

Linguaggi di markup: XHTML

Contenuti, stili, strutture

- Gli strumenti WYSWYG (e.g. Word) integrano
 - contenuti
 - stili (formati)
 - strutture
- Linguaggi generici come XML separano tutto
 - contenuti: XML
 - stili: CSS
 - strutture: XSLT

Extensible Markup Language (XML)

- Il testo è strutturato usando *tag* ben formati
 - apertura: <tag>
 - chiusura: </tag>
 - unici: <tag />
- I *tag* possono avere attributi
<tag attr1="valore1" attr2="valore2">

Caratteri speciali

- I *tag* sono formati da "<" e ">", quindi per indicare "<" si usa "<", ">" per ">"
 - anche "&" si deve scrivere in un altro modo: "&"

XHTML

- XHTML è una evoluzione di HTML basata su XML
- La struttura è interpretata in modo fisso dai navigatori
- Gli stili si basano invece su CSS (in HTML erano specificati come attributi)

Struttura di un documento XHTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE ...
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

} titolo, stili

Partizioni

- Sezioni: <h1> ... <h8>
- Paragrafi: <p></p>
- Capoversi:

Iper-collegamenti

- Sono formati da un riferimento ed una destinazione
 - destinazione remota
 - riferimento:
 - destinazione locale
 - riferimento:
 - destinazione:

Tabelle

- Sono racchiuse nei *tag* `<table></table>`
- Sono specificate per righe: `<tr></tr>`
- Ogni riga racchiude i dati: `<td></td>`
- Esempio

```
<table>  
<tr><td>1</td><td>2</td></tr>  
<tr><td>3</td><td>4</td></tr>  
</table>
```

1	2
3	4

Immagini

- Possono essere di tipo JPG o PNG
- Vengono inserite (nel punto esatto) con il *tag* ``
- Possono essere allineate a sinistra o a destra tramite uno stile

Le liste

- Possono essere
 - ordinate: ``
 - non ordinate: ``
- I singoli elementi si indicano tra ``
``
 `1`
``

Le definizioni

- Sono costituite da
 - delimitatori: `<dl></dl>`
 - testo: `<dt></dt>`
 - definizione: `<dd></dd>`
- Esempio:
`<dl>`
 `<dt>Concetto</dt>`
 `<dd>spiegazione del concetto</dd>`
`</dl>`

Concetto
spiegazione
del concetto

Gli stili in cascata (CSS)

- Si chiamano così perché ereditano le impostazioni dagli elementi precedenti
- Permettono di separare lo stile dal contenuto
- Possono essere interni al documento o esterni

Stili in linea, interni ed esterni

- Stili in linea: attributo `style`
- Stili interni

```
<style type="text/css">
...
</style>
```
- Stili esterni

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

Sintassi degli stili CSS

- `selettore { propr1:valore1; ..; proprn:valoren }`
- Un selettore può
 - avere un nome: `#nome`
 - modificare un *tag*: `tag`
 - specificare una classe di un tag: `tag.classe`

Stili del testo

- Famiglia
 - con grazie: `font-family: times`
 - senza grazie: `font-family: arial`
- Alterazioni
 - corsivo: `font-style: italic`
 - grassetto: `font-weight: bold`

Stile dei paragrafi

- Allineamento

- `text-align: justify`

- Rientri

- `text-indent: 0.5cm`

- Interlinea

- `line-height: 0.7cm`

Esempi di CSS

```
#grassetto-corsivo {  
    font-weight: bold;  
    font-style: italic;  
}
```

```
p.giallo {  
    color: #00FFFF;  
}
```

Esempi d'uso di CSS in XHTML

- Nomi: `<p id="grassetto-corsivo">`
- Tag: `<p>`
- Classi di tag: `<p class="giallo">`

Applicare gli stili ad aree particolari

- Si possono delimitare aree alle quali applicare stili diversi
 - il tag `div` crea "blocchi" nel testo
 - il tag `span` lascia il contenuto in linea

- Esempio

- `span` **cambia il testo** così

Invece un blocco come questo deve essere creato usando `div`