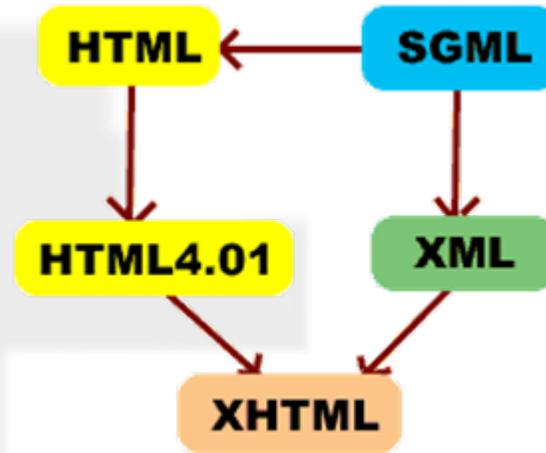


# XHTML



*Definizioni, DTD e semantica dei tag  
partendo dal “vecchio” HTML*

# HTML: i marcatori

- **Libro “cartaceo”**
  - Contenuto suddiviso in titoli, sottotitoli, paragrafi...
- **Libro “virtuale” sul Web**
  - Se desideriamo mantenerne intatta la struttura, dovremmo utilizzare qualche linguaggio universalmente comprensibile ai computer

# HTML: i marcatori

- Il linguaggio HTML alterna al testo vero e proprio alcuni marcatori descrittivi (**TAG**) che contengono informazioni **strutturali, semantiche e presentazionali**

# HTML: i marcatori

- **HTML è, quindi, un linguaggio di "marcatura":**
  - **Una grammatica che può essere utilizzata per strutturare informazioni**

# HTML: i marcatori

- Il “mattoncino” di un linguaggio di marcatura come HTML è l'elemento o TAG

# HTML: gli elementi

Un elemento è composto da

- **Nome**
  - ne identifica il tipo
- **Attributi**
  - valori associati agli elementi
- **Contenuto**
  - Per esempio testo

# HTML: gli elementi

- Ecco la sintassi di un elemento:

```
<nome_elemento></nome_elemento>
```

```
<nome_elemento>Contenuto</nome_elemento>
```

```
<nome_elemento nome_attributo="valore">  
Contenuto</nome_elemento>
```

# HTML: gli elementi

- Nel linguaggio HTML, per esempio, un paragrafo apparirà codificato in questa forma:

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

# XHTML – Perché?

- XML richiede che ogni suo elemento sia marcato correttamente, il che genera i documenti “well-formed”

# XHTML – Perché?

- **XHTML è una combinazione di HTML ed XML**
- **XHTML consiste di tutti gli elementi dell'HTML 4.01 combinati con la sintassi XML**

# XHTML – Perché?

- **Alcuni User Agent per cellulari o palmari non sono in grado di interpretare adeguatamente un linguaggio di markup scorretto**
- **Una pagina XHTML verrà letta correttamente da ogni device in grado di leggere XML**

# XHTML – Perché?

- **L'XHTML offre l'opportunità di costruire ADESSO documenti well-formed in grado di funzionare in tutti i browser ed anche ragionevolmente compatibili all'indietro**

# XHTML – Perché?

- Oggi troppe pagine sul Web contengono codice di pessima qualità
  - Il seguente codice è in grado di essere visualizzato pur non seguendo le regole dell'HTML:

```
<html>  
<head>  
<title>This is bad HTML</title>  
<body>  
<h1>Bad HTML  
</body>
```

# Prepararsi all'XHTML

- **XHTML non è poi così diverso dall'HTML 4.01**
- **Già trasformare il tuo codice in HTML 4.01 può essere un buon inizio**

# HTML vs XHTML

- **Le differenze principali:**
  - **Gli elementi XHTML devono essere opportunamente innestati**
  - **I documenti XHTML devono essere well-formed**
  - **I nomi dei tag devono essere scritti in minuscolo**
  - **Tutti gli elementi XHTML devono essere chiusi**

# Elementi ben innestati

- L'HTML permette che alcuni elementi ne contengano altri senza un criterio definito:

```
<b><i>Questo testo è grassetto e corsivo</b></i>
```

CODICE ERRATO

- In XHTML tutti gli elementi devono essere innestati correttamente:

```
<b><i>Questo testo è grassetto e corsivo</i></b>
```

CODICE CORRETTO

# Elementi ben innestati

```
<ul>
<li>Caffè</li>
<li>Tea
  <ul>
    <li>Black tea</li>
    <li>Green tea</li>
  </ul>
<li>Latte</li>
</ul>
```

CODICE ERRATO

```
<ul>
<li>Caffè</li>
<li>Tea
  <ul>
    <li>Black tea</li>
    <li>Green tea</li>
  </ul>
</li>
  <li>Latte</li>
</ul>
```

CODICE CORRETTO

# Documenti Well-Formed

- Tutti gli elementi XHTML devono essere inseriti all'interno del nodo radice `<html>`
- Tutti gli altri elementi possono avere elementi figli (*children*)
- Gli elementi figli devono essere correttamente innestati all'interno dell'elemento genitore

# Tag scritti in minuscolo

- Questo perché i documenti XHTML sono applicazioni XML, ed XML è **case-sensitive**
  - Tag come `<br>` e `<BR>` sono considerati come elementi differenti.

```
<BODY> <P>Questo è un paragrafo</p> </BODY>
```

CODICE ERRATO

```
<body> <p>Questo è un paragrafo</p> </body>
```

CODICE CORRETTO

# Chiudi tutti i Tag

- Tutti gli elementi XHTML devono essere chiusi
- Gli elementi non vuoti devono avere un tag di chiusura

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

CODICE ERRATO

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

CODICE CORRETTO

# Chiudi tutti i Tag

- Anche gli elementi vuoti devono essere chiusi
- Gli elementi vuoti devono avere un tag di chiusura oppure il tag di apertura deve terminare con `</>`

```
This is a break<br> Ecco una riga orizzontale:  
<hr> Ecco un'immagine 
```

CODICE ERRATO

```
This is a break<br /> Ecco una riga orizzontale:  
<hr /> Ecco un'immagine 
```

CODICE CORRETTO

## Nota di compatibilità:

- Affinchè il tuo XHTML sia compatibile con i browser odierni, bisogna aggiungere uno spazio aggiuntivo prima del carattere “/”:

```
<br />
```

```
<hr />
```

# Altre regole di XHTML

- I nomi degli attributi devono essere scritti in **minuscolo**
- I valori degli attributi devono essere chiusi fra i **doppi apici**
  - E' vietata la **minimizzazione** degli attributi
- L'attributo **id** rimpiazza l'attributo **name**

# L'attributo Lang

- E' applicabile a quasi tutti gli elementi XHTML
- Specifica la lingua del contenuto di quell'elemento
- Se vuoi usarlo, devi anche aggiungere l'attributo `xml:lang`...

```
<div lang="no" xml:lang="no">Heia  
Norge!</div>
```

# Elementi XHTML obbligatori

- **La dichiarazione del DOCTYPE è obbligatoria**
- **Gli elementi html, head, body e title devono essere presenti**

# Modello di base per una pagina XHTML

```
<!DOCTYPE qui ci va il doctype>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Qui ci va il title</title>
</head>
<body> Testo del corpo </body>
</html>
```

# II DocType

- La dichiarazione di DOCTYPE non è parte del documento XHTML, non è un elemento XHTML e non necessita di tag di chiusura.
- L'attributo xmlns nel tag <html> è obbligatorio in XHTML.
  - Comunque il validatore del w3c non “protesta” quando esso non è presente, dato che "xmlns=http://www.w3.org/1999/xhtml" è un valore fisso e verrà aggiunto al tag <html> anche se non lo si include

# II DocType

- **Specifica la sintassi della pagina in SGML**
  - **Regole da applicare al markup dei documenti**
  - **Insieme di elementi**

# II DocType

- **XHTML 1.0** specifica tre tipi di documenti XML che corrispondono a tre DTD:

**Transitional**

**Strict**

**FrameSet**

- Una DTD XHTML descrive in maniera formale la sintassi consentita e la grammatica del markup XHTML

# XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xht  
ml1-strict.dtd">
```

- Usalo per ottenere markup pulito ed efficiente, privo di “orpelli” di formattazione
- Usalo con i CSS
  - Include solo gli elementi che NON sono mai stati deprecati e che non appaiono in quello **FrameSet**

# XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xht  
ml1-transitional.dtd">
```

- Usalo come markup non necessariamente legato all'uso intensivo dei fogli di stile
  - E' sconsigliato se lo scopo è ottenere pagine accessibili e, più in generale, di elevata "qualità"

# XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xht  
ml1-frameset.dtd">
```

- Usalo (se proprio devi) per supportare l'adozione dei frames

# II DocType

- **Attenzione:**
  - **I browser più recenti (IE6) trattano lo stesso documento in maniera diversa a seconda del DOCTYPE**

# Validazione e DTD

- **La validazione di un documento XHTML si basa sulla corrispondente DTD**
- **Prima della validazione, la DTD corretta deve essere aggiunta nel primo rigo del documento**

# DTD Transitional???

## La DTD transitoria va utilizzata...

- **Quando desideriamo la compatibilità con i browser grafici più datati, forniti di scarso supporto per i fogli di stile**
  - In questo modo le nostre pagine web saranno viste più o meno allo stesso modo su Internet Explorer 3 - 6 e su Netscape 3 - 7
  - Così si mescolano ai contenuti della pagina tanti tag aventi il solo scopo di definire la presentazione visuale dei contenuti

# DTD Transitional???

- **NOTA:**
  - *Gli elementi deprecati fanno tutti parte della DTD transitional*
  - Utilizzarli **preclude** quindi, in base al punto di controllo **11.2** delle **WCAG 1.0**, il raggiungimento della conformità **Level-AA**

# DTD Strict!!!

**La DTD strict assicura:**

- **Maggiore indipendenza dei contenuti della pagina dai suoi aspetti di presentazione**
- **Accesso ai contenuti per il maggior numero possibile di programmi utente**

# DTD Strict!!!

**L'uso della DTD rigorosa è il miglior punto di partenza per il raggiungimento di una elevata accessibilità.**

# XHTML 1.1 & DTD

- **Le specifiche XHTML 1.1 eliminano le tre DTD presenti in HTML 4 e XHTML 1.0**

**In sostanza, l'unica DTD di XHTML 1.1  
è l'evoluzione della  
DTD strict di XHTML 1.0**

# DocType: i template

- I modelli per la corretta impostazione di documenti XHTML sono disponibili (anche) sulle pagine del Web Standard Project
  - *<http://www.webstandards.org/learn/templates/>*

# Contenuto, struttura e presentazione

- **Contenuto**
  - Si tratta delle informazioni che il documento comunica all'utente attraverso il testo, le immagini, il suono, i filmati, le animazioni ecc.
- **Struttura**
- **Presentazione**

# Contenuto, struttura e presentazione

- **Contenuto**
- **Struttura**
  - La struttura di un documento è la sua organizzazione logica. Le informazioni che contiene possono essere strutturate in introduzione, capitoli...
- **Presentazione**

# Contenuto, struttura e presentazione

- **Contenuto**
- **Struttura**
- **Presentazione**
  - E' il modo in cui un documento è reso o appare all'utente.
  - Il termine si riferisce alla modalità di fruizione dei contenuti: presentati su schermo oppure stampati, letti da un sintetizzatore vocale...

# La struttura di un documento

- **E' alla base della nascita di un'applicazione di qualità**
- **Deve essere studiata e pensata in maniera attenta**

# La struttura di un documento

**Comunicare la struttura logica  
soltanto per mezzo della  
presentazione visuale non è una  
soluzione corretta**

# La struttura di un documento

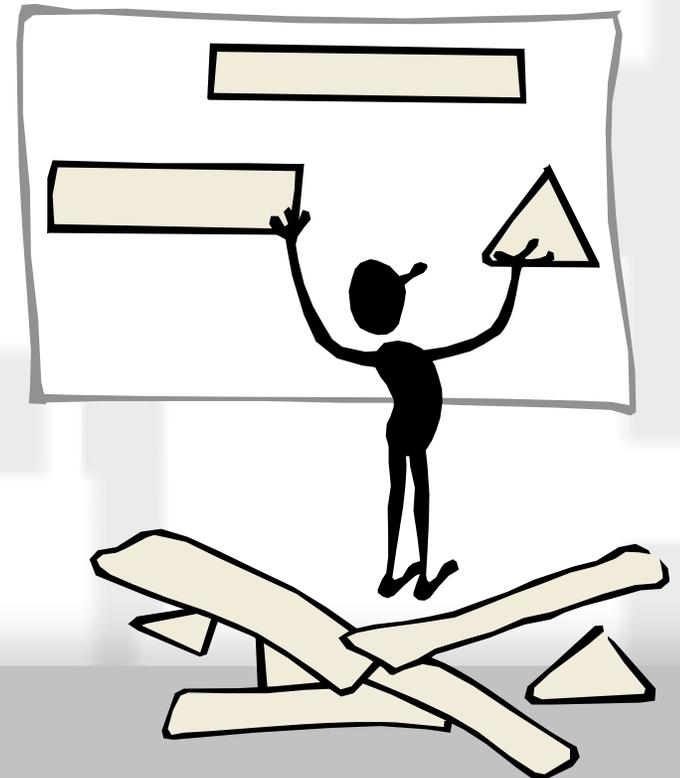
**Per esempio...**

- **...una struttura logica trasmessa solo attraverso la vista, è completamente inaccessibile per un utente non vedente, che esplora la pagina usando un sintetizzatore vocale**

# La struttura di un documento

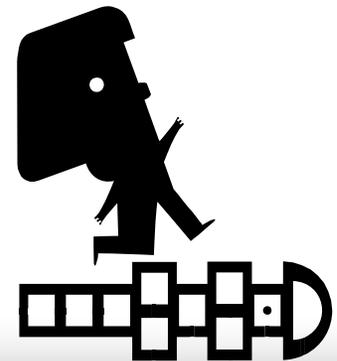
**Un documento correttamente strutturato  
deve avere i contenuti**

- **Ordinati**
- **Gerarchizzati**



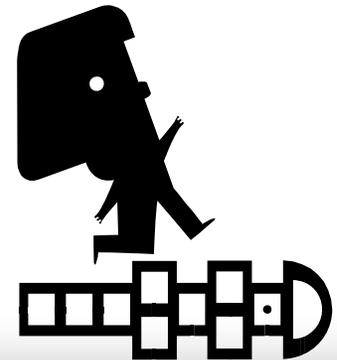
# La struttura di un documento

- Gli *screen reader* consentono di esplorare una pagina saltando da un titolo all'altro, da un link all'altro, da un campo modulo ad un altro...



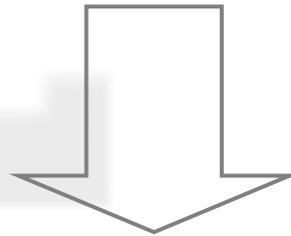
# La struttura di un documento

- Altre funzioni avvertono l'utente sul tipo di Tag che sta per essere “letto”:
  - Paragrafo
  - Lista
  - Tabella...



# La struttura di un documento

- **Contenuti strutturati correttamente...**



- **Il non vedente potrà farsi un'idea della pagina simile a quella che i vedenti riescono a ricavare dall'esame visuale**

# La struttura di un documento

- **Senza un'adeguata marcatura, la struttura logica rimarrà nascosta e sarà difficile per chi non vede comprendere la relazione tra le varie parti che compongono la pagina**

# La struttura di un documento

- **Buona strutturazione dei contenuti**
- **Criteri di presentazione corretti**

**Sono utili per tutti gli utenti**

- **In particolare per quelli affetti da deficit cognitivi, che ricevono notevole aiuto da un'organizzazione di pagina semplice e chiara.**

# La struttura di un documento

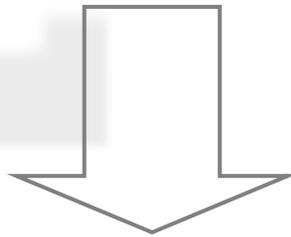
- **Infine una valida strutturazione dei contenuti garantisce migliori risultati ai fini dell'indicizzazione da parte dei motori di ricerca**

# La struttura di un documento

- **Il documento ben strutturato conserva grande chiarezza e comprensibilità se osservato senza i fogli di stile**
  - **WCAG 6.1: produrre documenti che rimangano significativi anche se riprodotti senza fogli di stile**

# La struttura di un documento

## Strutturare un documento



**Marcare strutturalmente i blocchi di testo che lo compongono**

# HTML = struttura logica

- **L'HTML è un linguaggio di marcatura strutturale sin dalle origini**
  - **Nato per suddividere un contenuto testuale in una serie di blocchi organizzati in base a gerarchie ed appartenenze**
  - **Solo in un secondo momento c'è stata l'esplosione dei formati proprietari**

# XHTML: elementi strutturali

- **H1 ... H6:** identificano i sei livelli di titolo previsti da HTML e XHTML (H1 è il più importante, H6 il meno importante)
- **P:** l'elemento per marcare i normali paragrafi di testo
- **OL, UL, LI:** gli elementi per definire liste numerate e non numerate
- **DL, DT, DD:** servono per creare liste di definizioni (DL identifica la lista, DT il termine da definire, DD la sua definizione)
- **DFN:** racchiude una definizione
- **TABLE, TR, TH, TD, THEAD, TBODY, TFOOT, CAPTION, COL, COLGROUP:** definiscono le caratteristiche strutturali di una tabella di dati
- **BLOCKQUOTE, Q, CITE:** da usare per inserire citazioni nel testo
- **FORM, FIELDSET, LEGEND, INPUT, LABEL, BUTTON, SELECT, OPTGROUP, OPTION, TEXTAREA:** servono per creare moduli da inoltrare al server
- **A, LINK:** definiscono collegamenti a risorse interne o esterne al documento
- **ABBR, ACRONYM:** identificano abbreviazioni ed acronimi
- **EM, STRONG:** denotano enfasi e forte enfasi
- **DIV, SPAN:** rappresentano semplici contenitori da collegare a definizioni di lingua o di stile (DIV per contenuti a livello di blocco, SPAN per contenuti in riga)
- **SAMP, CODE:** identificano il primo brani tratti dall'output di programmi, il secondo frammenti di codice di programmazione
- **KBD, VAR:** il primo identifica del testo che l'utente deve introdurre tramite tastiera, il secondo l'istanza di una variabile in un codice di programmazione
- **MAP, AREA:** definiscono una mappa immagine
- **HTML, HEAD, BODY:** rappresentano le principali divisioni strutturali di un documento (X)HTML

# Limite di (X)HTML

**Pochi elementi con cui si possono  
marcare strutturalmente dei blocchi di  
testo**

- **H1...H6** se si tratta di un titolo,
- **BLOCKQUOTE** se si tratta di una citazione,
- **P** se si tratta di un paragrafo
  - Abbiamo escluso le tabelle e le liste

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="it">

<head>
  <title>Documento XHTML 1.1</title>
</head>

<body>
<h1>Un titolo di primo livello</h1>
<p>Un paragrafo.</p>
<ul>
  <li>Voce elenco 1</li>
  <li>Voce elenco 2</li>
  <li>Voce <span>elenco 3</span></li>
</ul>
</body>

</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="it">

<head>
  <title>Documento XHTML 1.1</title>
</head>

<body>
<h1>Un titolo di primo livello</h1>
<p>Un paragrafo.</p>
<ul>
  <li>Voce elenco 1</li>
  <li>Voce elenco 2</li>
  <li>Voce <span>elenco 3</span></li>
</ul>
</body>

</html>
```

**Si nota la presenza di una struttura gerarchica nel codice di questa pagina.**

**Ogni pagina XHTML deve possedere una struttura rigida di questo tipo.**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang=

<head>
  <title>Documento XHTML 1.1</title>
</head>

<body>
<h1>Un titolo di primo livello</h1>
<p>Un paragrafo.</p>
<ul>
  <li>Voce elenco 1</li>
  <li>Voce elenco 2</li>
  <li>Voce <span>elenco 3</span></li>
</ul>
</body>

</html>
```

**Il body contiene  
degli elementi  
nidificati**

**Sono tutti elementi  
“di blocco” tranne  
SPAN che è un  
elemento “di linea”**

# Elementi di Blocco

- **In genere cominciano su nuove righe**
- **Contengono altri elementi di riga o di blocco**
- **Creano strutture più “ampie”**
- **Hanno una formattazione di default**

# Elementi di Linea *(Text level)*

- **Iniziano e terminano sulla stessa linea**
- **Contengono dati o altri elementi di linea**
- **Non hanno una formattazione predefinita**

# Documento X-HTML come gerarchia di elementi

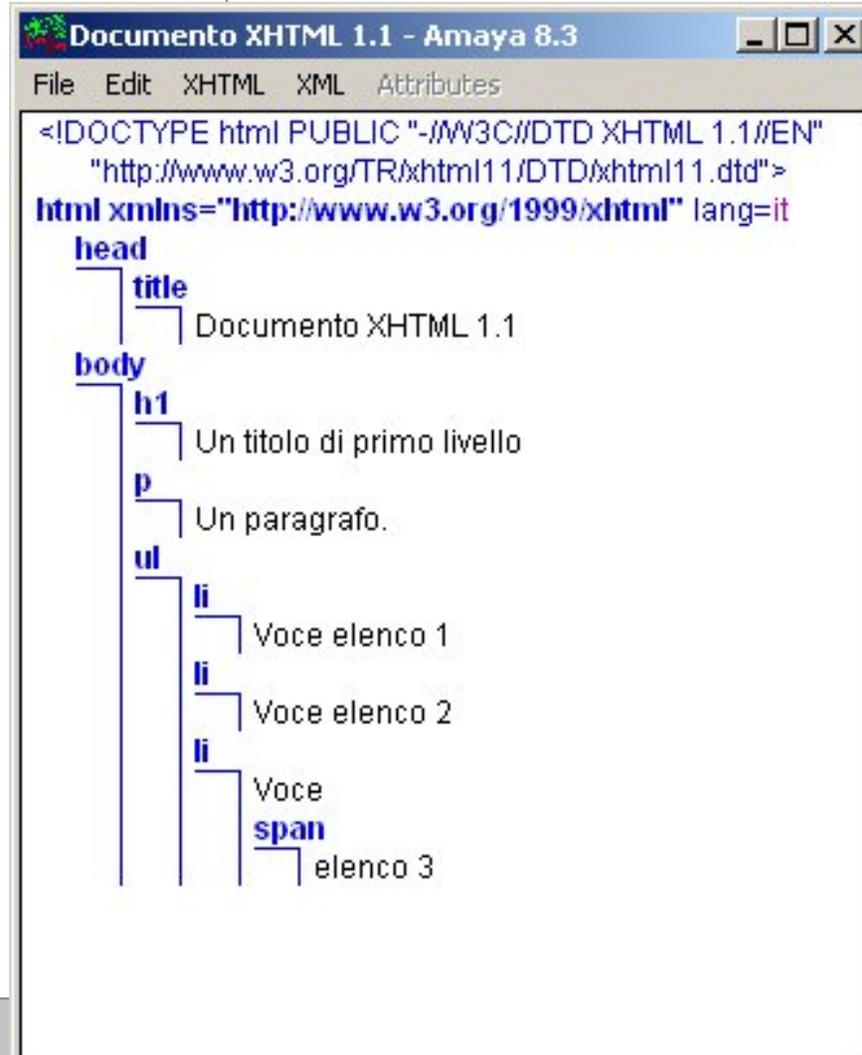
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="it">

<head>
  <title>Documento XHTML 1.1</title>
</head>

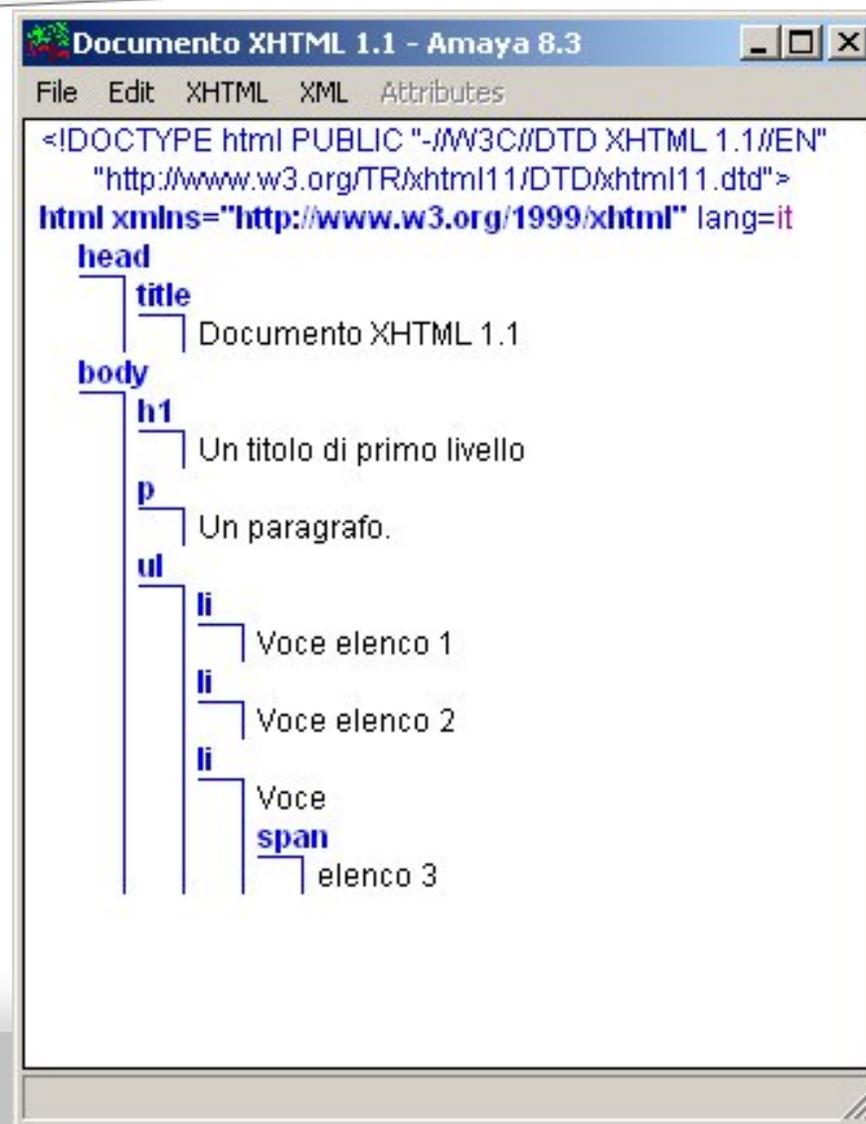
<body>
<h1>Un titolo di primo livello</h1>
<p>Un paragrafo.</p>
<ul>
  <li>Voce elenco 1</li>
  <li>Voce elenco 2</li>
  <li>Voce <span>elenco 3</span></li>
</ul>
</body>

</html>
```



# L'albero degli elementi

- Questa rappresentazione evidenzia la struttura ad albero degli elementi.
  - I tre elementi LI presenti sono tutti “figli” dell'elemento **UL**
  - **P** è uno dei figli di **BODY**.
- L'elemento **HTML** è sempre l'eccezione perchè, come si può facilmente notare, è sprovvisto di “genitori”.



# Elemento *head*

- **Titolo del documento**
- **Parole chiave utili ai motori di ricerca**
- **Altri dati non considerati come contenuto del documento.**
  - I browser di solito non riproducono gli elementi dell'*head* come contenuto
  - Però rendono disponibili agli utenti - attraverso altri meccanismi - le informazioni presenti in *head*

# Elemento *title*

- Ogni documento X-HTML deve avere un elemento *title* nella sezione *head*.
- Poiché gli utenti consultano spesso documenti fuori contesto, gli autori dovrebbero fornire titoli ricchi di informazioni contestuali.
  - Così, invece di un titolo scarno come "*Introduzione*", gli autori dovrebbero fornire un titolo significativo del tipo "*Introduzione all'apicoltura medioevale*"

# Elemento *meta*

- **L'HTML lascia agli autori la possibilità di specificare dei metadati**
  - *Informazioni di contorno su un documento, diverse dal contenuto di un documento*

# Elemento *meta*

Ogni elemento **META** specifica una coppia proprietà/valore.

- L'attributo *name* identifica la proprietà e l'attributo *content* specifica il valore della proprietà.

– Per esempio, la seguente dichiarazione imposta un valore per la proprietà Author:

```
<META name="Author" content="Dave  
Raggett">
```

# Elemento *meta*

- **Un uso comune di META è di specificare le parole chiave che un motore di ricerca può adoperare per accrescere la qualità dei risultati della ricerca**
  - **Quando una serie di elementi META fornisce informazioni dipendenti dalla lingua su un documento, i motori di ricerca possono attivare un filtro sull'attributo lang per visualizzare i risultati della ricerca in base alle preferenze linguistiche dell'utente**

# Elemento *meta*

- Ad esempio...

```
<-- Per chi parla l'inglese americano -->  
<META name="keywords" lang="en-us"  
content="vacation, Greece, sunshine">  
<-- Per chi parla l'inglese britannico -->  
<META name="keywords" lang="en" content="holiday,  
Greece, sunshine">  
<-- Per chi parla francese -->  
<META name="keywords" lang="fr" content="vacances,  
Gr&egrave;ce, soleil">
```

# Righe e paragrafi

- Tradizionalmente gli autori suddividono i loro pensieri e le argomentazioni in sequenze di paragrafi

**L'elemento  $P$  definisce un paragrafo**

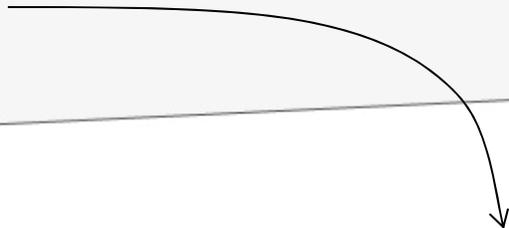
# Elemento *p*

- **Esso non può contenere elementi a livello di blocco (*incluso lo stesso P*)**
- **Non si dovrebbero utilizzare elementi P vuoti**
  - **I programmi utente dovrebbero ignorare gli elementi P vuoti**

# Elementi *Div* e *Span*

- Aiutano a strutturare il documento
- Questi elementi definiscono il contenuto o come in riga (*Span*) o come a livello di blocco (*Div*)
  - Tuttavia non impongono alcun altro idioma presentazionale sul contenuto
  - In pratica sono dei “contenitori” di altri elementi

## I link



- **Ciò che differenzia X-HTML dalla maggior parte degli altri linguaggi di marcatura sono le sue caratteristiche idonee a produrre documenti ipertestuali e interattivi**

# I link

- Un collegamento è una connessione da una risorsa Web ad un'altra
- Un *collegamento* ha due estremità – chiamate *àncore* – ed una direzione
  - Il collegamento parte dall'ancora "sorgente" e punta verso l'ancora "destinazione", che può essere una qualsiasi risorsa Web

# I link: elemento *a*

Risorse esterne al documento

```
<BODY> ...del testo...
```

```
<P>Troverete molto di più nel
```

```
<A href="capitolo2.html">capitolo due</A>
```

```
Date anche uno sguardo alla
```

```
<A href="../immagini/foresta.gif">mappa  
della foresta incantata.</A>
```

```
</BODY>
```

# I link: elemento *a*

Ancore interne al documento

```
<H1>Sommariao</H1>
```

```
<P>
```

```
<A href="#sezione1">Introduzione</A><BR>
```

```
<A href="#sezione2">Informazioni di base</A><BR>
```

```
<A href="#sezione2.1">Una nota piÙ personale</A><BR>
```

*...il resto del sommario...*

*...corpo del documento...*

```
<H2 id="sezione1">Introduzione</H2>
```

*...sezione 1...*

```
<H2 id="sezione2">Informazioni di base</H2>
```

*...section 2...*

```
<H3 id="sezione2.1">Una nota piÙ personale</H3>
```

*...sezione 2.1...*

# I link: elemento *a*

- Come sceglierli?

```
<a href="testo.html">Il testo del disegno di legge  
sull'accessibilità.</a>
```

**Collegamento troppo lungo**

```
Il testo del disegno di legge sull'accessibilità: <a  
href="testo.html">clicca qui</a>.
```

**"Clicca qui" è inadatto a chi usa la tastiera**

```
Il <a href="testo.html">testo</a> del disegno di legge  
sull'accessibilità.
```

**Soluzione accettabile**

## I link: elemento *a*

- Il parametro `title` è utile ad arricchire il collegamento con informazioni utili all'utente
  - Spesso il solo testo “sottolineato” non è sufficiente a fornire tutti i dettagli sulla risorsa collegata

# I link: elemento *a* - *Title*

Il `<a href="testo.html" title="Gli articoli del disegno di legge">testo</a>` del disegno di legge sull'accessibilità.

**Attributo "title":** descrive la pagina di destinazione

Il `<a href="testo.zip" title="documento .zip da 100Kb contenente il disegno di legge">testo</a>` del disegno di legge sull'accessibilità.

**Attributo "title":** descrive la tipologia di file che si sta per scaricare

Il `<a href="testo.html" title="[link su nuova finestra] Gli articoli del disegno di legge" target="_blank">testo</a>` del disegno di legge sull'accessibilità.

**Attributo "title":** informa che si aprirà una nuova finestra

# Elemento *link*

- Anch'esso definisce un collegamento
- A differenza di A, esso può apparire **soltanto nella sezione HEAD** di un documento
- **LINK non ha contenuto, e comunica informazioni di collegamento che possono essere riprodotte dai programmi utente in diversi modi**
  - ad es., una barra di navigazione con un menu a tendina di collegamenti

# Elemento *link*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Capitolo 2</TITLE>
  <LINK rel="Index" href="../indice.html">
  <LINK rel="Next" href="Capitolo3.html">
  <LINK rel="Prev" href="Capitolo1.html">
</HEAD>
...il resto del documento...
```

## Rel

Specifica la relazione del documento collegato con quello corrente

## Next

Rimanda al documento successivo in una sequenza lineare di documenti

## Prev

Rimanda al documento precedente in una serie ordinata di documenti.  
Alcuni programmi utente supportano anche il sinonimo "Previous"

# Elemento *link*

**Si usa anche per collegare un documento X-HTML a:**

- **Un CSS esterno**
- **Motori di ricerca**
  - Collegamenti a versioni alternative di un documento, scritte in un'altra lingua naturale
  - Collegamenti a versioni alternative di un documento, destinate a media differenti, ad esempio una versione particolarmente adatta per essere stampata
  - Collegamenti alla pagina iniziale di un insieme di documenti

# Le intestazioni *hx*

- **Un elemento di intestazione descrive brevemente l'argomento della sezione che esso introduce**
  - **Le informazioni di intestazione possono essere usate dai programmi utente per ricavare un sommario per un documento**

# Le intestazioni *hx*

**Vi sono in HTML sei livelli di intestazione**

- **h1 il più “importante”**
- **h6 il meno “importante”**
  - I programmi utente di tipo visuale riproducono di solito le intestazioni più importanti con caratteri più grandi di quelli usati per le meno importanti
  - Questo accade “grazie” a un CSS di default

# Le intestazioni *hx*

- **Le intestazioni rappresentano l'elemento più utilizzato per “strutturare” un documento X-HTML**
  - **Il loro utilizzo non è sempre semplice e banale... esistono delle regole da seguire**

# Le intestazioni *hx*

- È scorretto utilizzare elementi `<hx>` per incrementare la dimensione del testo
- In questo modo si danneggia la semantica del documento, creando difficoltà di accesso tramite tecnologie assistive

# Le intestazioni *hx*

- L'ordine di posizionamento dei titoli deve essere rispettato
  - Uno o più elementi `<h2>` devono seguire un elemento `<h1>`, mentre uno o più elementi `<h3>` devono seguire elementi `<h2>`
- È quindi scorretto saltare un titolo
  - È errato passare da `<h1>` a `<h3>` senza valorizzare l'elemento `<h2>`

# Le intestazioni *hx* - esempio

```
<h1>Titolo</h1>
```

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

```
<h2>Titolo argomento</h2>
```

```
<h3>Capitolo informativo</h3>
```

```
<p>... testo di esempio ..... </p>
```

```
<h2>Titolo argomento</h2>
```

```
<h3>Capitolo informativo</h3>
```

```
<p>... testo di esempio ..... </p>
```

```
<p>... testo di esempio ..... </p>
```

```
<h3>Capitolo informativo</h3>
```

```
<p>... testo di esempio ..... </p>
```

```
<h1>Titolo</h1>
```

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

# Le liste

**L'HTML offre agli autori diversi strumenti per specificare elenchi di informazioni. Essi possono contenere:**

- **Informazioni non ordinate**
- **Informazioni ordinate**
- **Definizioni**

# 1 - Le liste *ul*

```
<ul>  
<li>Informazioni non  
ordinate.</li>  
<li>Informazioni  
ordinate.</li>  
<li>Definizioni.</li>  
</ul>
```

Codice

- **Informazioni non ordinate.**
- **Informazioni ordinate.**
- **Definizioni.**

Risultato

## 2 – Le liste *ol*

```
<ol>
  <li>Mescolare
    accuratamente gli
    ingredienti
    secchi.</li>
  <li>Versare gli
    ingredienti
    liquidi.</li>
  <li>Mescolare per 10
    minuti.</li>
  <li>Cuocere in forno per
    un'ora a 300
    gradi.</li>
</ol>
```

Codice

1. Mescolare accuratamente gli ingredienti secchi.
2. Versare gli ingredienti liquidi.
3. Mescolare per 10 minuti.
4. Cuocere in forno per un'ora a 300 gradi.

Risultato

## 3 – Liste di definizioni *dl*

- In questo caso le voci di elenco consistono di due parti:
- Un termine e una descrizione. Il termine è fornito per mezzo dell'elemento **DT** ed è vincolato a contenuto in riga. La descrizione è data con l'elemento **DD** che racchiude contenuto a livello di blocco.

# 3 – Liste di definizioni *dl*

```
<dl>
  <dt><strong>prezzo più
basso</strong></dt>
  <dd>la nuova versione di
questo prodotto costa
significativamente meno
della precedente!</dd>
  <dt><strong>più facile da
adoperare</strong></dt>
  <dd>abbiamo modificato il
prodotto in modo che sia più
facile da usare! </dd>
  <dt><strong>sicuro per i
bambini</strong></dt>
  <dd>potete lasciare i vostri
bambini da soli in una stanza
con questo prodotto ed essi
non si faranno male (non è
una garanzia). </dd>
</dl>
```

Codice

- **Prezzo più basso**
  - La nuova versione di questo prodotto costa significativamente meno della precedente!
- **Più facile da adoperare**
  - Abbiamo modificato il prodotto in modo che sia più facile da usare!
- **Sicuro per i bambini**
  - Potete lasciare i vostri bambini da soli in una stanza con questo prodotto ed essi non si faranno male (non è una garanzia).

Risultato

# Le liste – attributi deprecati

- **Type = style-information**
  - Imposta lo stile di una voce di elenco. I valori attualmente disponibili sono pensati per programmi utente di tipo visuale
- **Start = number [CN]**
  - Solo per OL.
  - Specifica il numero iniziale della prima voce in un elenco ordinato. Il numero iniziale predefinito è "1". Si noti che mentre il valore di questo attributo è un intero, l'etichetta corrispondente può essere non numerica.
- **Value = number [CN]**
  - Solo per LI.
  - Questo attributo imposta il numero della voce di elenco corrente.
- **Compact [CI]**
  - Quando impostato, questo attributo booleano suggerisce ai programmi utente di tipo visuale di riprodurre l'elenco in una forma più compatta.

# Le liste – uso corretto

- **Le liste rappresentano uno degli strumenti che possono essere utilizzati per strutturare correttamente un documento X-HTML**

# Le liste – uso corretto

- **Non bisogna usare le liste come un mezzo per creare del testo rientrato**
- **Questo è un problema stilistico e dovrà gestito in modo appropriato per mezzo dei fogli di stile**

# Le tabelle

- E' uno degli elementi più utilizzati in tutte le pagine Web
- Il suo scopo “primitivo” è quello di ospitare dati all'interno di righe e colonne

# Le tabelle

- **In realtà le tabelle possono contenere molti altri tipi di informazioni, non solo parole e numeri, per esempio immagini e altre tabelle**

# Le tabelle

- **E' diventata prassi comune l'uso**  
**(meglio dire l'abuso) delle tabelle per**  
**la realizzazione del layout di un sito,**  
**tradendo quello che era lo scopo della**  
**loro creazione**

# Le tabelle

- **Attraverso l'annidamento di tabelle dentro tabelle è possibile realizzare impaginazioni complesse**
  - In questa maniera lo sviluppatore riesce con una certa facilità a “controllare” la griglia di layout ed il suo comportamento

# Le tabelle

- **Inoltre i software del tipo WYSIWYG** (*What You See Is What You Get: ciò che vedi è ciò che otterrai*) **Si basano quasi esclusivamente sull'uso delle tabelle**
- **Il software crea un codice che, nella migliore delle ipotesi è ridondante e disorganizzato, nella maggior parte dei casi risulta anche sintatticamente errato**

# Le tabelle

## Il risultato???

- **Pagine pesanti e lente da scaricare ed elaborare**
- **Pagine di solito non utilizzabili da dispositivi come i PDA**
- **Pagine inaccessibile da parte di utenti non vedenti che utilizzano screen reader**

# Elemento *Table*

- L'elemento `table` contiene tutti gli elementi che specificano didascalia, righe, contenuto e formattazione

```
<table border="1" summary="Questa
tabella fornisce alcune statistiche
sul moscerino della frutta: altezza e
peso medi, e percentuale di individui
con gli occhi rossi (per maschi e
femmine).">
```

```
<caption><em>una tabella
esemplificativa con celle
unificate</em></caption>
```

```
<tr>
```

```
<th rowspan="2"></th>
```

```
<th colspan="2">Media</th>
```

```
<th rowspan="2">Occhi<BR>rossi</th>
```

```
</tr>
```

```
<tr>
```

```
<th>altezza</th>
```

```
<th>peso</th>
```

```
</tr>
```

```
<tr>
```

```
<th>maschi</th>
```

```
<td>1.9</td>
```

```
<td>0.003</td>
```

```
<td>40%</td>
```

```
</tr>
```

```
<tr>
```

```
<th>femmine</th>
```

```
<td>1.7</td>
```

```
<td>0.002</td>
```

```
<td>43%</td>
```

```
</tr>
```

```
</table>
```

# Esempio

*Una tabella esemplificativa con celle unificate*

	Media		Occhi rossi
	altezza	peso	
Maschi	1.9	0.003	40%
Femmine	1.7	0.002	43%

# Oggetti, immagini, applet...

- **XHTML** consente agli autori di includere nelle loro pagine immagini, **applet** (*programmi che sono automaticamente scaricati ed eseguiti sulla macchina dell'utente*), **brani video ed altri documenti HTML**

# Oggetti, immagini, applet...

<b>Tipo di inclusione</b>	<b>Elemento specifico</b>	<b>Elemento generico</b>
<i>Immagine</i>	IMG	OBJECT
<i>Applet</i>	APPLET <i>Disapprovato</i>	OBJECT
<i>Un altro documento HTML</i>	IFRAME	OBJECT

# Immagini: *img*

```

```

```
<object  
data="http://www.unaditta.com/famiglia.png  
" type="image/png">  
Una foto della mia famiglia al lago  
</object>
```

# *Img : attributi alt e longdesc*

- ***Alt***
  - Descrizione testuale alternativa
- ***Longdesc***
  - Link a pagina testuale alternativa

# *Object*

- **La maggior parte dei programmi utente è in grado di rappresentare tipi di dati comuni quali testo, immagini GIF, colori, caratteri...**
- **Per rappresentare dati non supportati, essi devono avvalersi di applicazioni esterne**

# *Object*

- ***OBJECT*** consente di controllare se i dati debbano essere riprodotti esternamente o da un qualche programma che rappresenta i dati nel programma utente

# *Object*

Di solito (non sempre) è necessario specificare tre tipi di informazioni:

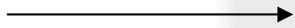
- **L'implementazione dell'oggetto incluso**
  - Se, per esempio, l'oggetto incluso è un *applet orologio*, l'autore deve indicare la posizione del codice eseguibile dell'applet
- **Posizione dei dati da riprodurre**
- **Valori aggiuntivi richiesti dall'oggetto al momento dell'esecuzione**
  - Per esempio, valori iniziali di alcuni parametri

# *Object* : esempio

```
<OBJECT  
classid="http://www.miosito.it/orologio.py">  
<PARAM name="altezza" value="40"  
valuetype="data" />  
<PARAM name="larghezza" value="40"  
valuetype="data" />  
Questo programma è in grado di visualizzare  
l'ora di sistema  
</OBJECT>
```

Esempio di *Object* con parametri di inizializzazione

# *Object* alternativi

- ***Object*** offre un meccanismo per specificare **oggetti alternativi**
- **Se un programma utente non è in grado di riprodurre *l'object* più esterno, esso tenta di riprodurre i contenuti**
  - **Questi possono essere costituiti da un altro elemento *object*, e così via...** 

# *Object* alternativi

```
<OBJECT title="La Terra vista dallo spazio"
classid="http://www.observer.mars/Terra.py">
  <!-- Altrimenti prova il video MPEG -->
  <OBJECT data="Terra.mpeg"
type="application/mpeg">
  <!-- Oppure prova l'immagine GIF -->
  <OBJECT data="Terra.gif" type="image/gif">
  <!-- Altrimenti riproduci il testo -->
  La Terra vista dallo spazio.
</OBJECT>
</OBJECT>
</OBJECT>
```

# *Object* alternativi

```
<object type="application/x-shockwave-flash"
  data="movie.swf" width="400" height="300">
  <param name="movie" value="movie.swf" />
  
  <ul title="opzioni alternative">
    <li><a href="pagina1.htm">Pagina 1</a>
    <li><a href="pagina2.htm">Pagina 2</a>
  </ul>
</object>
```

# Mappe immagine

- **Consentono di specificare regioni di un'immagine o di un oggetto e di assegnare una determinata azione a ciascuna regione**
  - es.: recuperare un documento, eseguire un programma...

# Mappe immagine

- **Lato Client**

- Se l'utente attiva con il mouse una regione di una mappa immagine lato client, il programma utente seleziona il collegamento specificato per quella regione e lo esegue

- **Lato Server**

- In questo caso, le coordinate dei pixel al momento del clic sono inviate al server specificato dall'attributo *href* dell'elemento *A*
- Il server interpreta le coordinate ed esegue una certa azione

# Mappa immagine lato client

```

<map id="csmap" name="csmap">
<area shape="rect" alt="Sardegna"
  coords="114,238,160,31" href="1.htm" />
<area shape="poly" alt="Sicilia"
  coords="304,330,292,354" href="2.htm" />
<area shape="poly" alt="Calabria"
  coords="304,283,319,12" href="3.htm" />
</map>
```

# Mappa immagine lato server

```
<p>
  <a href="generamappa.asp">
    
  </a>
  <a href="inizio.htm">Punto di Partenza</a>
<br />
  <a href="fine.htm">Punto di Arrivo</a>
</p>
```

I link testuali rappresentano la funzionalità alternativa per gli utenti che non possono usare il mouse

# I moduli

**E' una sezione di un documento XHTML contenente:**

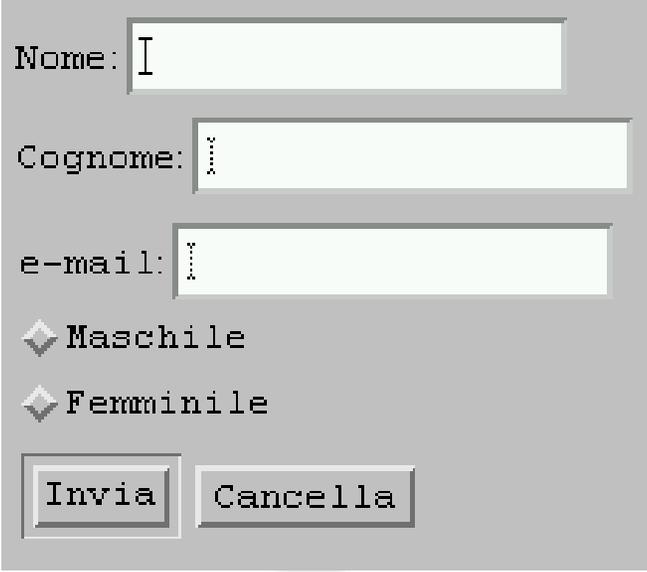
- **Normale contenuto**
- **Codice di marcatura**
- **Controlli**
  - **Caselle di spunta, pulsanti radio, menu...**
- **Etichette relativi ai controlli**

# I moduli

```
<form action="page.asp" method="post">
  <p>
    <label for="nome">Nome: </label>
    <input type="text" id="nome"> <br />
    <label for="cognome">Cognome: </label>
    <input type="text" id="cognome"> <br />
    <label for="email">e-mail: </label>
    <input type="text" id="email"> <br />
    <input type="radio" name="sesso"
    value="Maschile"> maschile<br />
    <input type="radio" name="sesso"
    value="Femminile"> femminile<br />
    <input type="submit" value="Invia">
    <input type="reset">
  </p>
</form>
```

# Moduli: i Controlli

- Elemento *Input*, parametro *type*:
  - Text
  - Password
  - Checkbox
  - Radio
  - Submit
  - Reset
  - Image
  - Button
  - Hidden
  - File



Nome:

Cognome:

e-mail:

Maschile

Femminile

# Moduli: elementi *Select*, *Option*

- ***Select*** crea un menu
  - Ogni scelta del menu corrisponde a un elemento ***Option***
  - Un ***Select*** deve contenere almeno un ***Option***

## Moduli: elemento *Optgroup*

- **Permette di raggruppare le scelte logicamente**
  - **E' utile in lunghi elenchi di opzioni**
  - **Tutti gli elementi *Optgroup* devono essere specificati direttamente all'interno di un elemento *Select***

# Moduli: elemento *Optgroup*

```
<select name="Province">
  ....
  <optgroup label="Lombardia">
    <option value="MI">Milano</option>
    <option value="BG">Bergamo</option>
  </optgroup>
  <optgroup label="Piemonte">
    <option value="TO">Torino</option>
    <option value="AT">Asti</option>
  </optgroup>
  ....
</select>
```



Milano

**Lombardia**

Milano

Bergamo

**Piemonte**

Torino

Asti

## Moduli: elementi *FieldSet*, *Legend*

- Un problema frequente è il raggruppamento di gruppi di campi di inserimento
- Un buon sistema di raggruppamento è dato dall'elemento *Fieldset* e dal complementare elemento *Legend*

# Moduli: elementi *FieldSet*, *Legend*

```
<fieldset>
  <legend>Codice di Accesso</legend>
  <p>
    <label for="password" accesskey="N">
      Password:
      <input type="password" size="5" id="password"
        name="password" />
    </label>
  </p>
</fieldset>
<fieldset>
  <legend>Informazioni Personali</legend>
  <!-- qui vanno i campi indirizzo, ecc. ecc. -->
</fieldset>
```

Codice di Accesso

Password:

Informazioni Personali

# Moduli: ordine di tabulazione

```
<form id="FormLogin" method="post" action="login.asp">
  <p>Nome Utente:
    <input type="text" size="40" value="nomeutente"
      id="nomeutente" tabindex="5" />
  </p>
  <p>Password:
    <input type="password" size="40" value=""
      id="password" tabindex="6" />
  </p>
  <p>
    <input type="submit" value="Accedi" id="accedi"
      tabindex="7" />
  </p>
</form>
```

# Moduli: attributo *Accesskey*

**Accesskey** si applica agli elementi:

- *a*
- *area*
- *button*
- *Input*
- *label*
- *legend*
- *textarea*

*FORM*

```
<FORM action="..." method="post">  
  <P>  
    <LABEL for="nutente" accesskey="U">  
      Nome utente </LABEL>  
    <INPUT type="text" name="utente"  
      id="nutente">  
  </P>  
  ...  
</FORM>
```

# Moduli: attributo *Accesskey*

- **Accesskey** funziona in modo differente a seconda del browser e del sistema operativo utilizzato
  - *Ecco una delle ragioni per cui il progetto WAI non concentra la sua attenzione SOLO sulle WCAG, ma si rivolge anche agli user agent ed agli strumenti di sviluppo*

# Moduli: attributo *Accesskey*

## Attivare le Access Key:

- S.O. MS Windows/MSIE: *Alt [key] Enter*
- S.O. MS Windows/Mozilla, Netscape, K-Meleon: *Alt [key]*
- S.O. MS Windows/Opera: *Shift + Esc [key]*
- S.O. Macintosh/MSIE: *Ctrl [key] Enter*
- S.O. Macintosh/Mozilla, Netscape: *Ctrl [key]*
- S.O. Macintosh/Opera: *Shift + Esc [key]*
- S.O. Linux Mandrake/Galeon/Mozilla: *Alt [key]*

# Moduli: attributo *Accesskey*

## Browser che supportano le Access Key:

- MS Windows Internet Explorer (*versione 4.xx o successiva*)
- Mozilla (*9.xx, versione 1.xx o successiva*)\*
- Netscape Navigator (*versione 6.xx o successiva*)\*
- Opera (*versione PC: 7.x o successiva, versione Mac: 6.x o successiva*)
- K-Meleon (*versione 6.xx o successiva*)
- Spirit MetaBrowser (*versione 1.xx o successiva*)
- Galeon (*versione 1.xx o successiva*)

# Moduli: attributo Accesskey

- **Browser che *non* supportano le Access Key:**
  - **Opera** (*versione PC: sino alla 7.x, versione Mac: sino alla 6.x*)
  - **Netscape Navigator** (*versione 4.xx*)
  - **Konqueror** (*versione 2.xx*)

# Moduli: attributo Accesskey

- **Eccezioni:**
  - **Con Mac OS 9**, le access keys non sono supportate da Internet Explorer 4, Internet Explorer 4.5, Mozilla, Netscape 4 e Opera 5.
  - **Con Mac OS X 10**, le access keys non sono supportate da Mozilla e OmniWeb 4

# Problemi con *Accesskey*

**Molte combinazioni di tasti (ad esempio ALT-F) sono già utilizzate:**

- Dai browser come tasti di accesso rapido**
- Dalle tecnologie assistive come tasti di scelta rapida**

# Problemi con *Accesskey*

**Se si desidera rispettare l'universalità del Web, ossia rendere accessibili le pagine a qualsiasi utente di qualsiasi nazionalità esistono solamente tre tasti liberi:**

**/ (slash)**

**\ (backslash)**

**] (parentesi quadra chiusa)**

# Problemi con Accesskey

- Una soluzione potrebbe essere l'uso dei caratteri numerici
- Così potranno esserci problemi *solo* con utenti che utilizzano tastiere con lingue orientali o con alcuni screen readers:
  - *IBM Home Page Reader*, ad esempio, con la combinazione ALT + 1 effettua la lettura dei titoli della pagina (elementi "h")

# Problemi con Accesskey

- **Per esempio, il Governo Canadese ha espressamente sconsigliato di utilizzare le accesskey per i siti delle pubbliche amministrazioni:**
  - *[http://www.cio-dpi.gc.ca/clf-upe/6/skip\\_e.asp](http://www.cio-dpi.gc.ca/clf-upe/6/skip_e.asp)*

# Problemi con *Accesskey*

- Altri consigliano di creare una “*dichiarazione di accessibilità*” e di assegnarvi l’accesskey ‘0’
  - Qui andrebbero indicate tutte le accesskey e tutte le funzionalità per l’accesso al sito web ed ai suoi contenuti

```
<a href="/guida.html"  
accesskey="0">Accessibilità</a>
```

## Attributi comuni: *id*

- Questo attributo assegna un identificativo ad un elemento
- Il suo valore deve essere *univoco all'interno di un documento*
  - Quasi tutti gli elementi XHTML possono essere identificati da un id

# Attributi comuni: *id*

***Id* può essere utilizzato:**

- **Come selettore di fogli di stile**
- **Come àncora di destinazione per i link**
- **In uno script per riferirsi ad un certo elemento**
- **Come nome di un elemento OBJECT dichiarato**
- **Per altri scopi di elaborazione dei programmi utente**
  - Per identificare i campi durante l'estrazione di dati da pagine XHTML verso un database
  - Per tradurre documenti XHTML in altri formati

# Attributi comuni: *class*

- **Class** assegna un nome di classe (*o un insieme di nomi di classe*) ad un elemento
- Un qualsiasi numero di elementi può essere assegnato allo stesso nome (*o agli stessi nomi*) di classe
  - Quasi tutti gli elementi XHTML possono essere associati ad una classe

# Attributi comuni: *class*

## **Class è utilizzato:**

- **Come un selettore di fogli di stile**
  - Allo scopo di assegnare certe informazioni di stile ad un insieme di elementi
- **Per scopi generali di elaborazione da parte dei programmi utente**

## Attributi comuni: *Title*

- L'attributo *Title* offre informazioni per la consultazione relativa all'elemento cui si riferisce
- Può “commentare” un qualsiasi numero di elementi
  - Da non confondersi con l'elemento *Title*, che offre informazioni su un intero documento e può apparire una volta soltanto

# Attributi comuni: *Title*

I valori di *title* possono essere rappresentati dai programmi utente in diversi modi:

- **Browser visuali:** mostrano frequentemente il titolo come un "consiglio utile"
- **Screen reader:** possono dire a voce quanto presente nel *title*
  - Per esempio, impostare *title* in un collegamento consente ai programmi utente di dire agli utenti qualcosa sulla natura della risorsa collegata