

Università Degli Studi di Ferrara

Corso di Laurea in Informatica - A.A. 2021 - 2022

Basi di Dati e Laboratorio

Lez. HTML - CSS - PHP

HTML

HyperText Markup Language

NON È: Un linguaggio di programmazione

È: Un linguaggio a marcatori

HTML

Lo scopo del linguaggio HTML è quello di **rappresentare** (cioè visualizzare) le informazioni.

Esistono altri linguaggi a marcatori che hanno scopi diversi dalla rappresentazione delle informazioni, come ad esempio l'XML che è orientato alla **descrizione** delle informazioni.

Modello client-server

Il modello di comunicazione client-server è basato su due entità:

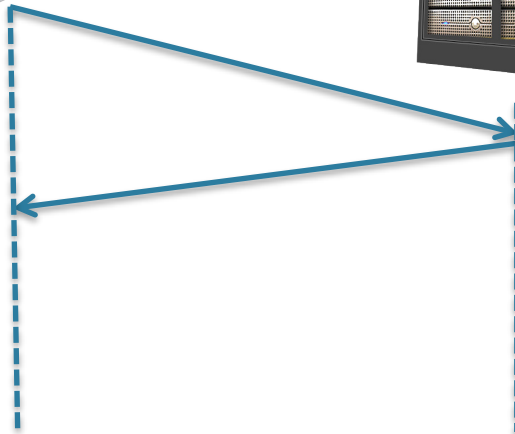
Client (Browser)

Effettua la richiesta di accesso (visita) di una pagina web al Web Server

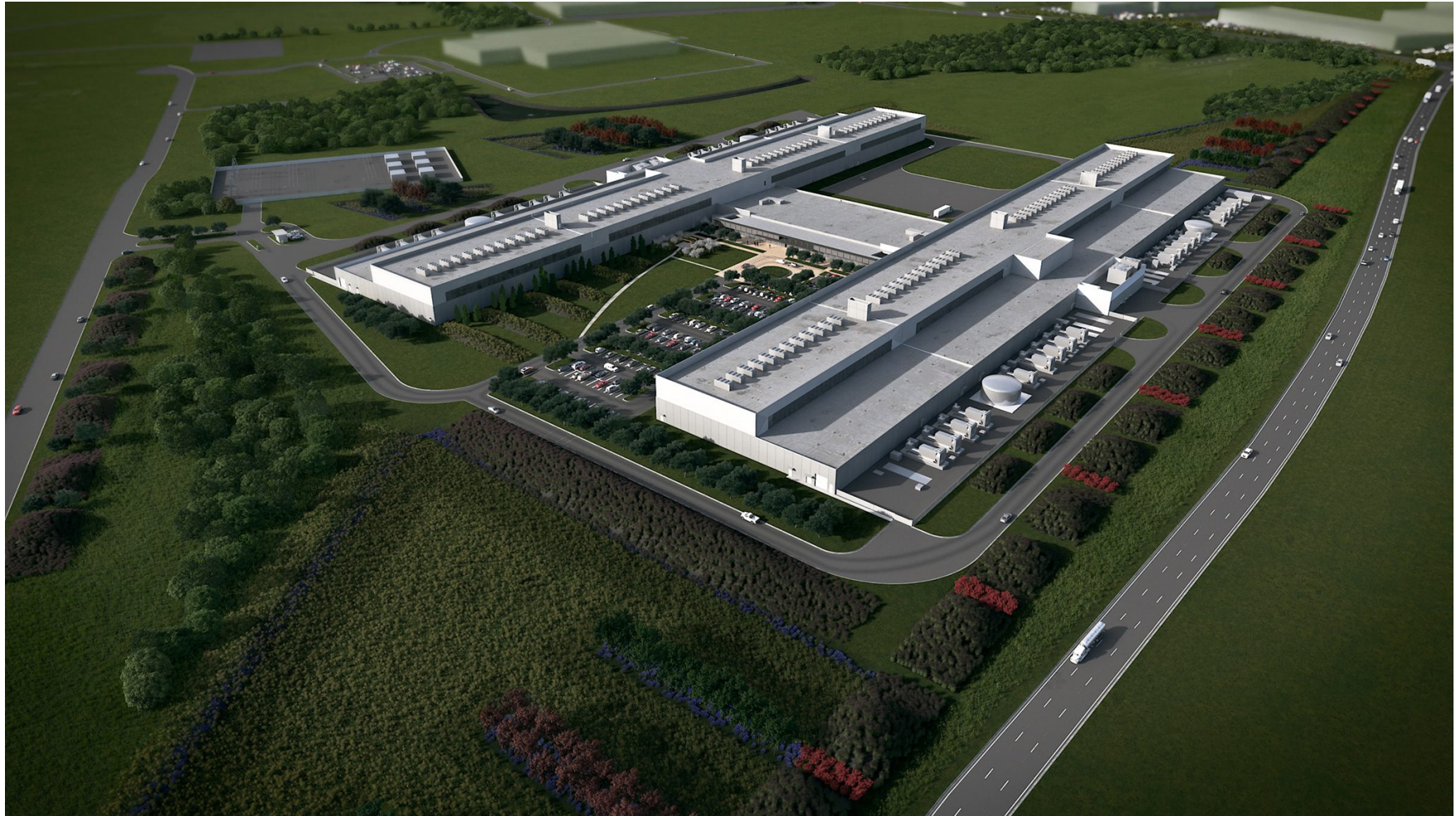


Web Server

Risponde alla richiesta inviando un documento HTML



Server



Server



Server



I Tag HTML

I **Tag** sono gli elementi fondamentali del linguaggio HTML

Sono: delle parole chiave racchiuse fra parentesi angolari < ... >

Usati: (solitamente) in "coppia" come ad esempio: <p> e </p>

Consentono :

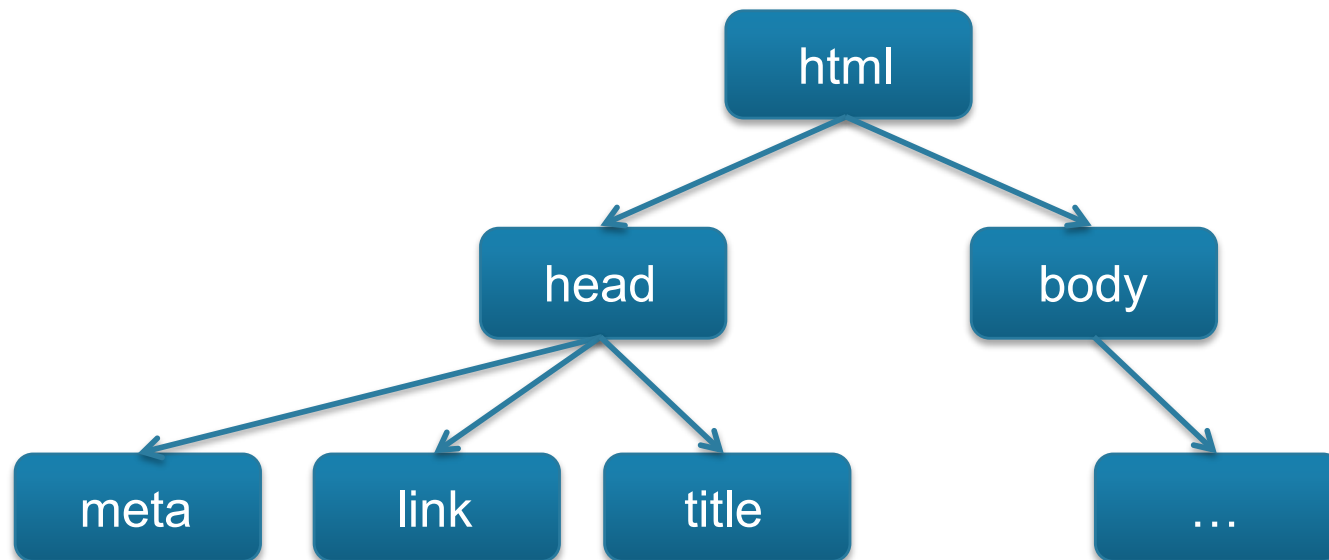
- Formattare il testo (h, p, strong, ...);

- Definire hyperlink (a);

- Raccogliere informazioni dall'utente (form).

HTML

Un documento HTML ha una struttura ad **albero** i cui elementi (i tag) sono annidati creando una gerarchia



Tag di struttura

<!DOCTYPE html>

<html>

<html> è il tag radice

<head>

<head> identifica la sezione header

</head>

<body>

<body> identifica il corpo del documento

</body>

</html>

Tag di blocco e liste

`<body>`

`<p></p>`

`<hx></hx>`

`<a>`

`<div></div>`

``

``

``

``

``

`
`

`</body>`

Tutti i tag di blocco o di lista vanno racchiusi all'interno del tag `body`

I tag `ol` ed `ul` vengono utilizzati per gli elenchi, i primi per gli elenchi ordinati, i secondi per gli elenchi non ordinati

Il tag `li` viene utilizzato per gli elementi degli elenchi e devono essere all'interno dei tag `ul` o `ol`

I tag `img` e `br`, sono tag "speciali" che possono non avere il tag di chiusura

Attributi dei Tag

Tutti i Tag HTML ammettono attributi.

Gli attributi sono elementi che vengono inseriti nel tag di apertura per meglio definire il comportamento dei tag e sono inseriti con la seguente formato:

`attributo="valore"`

Ogni tag ammette uno specifico insieme di attributi, ad esempio il tag `<a>` ammette l'attributo `href`:

` Io sono un link `

Paragrafo

<p>Questo è un paragrafo</p>

Si utilizza il tag
 per andare a capo.

Titoli

<h1>Questo è il titolo principale</h1>

<h2>Questo è un Sottotitolo</h2>

<h3>Questo è un Sottotitolo</h3>

<h4>Questo è un Sottotitolo</h4>

<h5>Questo è un Sottotitolo</h5>

<h6>Questo è un Sottotitolo</h6>

All'interno di una pagina, ci deve essere un solo titolo h1, ma è possibile inserire più titoli h2. È importante che i titoli h2 ... h6 siano ben gestiti gerarchicamente all'interno della pagina.

Ancore

`Questo è un lin`

Il tag `<a>` ha 3 attributi:

- `href`: Specifica l'indirizzo a cui rimandare l'utente;
- `target`: Specifica il comportamento per l'apertura.
- `title`: Specifica il titolo dell'ancora

Contenitori

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

Il tag `<div>` (Content Division) è utilizzato per suddividere la pagina in blocchi o sezioni.

I tag contenitore, sono molto frequentemente caratterizzati da attributi `id` e `class` per identificarne lo scopo ed il ruolo.

Le liste

``

` ... `

` ... `

``

``

` ... `

` ... `

``

I tag lista possono essere di 2 tipi:

- ordinate ``
- non ordinate ``

La differenza è che di base, gli elementi di una lista ordinata sono preceduti da un numero progressivo, mentre gli elementi della lista non ordinata, da un bullet (•) chiamato "disc".

Immagini

```

```

Il tag `` ha 2 attributi principali:

- `src`: Specifica il percorso dell'immagine sul server o sul web
- `alt`: Specifica un titolo da utilizzare al posto dell'immagine (didascalia)

Il tag `` inoltre ammette altri attributi che ne modificano la visualizzazione come la larghezza (`width`) e l'altezza (`height`).

Altri tag

`
`

Il tag `
` "rompe" la riga di testo e manda a capo la visualizzazione.

`<p>` Questa è una riga interrotta dall'andata ``
a capo`</p>`

`<hr />`

Il tag `<hr />` visualizza una riga di separazione

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

`<p>` Paragrafo del capitolo `</p>`

``

`<p>` Paragrafo del capitolo preceduto da una riga di
separazione`</p>`

Tag in linea

<body>

</body>

Si chiamano tag in linea perchè a differenza di altri non “rompono” la riga.

Tutti i tag inline vanno racchiusi all’interno di altri tag di blocco o liste per definire il comportamento di alcune porzioni di blocchi

Tag in linea

`<p>Questa riga contiene una parte in grassetto</p>`
Questa riga contiene una parte **in grassetto**

`<p>Questa riga contiene una parte in corsivo</p>`
Questa riga contiene una parte *in corsivo*

`<p>Questa riga contiene una parte speciale</p>`
Questa riga contiene una parte speciale

Attributi dei Tag

Tutti i Tag, in genere, ammettono l'uso di alcuni attributi di qualificazione globali detti anche Core Attributes (**coreattrs**) fra cui: `id`, `class`, `style`.

Identificatore	Nome	Specifiche
<code>id</code>	Identificatore	Specifica l'identificatore di un elemento (deve essere unico nel documento)
<code>class</code>	Classe	Specifica la classe (o insieme) di appartenenza
<code>style</code>	Stile	Permette di specificare lo stile dell'elemento "in linea"

Attributi dei Tag

Alcuni esempi

```
<h1 id="titolo"> Io sono il titolo </h1>
```

```
<li class="elemento"> Io sono un punto di un elenco </li>
```

```
<a href="..." style="color:red;"> Io sono un link rosso </a>
```


Tag <table>

L'elemento probabilmente più utilizzato per visualizzare i dati, è il tag <table> che definisce una tabella.

I tag <tr> e <td> definiscono rispettivamente le righe e le colonne della tabella.

Le <table> sono costruite per righe:

```
<tr>  
  <td>Colonna 1</td>  
  <td>Colonna 2</td>  
  <td>Colonna 3</td>  
</tr>
```

Tag <thead>

Il tag <thead> definisce la testata di una tabella. Può contenere più righe di intestazione che si definiscono con il solito <tr> ma per le colonne dell'intestazione è bene usare il tag <th>

```
<table>
  <thead>
    <tr>
      <th>Titolo Colonna 1</th>
      <th>Titolo Colonna 2</th>
      <th>Titolo Colonna 3</th>
    </tr>
  </thead>
</table>
```

Tag <tbody>

Il tag <tbody> definisce il corpo della tabella. Può contenere più righe che si definiscono con i tag <tr>.

Ogni riga a sua volta può contenere più colonne definite con il tag <td>.

```
<table>
  <thead> ... </thead>
  <tbody>
    <tr>
      <td>Colonna 1</td>
      <td>Colonna 2</td>
      <td>Colonna 3</td>
    </tr>
  </tbody>
</table>
```

Tag <tfoot>

Il tag <tfoot> definisce il footer della tabella. Può contenere più righe che si definiscono con i tag <tr> che a sua volta può contenere più colonne definite con il tag <td>.

```
<table>
  <thead> ... </thead>
  <tbody> ... </tbody>
  <tfoot>
    <tr>
      <td>Colonna 1</td>
      <td>Colonna 2</td>
      <td>Colonna 3</td>
    </tr>
  </tfoot>
</table>
```

Tag `<form>`

Il tag `<form>` viene utilizzato per costruire un modulo di raccolta dati che vengono inviate al server per poi essere elaborate, ad esempio per essere salvate su un database.

Attraverso i `form`, il browser:

- Raccoglie i dati dall'utente
- Crea una connessione HTTP con il server all'URL specificato nell'attributo `action` e invia i dati al server

Tag <form>

<form action="form.php" method="POST">

Il tag <form> possiede 2 attributi principali ed indispensabili:

- `action`: l'indirizzo della pagina a cui inviare i dati
- `method`: Il metodo (o verbo) HTTP da utilizzare

Tag <form>

Il tag <form> racchiude un gruppo di altri tag che servono a raccogliere le informazioni:

<input>

<textarea>

<select> + <option>

<button>

HTML5 inoltre introduce nuovi tag:

<datalist> + <option>

<fieldset>

<label>

Tag <input>

<input type="..." />

- text
- password
- radio
- checkbox
- submit

HTML5 inoltre introduce nuovi input:

- date
- color
- email
- number

Costruzione di un Form

```
<form action="form.php" method="POST">
  Nome:<br/>
  <input type="text" name="nome">
  <br/>
  Cognome:<br/>
  <input type="text" name="cognome">
  <br>
  <input type="submit" value="Invia">
</form>
```

Il form viene quindi costruito racchiudendo tutti i campi `<input>` nel tag `<form>` avendo cura di aggiungere:

- gli attributi `action` e `method` al tag `<form>`
- di includere un tag `<input>` di tipo `submit`

Tag <form>

Le informazioni riguardanti i form sono davvero tantissime, in questo corso preferisco affrontarle in maniera più pratica durante la realizzazione del progetto.

Potete comunque trovare tutte le informazioni aggiornate sulla pagina http://www.w3schools.com/html/html_forms.asp

Fogli di stile

CSS (**C**ascading **S**tyle **S**heet) è un linguaggio che definisce lo stile di un documento HTML.

Attraverso i fogli di stile è possibile definire l'aspetto dei vari elementi del codice HTML ad esempio:

- Colore, Font, interlinea di un titolo o di un paragrafo
- Altezza, larghezza, bordo di un contenitore o di una sezione
- Allineamento, bulleting, spaziatura di un elenco puntato

Maggiori informazioni su: <http://www.w3schools.com/css/>

Regole CSS

Le regole CSS hanno la seguente forma:

h1 { **font-size:** **25px;** }

↑ ↑ ↑

Selettore **Proprietà** **Valore**

Selettori CSS

Le regole CSS vengono definite con dei selettori:

- Nome tag: `h1...h6, p, a, form, li, ol, ul ...`
- Classi: `.nome-classe`
- Identificatori: `#nome-id`

Fogli di stile

Il codice CSS può essere inserito all'interno del documento HTML in diversi modi, noi vedremo solamente quello che prevede di organizzare il codice in file css ed inclusi attraverso il tag `<link>`

CSS può essere inserito all'interno del documento HTML in diversi modi, noi vedremo solamente quello che prevede di organizzare il codice in file css ed inclusi attraverso il tag `<link>`

```
<link rel="stylesheet" href="nome-file.css">
```

Negli esempi, io utilizzerò uno stile che non prevede l'uso di classi o attributi disponibile al link:

```
https://watercss.kognise.dev/
```

Per il vostro progetto potete scegliere una delle alternative elencate qui:

```
https://css-tricks.com/no-class-css-frameworks/
```

Esercizio 1

Utilizzando il file di esempio HTML (esempio_1.html) create due documenti in cui inserite diversi esempi di tag HTML.

Aggiungete i link con il tag `<a>` per collegare i due documenti l'uno all'altro.

Aggiungete infine il foglio di stile scegliendo dall'elenco indicato.

php

Personal **H**ome **P**age Tool

PHP **H**ypertext **P**rocessor

<http://www.php.net>

1994 Rasmus Lerdorf

Il linguaggio PHP

Sintassi simile al linguaggio **C**

Largamente utilizzato (Wordpress, Joomla, Magento ...)

Ottime prestazioni (specialmente dalla versione 7.0)

Il linguaggio PHP

Linguaggio interpretato

Linguaggio Object Oriented (dalla versione 3.0)

Linguaggio a tipizzazione dinamica (dynamic typing)

Il linguaggio PHP

Inserire il codice PHP in una pagina HTML è molto semplice:

- Nominare il file come .php e non .html
- Inserire il codice PHP fra i tag `<?php` e `?>`

Ad esempio:

```
<h1><?php echo "Titolo"; ?></h1>
```

Il linguaggio PHP

Altri semplici esempi:

```
<?php $titolo = "Titolo"; ?>
```

```
<h1> <?php echo $titolo; ?> </h1>
```

```
<?php $prezzo = 5.00 + (5.00*0.22); ?>
```

```
<p> <?php echo $prezzo . "€"; ?> </p>
```

array

```
<?php
```

```
$cars = array( "Volvo", "BMW", "Toyota" );
```

```
$cars = [ "Volvo", "BMW", "Toyota" ];
```

```
?>
```


Cicli

```
<?php  
  
$cars = array( "Volvo", "BMW", "Toyota" );  
  
foreach ( $cars as $index => $car ) {  
    echo $car;  
    echo $cars[ $index ];  
}  
  
?>
```

Alcune funzioni utili

```
// numero di elementi di un arr.  
count($arr);
```

```
// aggiunge $el ad $arr.  
array_push($arr, $el);
```

```
// divide $str in parti separate da $sep  
explode($sep, $str);
```

Selezione

```
<?php
```

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
if (count($cars) > 10) {
```

```
    echo "Ci sono più di 10 elementi";
```

```
} else {
```

```
    echo "Ci sono al massimo 10 elementi";
```

```
}
```

```
?>
```

Altre funzioni utili

// Controlli

isset(\$var);

empty(\$var);

is_null(\$var);

// File

fopen('...');

fclose(\$link);

fgetcsv(\$link, \$lim, ';');

Esercizio 2

Costruite un array bidimensionale e attraverso un ciclo (quello volete) mostrate il contenuto in una tabella HTML `<table>`

Fate lo stesso esercizio con i restanti cicli in modo da avere un esempio per tutti i loop: `for`, `foreach` e `while`.

Esempio 1

Invio dati da modulo e ricezione

Esercitazione

Vi ho assegnato una esercitazione da svolgere in laboratorio, la consegna è disponibile su Classroom