazione
- ▶ Linguaggio di markup (tag)
- ▶ Nessun meccanismo di decisione e iterazione
- ▶ Poche regole sintattiche



Da HTML a XHTML

▶ HTML

- ▶ W3C (WWW Consortium): HTML 2.0, HTML 3.2, HTML 4.01
- ▶ Per saper usare XHTML bisogna conoscere HTML
- ▶ Le differenze tra i due non sono così marcate
 - ▶ Le discutiamo alla fine di questa presentazione



Storia degli standard web

- ▶ Working Draft (WD)
 - ▶ Documento che il W3C pubblica per ottenere una revisione dalla comunità (include W3C Member, pubblico e altre organizzazioni tecniche)
- ▶ Candidate Recommendation (CR)
 - ▶ Documento revisionato in dettaglio che il W3C ritiene soddisfi i requisiti tecnici del Group
 - ▶ W3C pubblica una Candidate Recommendation per raccogliere esperienza implementativa
- ▶ Proposed Recommendation (PR)
 - ▶ Un technical report maturo che, dopo una dettagliata revisione che valuta l'implementabilità e la correttezza della soluzione, W3C invia al W3C Advisory Committee per un endorsement finale
- ▶ W3C Recommendation (REC)
 - ▶ Una specifica o insieme di guideline che, dopo aver raccolto un ampio consenso, ha ricevuto l'endorsement dei W3C Member e del W3C Director
 - ▶ W3C raccomanda il deployment e l'utilizzo della Recommendation
 - ▶ W3C Recommendation sono simili a standard pubblicati da altre organizzazioni



Storia degli standard web

91-92	93-94	95-96	97-98	99-00	01-02	03-04	05-06	07-08	09-10	11-12	13-14
HTML 1	HTML 2	HTML 3	HTML 4	XHTML 1					HTML 5		
		CSS 1	CSS 2			Web 2.0			CSS3		
		JS	XML 1.0, DOM	DOM 2		XML 1.1	Ajax		DOM, APIs		

HTML

1991	HTML 1 (HTML Tag)
1994	HTML 2 Draft
1995	HTML 3 Internet Draft
1997	HTML 3.2 W3C Rec
1997-98	HTML 4.0 W3C Proposed Rec
2008	HTML 5 W3C Working Draft
2012	HTML 5 W3C Candidate Rec
2014	HTML 5 W3C Rec



Storia degli standard web

- ▶ Ultima parte del 1991: prima descrizione di HTML (HTML tag)
- ▶ 1993-94: HTML 2.0 pubblicato
- ▶ Marzo 1995: HTML 3.0 pubblicato come internet draft
- ▶ Dicembre 1997: HTML 4.0 is published by the W3C
- ▶ Febbraio-Marzo 1998: XML 1.0 is published



Storia degli standard web

- ▶ Dicembre 1999-Gennaio 2000: ECMAScript 3rd Edition, XHTML 1.0 (HTML tag riformulati in XML)e HTML 4.01 recommendation pubblicate
- ▶ Maggio 2001: XHTML 1.1 recommendation pubblicata
- ▶ Agosto 2002: XHTML 2.0 first working draft rilasciato
- ▶ Dicembre 2002: XHTML 2.0 second working draft pubblicato
- ▶ Gennaio 2008: First W3C working draft di HTML5 pubblicato



Scrivere una pagina Web

- ▶ Creazione file .html o .htm
- ▶ Editor Testuale vs Editor Visuale
- ▶ Visualizzazione pagina tramite browser





Struttura della pagina HTML

HTML - Tag

- ▶ Tag sono marcatori che identificano porzioni di testo
- ▶ Permettono la personalizzazione della pagina, e di identificare e processare le porzioni di testo contrassegnate da una certa marcatura
- ▶ Nome tag = nome funzione



Struttura tag

- ▶ Tag contenuto tra parentesi triangolari
- ▶ Tag di inizio, tag di fine
- ▶ Contenuto del tag
- ▶ `<tag attributi>contenuto</tag>`



Esempio tag

- ▶ Tag con contenuto

- ▶ `testo`

- ▶ Tag senza contenuto

- ▶ ``



Struttura della pagina - 1

- ▶ Riga di intestazione

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//IT">
```

- ▶ Indichiamo le specifiche W3C adottate
- ▶ Dice al browser quale linguaggio è stato usato per il rendering



Struttura della pagina - 2

- ▶ Il tag `<html>` indica l'inizio di una pagina HTML
- ▶ Tutto ciò compreso all'interno del tag `<html>` è il codice HTML

`<html>`

... codice HTML ...

`</html>`



Struttura della pagina - 3

- ▶ Documento HTML diviso in due

- ▶ Testa

- ▶ Corpo



Testa

- ▶ Contiene informazioni non immediate
- ▶ Descrive come il documento deve essere letto e interpretato
- ▶ Contiene meta-tag, script, stili
- ▶ Racchiuso tra il tag <head>



Corpo

- ▶ Contiene il documento vero e proprio
- ▶ Contiene tutti i tag per la realizzazione del sito Web (ad esempio)
- ▶ Racchiuso tra il tag <body>



Esempio

- ▶ Scrivere una pagina HTML con i tag essenziali (html, head, body) inserendo una stringa all'interno del corpo

- ▶ primo.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">  
<html>  
<head>  
</head>  
<body>  
Hello World!  
</body>  
</html>
```



Indentazione

- ▶ Buona norma utilizzare caratteri di tabulazione
- ▶ Aumenta leggibilità
- ▶ Diminuisce tempi di modifica



Esempio - Indentazione

```
<html>
```

```
  <head>
```

```
  </head>
```

```
  <body>
```

```
    Hello World!
```

```
  </body>
```

```
</html>
```

```
<html><head></head>  <body>
```

```
  Hello World!</body></html>
```



Esempio - Indentazione

- ▶ Scrittura equivalente per il browser
- ▶ Non per l'uomo
 - ▶ Un'unica pagina HTML su una stessa riga risulterebbe illeggibile



Annidamento

- ▶ Caratteristica HTML: tag annidati

```
<tag1 attributi>
```

```
    contenuto1
```

```
        <tag2 attributi>
```

```
            contenuto2
```

```
        </tag2>
```

```
</tag1>
```

- ▶ Formattazioni successive



Annidamento

- ▶ Esempio

```
<font color="blue">
```

```
    Hello
```

```
        <font color="red">
```

```
            World!
```

```
        </font>
```

```
</font>
```

- ▶ Hello World!



Commenti

- ▶ Aumenta la leggibilità del codice
- ▶ Buona norma commentare SOLO le parti significative
- ▶ Permettono insieme all'indentazione di orientarsi all'interno di un grosso documento



Esempio - Commenti

- ▶ `<!--` = inizio commento
- ▶ `-->` = fine commento
- ▶ `<!-- questo è un commento -->`



Maiuscolo o minuscolo

- ▶ Case unsensitive

- ▶ `testo`

- ▶ `testo`

- ▶ XHTML case sensitive

- ▶ Consigliabile usare carattere minuscolo



Altre caratteristiche

- ▶ Non è sensibile agli spazi

```
<font color="blue">Hello    World    !    </font>
```

- ▶ Non è sensibile alle linee vuote

```
<font color="blue">Hello
```

```
World    !
```

```
</font>
```


- ▶ Stringa visualizzata: Hello World !



Esercizio 1

- ▶ Realizzare una pagina HTML che visualizza la scritta *Hello World!* sfruttando l'annidamento e utilizzando commenti e indentatura.
- ▶ Esempio1.html





Testa (<head>)

Testa

- ▶ Contiene informazioni/metadati relativi al documento
 - ▶ Ad esempio, titolo, keyword, altri dati non considerati contenuto del documento
 - ▶ Browser di solito non presentano elementi che compaiono nella testa (HEAD)
 - ▶ Rendono disponibili queste informazioni attraverso altri meccanismi



Testa: elemento title

- ▶ Elemento <title>
- ▶ Indica il titolo della pagina
- ▶ Viene visualizzato in alto a sinistra nei browser più datati o nella scheda nei browser più moderni



Metadati

- ▶ Elemento <meta>
 - ▶ Definisce informazioni relative alla pagina piuttosto che contenuto vero e proprio
- ▶ Contiene tre attributi principali
 - ▶ Name, content, http-equiv
 - ▶ Name contiene il nome della proprietà
 - ▶ Author, Description, Copyright, Generator, Language, Keywords
 - ▶ Content contiene il valore della proprietà
 - ▶ http-equiv specifica caratteristiche dell'HTTP response header



Esempi

<META NAME="author" CONTENT="Claudio">

<META NAME="description" CONTENT="...">

<META NAME="copyright" CONTENT="...">

<META NAME="keyword" CONTENT="...">



Meta Tag - Keywords

- ▶ Indica alcune informazioni sul contenuto del sito
- ▶ Separate da virgola, punto e virgola o spazio
- ▶ Sono chiavi di ricerca



Meta Tag - Keywords

- ▶ Motore di ricerca ricava chiavi di indicizzazione
- ▶ Attenzione nella scrittura delle keywords
 - ▶ Evitare termini generici e particolari
 - ▶ Riportare termine italiano e inglese
 - ▶ Singolare e plurale
- ▶ Evitare di ripetere le stesse parole e usare Keyword astute
- ▶ `<meta name="keywords" content="calcio soccer tennis basket pallacanestro pallavolo volley inter milan juventus roma lazio...">`



Attributo HTTP-EQUIV

- ▶ Informazioni sulla comunicazione server-browser
 - ▶ Può essere usato al posto dell'attributo name
- ▶ Attributo http-equiv fornisce un header HTTP per informazioni/valore dell'attributo content
 - ▶ Può essere usato per simulare un HTTP response header
 - ▶ HTTP server usano questo attributo per ottenere informazioni circa l'HTTP response header
- ▶ Alcuni valori
 - ▶ content-type, default-style, refresh
- ▶ Esempio
 - ▶ `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`



Esempio

▶ Esempio2.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
  Transitional//IT">
```

```
<html>
```

```
  <head>
```

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

```
    <title>Pagina di prova</title>
```

```
  </head>
```

```
  <body>
```

```
    <!-- Scriveremo qui -->
```

```
    Hello World!
```

```
  </body>
```

```
</html>
```





Corpo (<body>)

Corpo

- ▶ Il corpo di un documento contiene il contenuto vero e proprio
- ▶ Il browser presenta il contenuto in diversi modi
 - ▶ Browser visuali: è una tela su cui disegnare testi immagini colori...
 - ▶ Agenti audio: lo stesso contenuto è parlato
- ▶ Contiene tutte le informazioni per la visualizzazione della pagina
 - ▶ Siccome gli style sheet sono il modo suggerito per specificare come verrà presentato un documento gli attributi del BODY sono stati deprecati





Tag interni al body

Elementi inline vs block

- ▶ Alcuni elementi HTML nel body sono "block-level" altri "inline" ("text level")
- ▶ Content model
 - ▶ Elementi block-level possono contenere elementi inline e altri elementi block-level
 - ▶ Elementi inline contengono solo dati e altri elementi inline
- ▶ Formatting
 - ▶ Elementi block-level sono formattati in maniera differente dagli elementi inline
 - ▶ Elementi block-level iniziano su una linea nuova, elementi inline no



Testo – Tag predefiniti

- ▶ Tag-contenitori di testo (elementi block-level, unica eccezione ``)
 - ▶ `<h1>`, `<h2>` ... (h = heading)
 - ▶ `<p>`
 - ▶ ``
 - ▶ `<div>`



Titoli - Heading

- ▶ Sono previste sei grandezze predefinite per i titoli
- ▶ Si va da <h1> che è la grandezza maggiore fino ad <h6> che è la grandezza minore
- ▶ Sintassi: <h1>Titolo</h1>



Esempio

- ▶ `<h1>Titolo1</h1>`
- ▶ `<h2>Titolo2</h2>`
- ▶ `<h3>Titolo3</h3>`
- ▶ `<h4>Titolo4</h4>`
- ▶ `<h5>Titolo5</h5>`
- ▶ `<h6>Titolo6</h6>`

- ▶ Esempio6.html



Paragrafo

- ▶ `<p>`
- ▶ Unità base di suddivisione del testo
- ▶ Sintassi: `<p>paragrafo</p>`
- ▶ `<p>` lascia una riga vuota prima e dopo il testo



Blocco testo

- ▶ `<div>` è un blocco contenitore
- ▶ Blocco testo va a capo ma non lascia righe di spazio
- ▶ Sintassi: `<div>blocco 1</div>`
- ▶ Non impone rappresentazioni di default (a parte il fatto di essere un elemento block-level)



Contenitore

- ▶ ``
- ▶ Contenitore generico
- ▶ Può essere annidato
- ▶ Elemento inline (continua sulla stessa riga del tag che lo contiene (ad es. `<div>`))
 - ▶ Non va a capo
- ▶ Non impone rappresentazioni di default (a parte il fatto di essere un elemento inline)



Esempio

- ▶ Pagina contenente un titolo, un paragrafo, un blocco di testo e un contenitore

...

```
<h2>Titolo della pagina</h2>
```

```
<p>paragrafo 1</p>
```

```
<p>paragrafo 2</p>
```

```
<div>blocco 1</div>
```

```
<div>blocco 2
```

```
  <span>contenitore 1</span>
```

```
  <span>contenitore 2</span>
```

```
</div>
```

- ▶ Esempio7.html



Struttura della pagina con tag <div>

- ▶ È uno tra i tag più utilizzati nella creazione di pagine Web, soprattutto nella creazione di layout
- ▶ Fornisce un vero e proprio elemento strutturale della pagina
- ▶ Suddivide gli spazi in zone per progettare il sito in modo semplice e dettagliato
- ▶ Molto utile quando usato insieme a fogli di stile



Struttura della pagina con tag <div>

- ▶ Facciamo subito un esempio: immaginando di dover costruire un sito pensiamo a come esso debba essere strutturato, tipicamente abbiamo
 - ▶ Un contenitore (container)
 - ▶ Una parte alta (header)
 - ▶ Un corpo centrale (middle)
 - ▶ Un menù (navigation)
 - ▶ Una parte bassa (footer)
 - ▶ Spesso anche un pannello laterale (sidebar)



Struttura della pagina con tag <div>

```
<div id="container">
```

```
  <div id="header">
```

```
    <div id="navigation"></div><!--#navigation-->
```

```
  </div><!--#header-->
```

```
  <div id="main"></div><!--#main-->
```

```
  <div id="sidebar"></div><!--#sidebar-->
```

```
  <div id="footer"></div><!--#footer-->
```

```
</div><!--#container-->
```



Struttura della pagina con tag <div>



Testo - Stili

- ▶ Due tipi di stili
 - ▶ Fisici: definiscono lo stile grafico
 - ▶ Logici: forniscono informazioni sul ruolo svolto dal testo
 - ▶ Opzionale: impongono uno stile grafico



Stili fisici - 1

- ▶ `` bold: formatta testo in **grassetto**
- ▶ `<i>` italic: formatta il testo in *corsivo*
- ▶ `<u>` underline: sottolinea il testo
- ▶ `<strike>`: testo barrato (usato per correzioni)
- ▶ Tutti elementi inline



Stili fisici - 2

- ▶ `<sup>` apice: $E=mc^2$
- ▶ `<sub>` pedice: H_2O
- ▶ Utili per la scrittura di formule matematiche



Stili logici - esempi

- ▶ `<abbr>` abbreviazione: non comporta nessun cambiamento grafico
- ▶ `<address>` indirizzo, `<cite>` citazioni: testo in corsivo
- ▶ `<samp>` esempio: testo a spaziatura fissa
- ▶ Esempio8.1.html



Testo - Font

- ▶ Font = colore, dimensione e tipo di carattere
- ▶ Carattere predefinito Times new roman
 - ▶ Poco leggibile
 - ▶ Meglio Verdana, Arial o Helvetica



Esempio

- ▶ `testo`
- ▶ E' buona norma:
 - ▶ Utilizzare caratteri sicuri (sicuramente visualizzabili)
 - ▶ Non indicare un solo carattere
 - ▶ Solo come ultima spiaggia sarà usato il Times



Testo - Font

- ▶ Scegliere il colore del testo
- ▶ `testo blu`
- ▶ La scelta del colore e del carattere può essere fatta nello stesso tag
- ▶ `testo blu`



Font annidati

```
<font color="blue" face="verdana">
```

```
  testo blu
```

```
    <font color="red" >
```

```
      testo rosso
```

```
    </font>
```

```
</font>
```



Dimensione del testo

- ▶ Attributo size del tag font
- ▶ Due modi per impostare la dimensione
 - ▶ Valore tra 1 e 7
 - ▶ `Dimensione 3`
 - ▶ Valori relativi alla dimensione base del font
 - ▶ `Dimensione +2`
 - ▶ Dimensione del font di base
 - ▶ Tag `<basefont>` permette di cambiare il font di base
- ▶ Esempio8.2.html



Elenchi

- ▶ Elenchi ordinati
- ▶ Elenchi non ordinati
- ▶ Elenchi di definizioni



Sintassi

<elenco>

<elemento> primo elemento

<elemento> secondo elemento

</elenco>

- ▶ Chiusura elemento opzionale
- ▶ Sintassi degli elenchi di definizione leggermente diversa



Elenchi ordinati

- ▶ Numerazione degli elementi
- ▶ Numerazione progressiva
- ▶ Tag elenco ordinato `` (ordered list)
- ▶ Tag elemento `` (list item)



Esempio elenchi ordinati

```
<div>Titolo</div>
```

```
<ol>
```

```
  <li>Primo elem prima tabella
```

```
  <li>Secondo elem prima tabella
```

```
  <ol>
```

```
    <li> Primo elem seconda tabella
```

```
    <li> Secondo elem seconda tabella
```

```
  </ol>
```

```
</ol>
```

► Esempio9.html



Elenchi non ordinati

- ▶ Nessuna numerazione
- ▶ Elenchi puntati
- ▶ Tabella tag `` (unordered list)



Esempio elenchi non ordinati

```
<ul>
```

```
  <li> Primo elemento
```

```
  <li> Secondo elemento
```

```
  <ul>
```

```
    <li> Primo elemento
```

```
    <li> Secondo elemento
```

```
  </ul>
```

```
</ul>
```

▶ Esempio11.html



Elenchi di definizioni

- ▶ Elenco di definizione <dl> (definition list)
- ▶ Termine da definire <dt> (definition term)
- ▶ Definizione del termine <dd> (definition description)



Elenchi di definizioni - Esempio

`<p>` Titolo

`<dl>`

`<dt><p>`

`<dd>`apertura paragrafo

`<dt><div>`

`<dd>`apertura blocco di testo

`<dt>`

`<dd>`apertura elemento inline

Altri tag...

`<dl>`

▶ Esempio13.html





Iperestualità (Link)

Introduzione

- ▶ Testi vs ipertesti
- ▶ Ipertesti caratteristica che ha reso grande il web
- ▶ Link = ponte tra un testo e un altro



Link

- ▶ I link sono formati da due componenti
 - ▶ Il contenuto (testo o immagine) che nasconde il collegamento
 - ▶ La risorsa puntata



Link

▶ Sintassi

- ▶ Clicca `` qui `` per visualizzare il collegamento.
- ▶ La testa del link è *qui*
- ▶ La coda è *indirizzo*

▶ Coda

- ▶ Pagina HTML, immagine, documenti, altri file



Destinazioni

- ▶ Pagina HTML: aperta nel browser
- ▶ Immagine (.gif, .jpeg): aperta nel browser
- ▶ Altri documenti (.doc, .pdf):
 - ▶ Visualizzata nel browser
 - ▶ Richiesta di salvataggio
- ▶ Altri file (.zip, .exe) richiesta di salvataggio



Esempio

- ▶ Link a pagina HTML
 - ▶ `Clicca qui`
- ▶ Link a documenti
 - ▶ `Clicca qui`
- ▶ Link ad immagini
 - ▶ `Clicca qui`
- ▶ Link ad altri file
 - ▶ `Clicca qui`



Un tipo particolare di link

- ▶ Link ad un indirizzo email
- ▶ Sintassi
 - ▶ ` Mandami un'email`
- ▶ Esempio15.html



Caratteristiche link

- ▶ Diversi stati
 - ▶ Link a riposo
 - ▶ Colore blu #0000FF
 - ▶ Link visitato
 - ▶ Colore violetto
 - ▶ Link attivo
 - ▶ Passaggio da una pagina all'altra
 - ▶ Utile nel passato
 - ▶ Link al passaggio del mouse
 - ▶ Solo con fogli di stile



Manipolazione dei link

- ▶ Attributi link, alink, vlink del tag body
- ▶ `<body link="red">` cambia il colore dei link
- ▶ `<body vlink="green">` cambia il colore dei link visited
- ▶ `<body alink="yellow">` cambia il colore dei link attivi
- ▶ Esempio link classici: Esempio15.html
- ▶ Esempio link personalizzati: Esempio16.html



Percorsi dei link

- ▶ Percorsi assoluti
 - ▶ Viene indicato il percorso per esteso
- ▶ Percorsi relativi
 - ▶ Fanno riferimento alla posizione del file a cui si riferisce al link a partire dalla posizione della pagina che stiamo sviluppando



Percorsi assoluti

- ▶ Leggi l'esempio ` qui `
- ▶ Nel caso di file sull'hard disk
- ▶ Leggi l'esempio ` qui `



Percorsi relativi

- ▶ `Clicca qui`
 - ▶ Indica al browser di cercare il file nella stessa directory
- ▶ `Clicca qui`
 - ▶ Indica al browser di cercare il file nella sottodirectory sottodir
- ▶ `Clicca qui`
 - ▶ .. Indica al browser di cercare il file nella directory padre
- ▶ Esempio18.html



Considerazioni

- ▶ Nomi dei file
 - ▶ Evitare gli spazi (mio_file.html)
 - ▶ Ricordarsi che maiuscole e minuscole fanno la differenza
- ▶ `` non funziona quando caricate la pagina sul vostro sito
- ▶ Allora diventerà
 - ▶ ``



Link interni o ancore

- ▶ Permette di creare un indice
- ▶ Ancora
 - ▶ ` àncora `
- ▶ Riferimento all'àncora
 - ▶ ` vai all'àncora `
- ▶ Riferimento ad inizio pagina
 - ▶ ` torna su `
- ▶ `ancora.html`



Colorare i link

- ▶ Colorare solo un link
- ▶ Annidare il font all'interno del tag del link
- ▶ ``
 ``
 link verde
 ``
 ``
- ▶ AttributiLink.html



Immagini

- ▶ Il web non è solo ipertesto ma bensì ipermedia
- ▶ Sintassi: ``
 - ▶ Attributo src: sorgente del file (percorso di memorizzazione file)
 - ▶ Attributo alt: descrizione mostrata al passaggio sull'immagine
 - ▶ Attributi width e height: dimensioni immagine
- ▶ Immagine come link
 - ▶ ``
 ``
 ``





Tabelle (table)

Tabelle

- ▶ Definizione tramite l'elemento `<table>`
- ▶ Ogni riga definita con il tag `<tr>` contenuto in `table`
- ▶ Ogni cella definita con il tag `<td>` contenuto in `<tr>`
- ▶ Una riga può essere anche divisa in table heading usando l'elemento `<th>`
 - ▶ Di solito mostrati centrati e in grassetto
- ▶ Elemento `<caption>` per definire la caption della tabella



Esempio

```
<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

▶ Esempiot1.html



Bordi

- ▶ Attributo border
 - ▶ Se non specificato nessun bordo
 - ▶ `<table border="1" style="width:100%">`
- ▶ Esempiot2.html



Celle collassate in una riga/colonna

- ▶ Attributo rowspan

- ▶ `<th rowspan="2">Telephone</th>`
- ▶ `<td rowspan="2">Telephone</td>`

- ▶ Attributo colspan

- ▶ `<th colspan="2">Telephone</th>`
- ▶ `<td colspan="2">Telephone</td>`

- ▶ Esempiot3.html





Form

Introduzione

- ▶ Form usate per collezionare input utente
- ▶ Basato sull'elemento `<form>`
- ▶ Specifica diversi elementi
 - ▶ input, checkbox, radio button, submit button...



Elemento input

- ▶ Elemento più importante
- ▶ Diverse varianti che dipendono dall'attributo type
 - ▶ Text: definisce input testo normale
 - ▶ Radio: definisce un input radio button (per selezionare una tra diverse scelte)
 - ▶ Submit: definisce un bottone submit (per sottomettere la form)



Text input

- ▶ `<input type="text">`
 - ▶ Definisce un campo testo di una linea
 - ▶ Esempiof1.html
 - ▶ `<input type="password">` un tipo speciale di campo testo dove il valore inserito viene nascosto

`<form>`

First name:`
`

`<input type="text" name="firstname">`

`
`

Last name:`
`

`<input type="text" name="lastname">`

`</form>`



Text area

- ▶ `<textarea>`

- ▶ Definisce un campo testo a linee multiple
- ▶ Esempiof1.html

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```



Radio button input

- ▶ `<input type="radio">`
 - ▶ Definisce un radio button che permette **una** scelta tra diverse opzioni
 - ▶ Esempiof2.html

```
<form>
```

```
  <input type="radio" name="sex" value="male"  
checked>Male
```

```
  <br>
```

```
  <input type="radio" name="sex" value="female">Female  
</form>
```



Checkbox input

- ▶ `<input type="checkbox">`
 - ▶ Definisce una serie di checkbox che permettono una scelta **multipla** tra diverse opzioni
 - ▶ Esempiof2.html

`<form>`

`<input type="checkbox" name="vehicle1" value="Bike"> I
have a bike`

`
`

`<input type="checkbox" name="vehicle2" value="Car"> I
have a car`

`</form>`



Drop-down list

- ▶ `<select>`
 - ▶ Definisce un menù a tendina
 - ▶ Elemento `<option>` definisce le possibili opzioni
 - ▶ `<option value="fiat" selected>Fiat</option>`
 - ▶ Esempiof3.html

```
<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```



Submit button

- ▶ `<input type="submit">`
 - ▶ Definisce un bottone per sottomettere la form ad un handler
 - ▶ Handler è una pagina lato server che processa i dati della form
 - ▶ Handler è definito con l'attributo `action`
 - ▶ Esempiof4.html

```
<form action="action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey">  
  <br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse">  
  <br><br>  
  <input type="submit" value="Submit">  
</form>
```

- ▶ Permette anche di eseguire semplici azioni
 - ▶ `<button type="button" onclick="alert('Hello World!')">Click Me!</button>`



Action attribute

- ▶ Definisce le azioni che devono essere fatte quando la form è inviata
- ▶ Sottomissione fatta solitamente con il bottone submit
- ▶ La pagina viene inviata a una pagina web lato server
 - ▶ Ad esempio uno script lato server gestisce la form inviata
 - ▶ `<form action="action_page.php">`
 - ▶ Se l'attributo action è omissso si considera la pagina corrente di default



Method attribute

- ▶ Specifica l'operazione HTTP (GET o POST) da usare per inviare la form
 - ▶ `<form action="action_page.php" method="get">`
 - ▶ `<form action="action_page.php" method="post">`
- ▶ GET (default) usata per invio di form passive (ad es., search engine query) e senza informazioni sensibili
- ▶ I dati inviati tramite la form sono visibili nell'indirizzo
 - ▶ `action_page.php?firstname=Mickey&lastname=Mouse`
- ▶ GET è ottima per piccola quantità di dati



Method attribute

- ▶ Specifica l'operazione HTTP (GET o POST) da usare per inviare la form
 - ▶ `<form action="action_page.php" method="get">`
 - ▶ `<form action="action_page.php" method="post">`
- ▶ GET (default) usata per invio di form che aggiornano dati
- ▶ POST fornisce maggiore sicurezza perchè i dati non sono visibili nell'indirizzo (ad es., password)



Name attribute

- ▶ Ogni campo input deve avere un nome

```
<form action="action_page.php">
```

```
  First name:<br>
```

```
  <input type="text" name="firstname" value="Mickey">
```

```
  <br>
```

```
  Last name:<br>
```

```
  <input type="text" name="lastname" value="Mouse">
```

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

```
  <input type="submit" value="Submit">
```

```
</form>
```



Fieldset

- ▶ Raggruppa campi in una form tramite elemento <fieldset>
- ▶ Si può definire una caption per la form <legend>
- ▶

```
<form action="action_page.php">  
  <fieldset>  
    <legend>Personal information:</legend>  
    First name:<br>  
    <input type="text" name="firstname" value="Mickey">  
    <br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse">  
    <br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>
```
- ▶ Esempio5.html



Altri attributi

- ▶ Attributo value specifica il valore iniziale
 - ▶ `<input type="text" name="firstname" value="John">`
- ▶ Attributo readonly specifica che il valore non può essere cambiato
 - ▶ `<input type="text" name="firstname" value="John" readonly>`
- ▶ Attributo disabled specifica un campo disabilitato e non utilizzabile (non viene inviato)
 - ▶ `<input type="text" name="firstname" value="John" disabled>`



Altri attributi

- ▶ Attributo size specifica la dimensione in caratteri del campo
 - ▶ `<input type="text" name="firstname" value="John" size="40">`
- ▶ Attributo maxlength specifica la dimensione massima del contenuto del campo
 - ▶ `<input type="text" name="firstname" maxlength="10">`





XHTML

XHTML

- ▶ XHTML = **EX**tensible **H**yper**T**ext **M**arkup **L**anguage
- ▶ XHTML è quasi identico a HTML
- ▶ XHTML è più stringente di HTML
- ▶ XHTML è supportato dai principali browser



Perché XHTML?

- ▶ Molte pagine web contengono HTML errato
- ▶ Il seguente codice viene visualizzato bene nella maggior parte dei casi anche se non segue le regole HTML

```
<html>
```

```
<head>
```

```
  <title>This is bad HTML</title>
```

```
<body>
```

```
  <h1>Bad HTML
```

```
  <p>This is a paragraph
```

```
</body>
```



Perché XHTML?

- ▶ Diversi browser che eseguono su diversi device
 - ▶ Interpretare un markup sbagliato richiede maggiori risorse o potenza computazionale (più complicato per i device più piccoli)
- ▶ XML è un linguaggio di markup che esploreremo più avanti in questo insegnamento dove i documenti devono essere ben formati (markup corretto)
- ▶ XHTML combina HTML e XML
- ▶ XHTML è HTML ridisegnato come XML
- ▶ XHTML definito per essere visualizzabile dai browser per HTML



HTML vs X-HTML

- ▶ Struttura documento
 - ▶ XHTML DOCTYPE è obbligatorio
 - ▶ L'attributo xmlns in <html> è obbligatorio
 - ▶ <html>, <head>, <title>, e <body> sono obbligatori
- ▶ Elementi XHTML
 - ▶ Elementi XHTML devono essere innestati correttamente
 - ▶ Elementi XHTML devono sempre essere chiusi
 - ▶ Elementi XHTML devono essere minuscoli
 - ▶ Elementi XHTML devono avere un'unico elemento radice



HTML vs X-HTML

- ▶ Attributi XHTML
 - ▶ Nomi di attributi devono essere minuscoli
 - ▶ Valori di attributi devono essere quoted
 - ▶ Minimizzazione degli attributi vietata



Come convertire HTML in X-HTML

- ▶ Aggiungere un XHTML `<!DOCTYPE>` nella prima linea di ogni pagina
- ▶ Aggiungere un attributo `xmlns` all'elemento `html` di ogni pagina
- ▶ Cambiare tutti i nomi di elemento in minuscolo
- ▶ Chiudere tutti gli elementi vuoti
- ▶ Cambiare tutti i nomi di attributo in minuscolo
- ▶ Quotare tutti i valori di attributi



Validazione

- ▶ W3C segue l'evoluzione del Web
- ▶ W3C fornisce linee guida e regole per la specifica delle pagine
- ▶ W3C fornisce un meccanismo per la validazione delle pagine web
 - ▶ <http://validator.w3.org>



Conclusioni

- ▶ Linguaggio per la rappresentazione delle pagine web
- ▶ HTML
 - ▶ Block-level vs inline
 - ▶ Link
 - ▶ Tabelle
 - ▶ Form
- ▶ XHTML

