

Università degli Studi della Campania

Luigi Vanvitelli - Dipartimento di Ingegneria

Laurea Triennale in Ingegneria Elettronica e Informatica

Laboratorio di Sviluppo di Applicazioni per IoT
a.a. 2023-2024

Introduzione alle Applicazioni Web

Docente: Carlo Mazzocca
e-mail: carlo.mazzocca@unibo.it

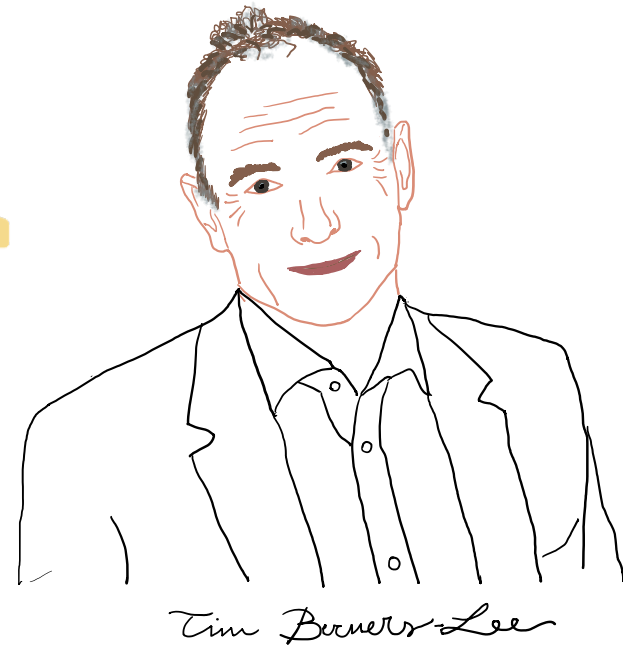
Storia del Web

Il Word Wide Web (WWW) è stato proposto nel 1989 da Tim Berners-Lee

L'idea alla base è di fornire strumenti adatti a **condividere documenti** statici:

- in **forma ipertestuale**
- **su rete Internet** attraverso un protocollo semplice

L'obiettivo era di sostituire i sistemi di condivisione basati su protocolli più vecchi come FTP

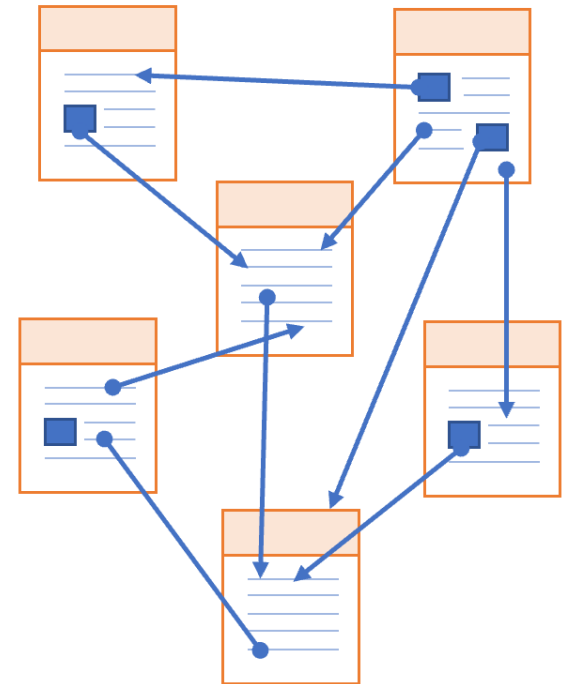


Iper testi

Un ipertesto è un insieme di documenti messi in relazione tramite collegamenti monodirezionali (link)

I link consentono di passare da un punto di un documento ad un altro qualsiasi dei documenti dell'ipertesto

Qualsiasi documento della rete può essere il successivo



Elementi Chiave

Tre elementi concettuali:

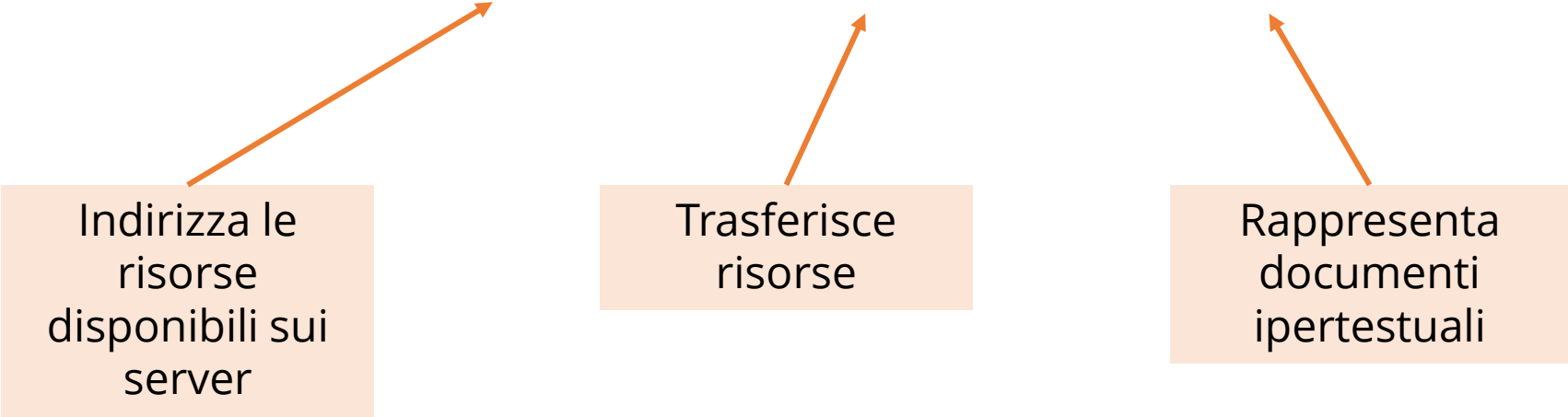
1. Un meccanismo per **localizzare un documento**
2. Un protocollo per **accedere alle risorse**
3. Un linguaggio per **descrivere i documenti**

Due elementi “fisici”:

1. Un **server** per erogare le risorse che costituiscono i documenti
2. Un **client** per visualizzare i documenti e di consentire la navigazione

Formula del Web

WWW = URL + HTTP + HTML



Indirizza le
risorse
disponibili sui
server

Trasferisce
risorse

Rappresenta
documenti
ipertestuali

Modello Client/Server

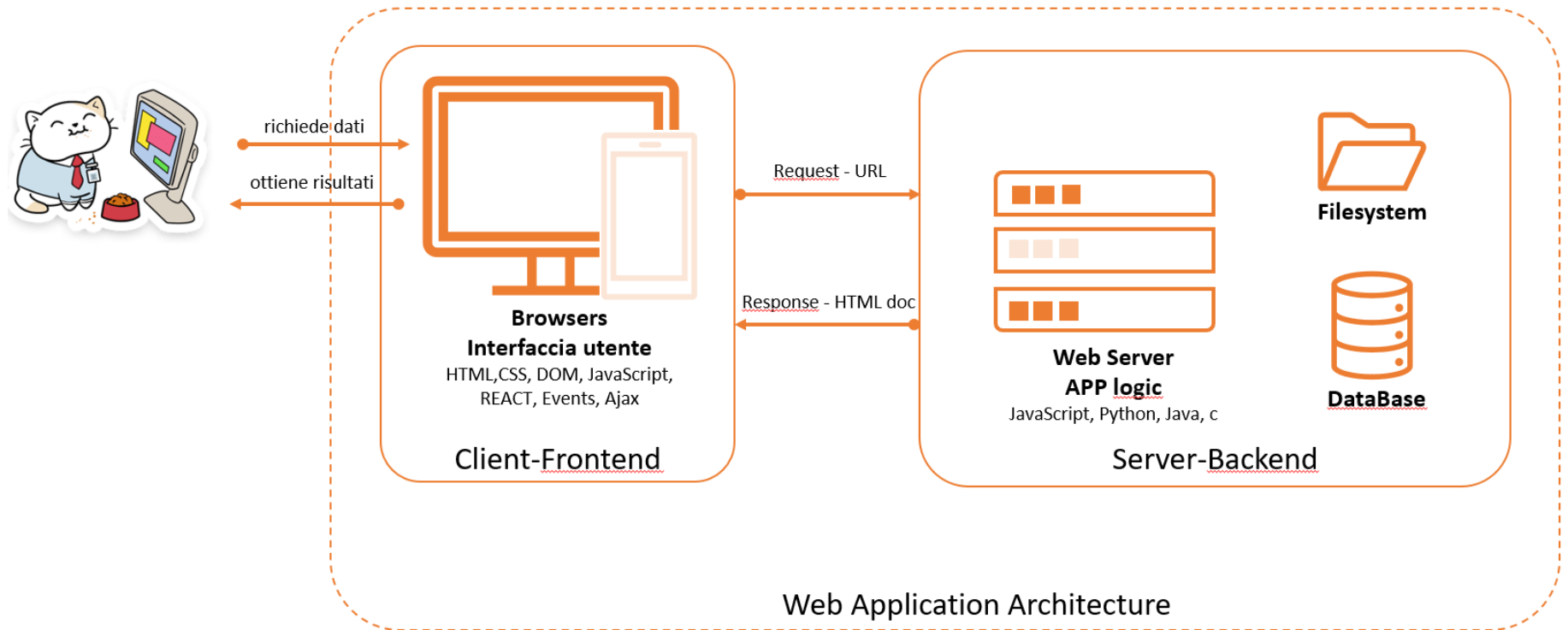
CLIENT ATTIVI

1. Utilizzano protocollo HTTP per connettersi ai server
2. Usano URL per identificare risorse
3. Richiedono pagine Web

SERVER PASSIVI

1. Rimangono in ascolto di eventuali connessioni
2. Utilizzano il protocollo HTTP per interagire con i client
3. Forniscono ai client le pagine Web richieste

Modello del Web



Uniform Resource Identifier

Gli Uniform Resource Identifier (URI) forniscono un meccanismo semplice ed estensibile per **identificare una risorsa**

Risorsa è una qualunque entità abbia una identità

Caratteristiche di un URI:

- Non fa necessariamente riferimento a risorse disponibili in rete
- Non si riferisce ad una versione dell'entità esistente in un dato momento

Uniform Resource Locator

Un Uniform Resource Locator (URL) tiene conto anche della **modalità per accedere alla risorsa**

Il nome dello schema corrisponde al protocollo utilizzato. Ad esempio in uno schema HTTP-like:

```
<protocol>:// [<username>:<password>@]<host>[:<port>] [/<path>[?<query>][#fragment]]
```

Questa forma vale per diversi protocolli di uso comune: HTTP, HTTPS, FTP, WAP, ... ma non, ad esempio, per la posta elettronica

Schema Indirizzo HTTP

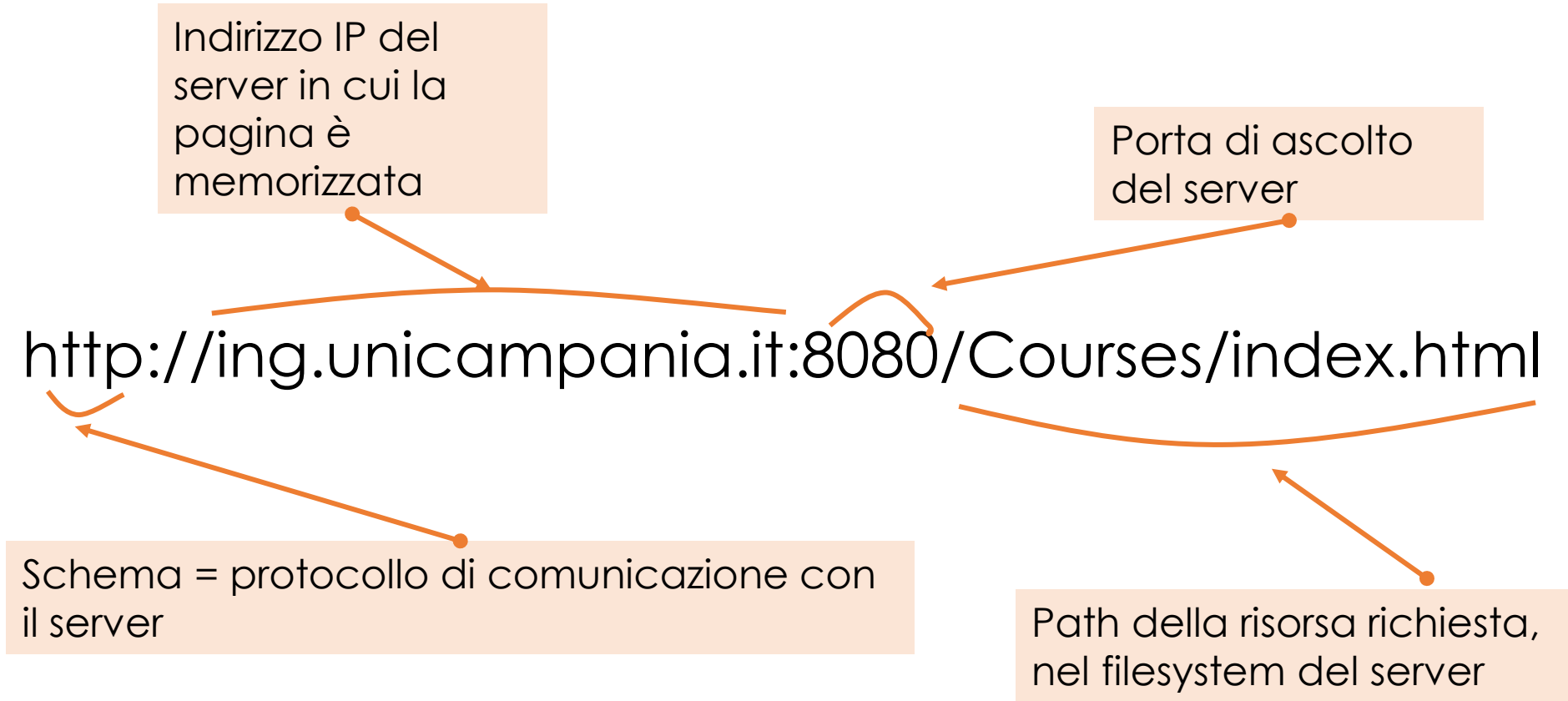
Indirizzo IP del server in cui la pagina è memorizzata

Porta di ascolto del server

http://ing.unicampania.it:8080/Courses/index.html

Schema = protocollo di comunicazione con il server

Path della risorsa richiesta, nel filesystem del server



Protocollo HTTP

- HTTP è il protocollo di **livello applicativo** utilizzato per **trasferire le risorse Web** da server a client
- Gestisce sia le richieste inviate al server che le risposte inviate al client
- È un protocollo **stateless** e basato su TCP

Connessione HTTP

Server rimane
in ascolto

Client apre
una
connessione
TCP sulla
porta 80

Server
accetta la
connessione

Server invia
la risposta e
chiude la
connessione
.

Client
manda una
richiesta



HTTP (1.0)

CLIENT

1. Il client HTTP inizia una connessione TCP verso il server http sull'host sulla porta 80

3. Il client http invia un messaggio di richiesta http contenente URL

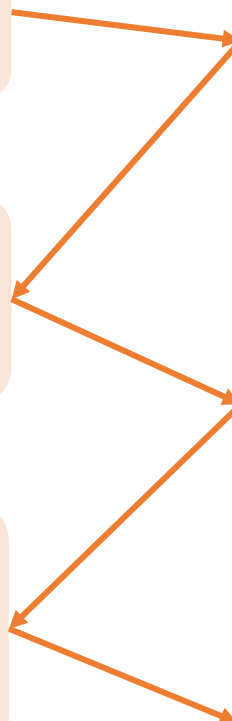
5. Il client http riceve il messaggio di risposta contenente il file HTML, visualizza la pagina HTML

SERVER

2. Il server HTTP è "in ascolto" sulla porta 80. "Accetta" la richiesta di connessione e ne dà conferma al client

4. Il server HTTP riceve il messaggio di richiesta, costruisce un messaggio di risposta contenente l'oggetto richiesto e lo invia

6. Il server HTTP chiude la connessione TCP



Richiesta HTTP

Request line
contiene i
comandi (GET,
POST, ...)
l'oggetto e la
versione del
protocollo

Header lines

NB: Body vuoto

```
GET /somedir/page.html HTTP/1.1
Host: www.unicampania.it
Connection: close
User-agent: Chrome/37.0
Accept: text/html, image/gif,image/jpeg
Accept-language:fr C
```

Chiudi la
connessione al
termine della
richiesta

GET – Comandi di Richiesta

- Utilizzato per richiedere una risorsa ad un server
- Metodo più frequentemente utilizzato
- Richiede il passaggio di parametri (URL)

POST – Comandi di Richiesta

I dettagli per l'identificazione ed elaborazione delle risorse sono contenuti nel body del messaggio

Comporta una trasmissione di informazioni dal client al server senza la creazione di una nuova risorsa

PUT – Comandi di Richiesta

- Comporta la creazione di una nuova risorsa
- L'argomento è la risorsa che si vuole ottenere facendo una GET con lo stesso nome
- Generalmente disabilitato sui server pubblici

DELETE – Comandi di Richiesta

- Richiede la cancellazione della risorsa riferita da un URL
- Generalmente disabilitato sui server pubblici

HEAD, OPTION, TRACE

- **HEAD:** Ottenere gli header senza body e verificare la validità dell'URL
- **OPTIONS:** Richiedere le informazioni sulle opzioni disponibili per la comunicazione
- **TRACE:** Vedere cosa è stato ricevuto dal server, utilizzato nella diagnostica e nel testing dei servizi Web

Risposta HTTP

Status line:

protocollo, codice di stato, status phrase

Header lines

Dati di risposta ad esempio pagina HTML richiesta

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 2008 12:00:15 GMT
Server: Apache/2.3.0 (Unix)
Last-Modified: Mon, 22 Jun 2008 ...
Content-Length: 6821
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.51 Transitional//EN">
<html>...</html>
```

Codice di Stato

Un numero di tre cifre, di cui la prima indica la classe della risposta e le altre due la risposta specifica

- **1xx: Informational.** Una risposta temporanea alla richiesta, durante il suo svolgimento
- **2xx: Successful.** Il server ha ricevuto, capito e accettato la richiesta
- **3xx: Redirection.** Il server ha ricevuto e capito la richiesta, ma sono necessarie altre azioni del client
- **4xx: Client error.** La richiesta del client non può essere soddisfatta per un errore da parte del client
- **5xx: Server error.** Il server non è in grado di soddisfare la richiesta

Esempi Significativi

- **200 Ok**, GET con successo
- **201 Created**, PUT con successo
- **301 Moved permanently**, URL non valida, il server conosce la nuova posizione
- **400 Bad request**, errore sintattico nella richiesta
- **403 Forbidden**, richiesta non autorizzabile
- **404 Not found**, URL errato.
- **500 Internal server error**, tipicamente un CGI mal fatto.

Cookie

I cookie sono struttura dati che si muove come un token, dal client al server e viceversa

- Possono essere generati sia dal client che dal server
- Vengono sempre passati ad ogni trasmissione di request e response
- Forniscono un **supporto per il mantenimento di stato**

HTML

WWW = URL + HTTP + HTML

- HTML il linguaggio utilizzato per **descrivere le pagine** che costituiscono i nodi dell'ipertesto
- È un linguaggio di **codifica del testo del tipo a marcatori** (markup)
- Consente di **rappresentare un documento in modo che sia trattabile dall'elaboratore in quanto testo**

Tag HTML

I **tag** sono usati per definire il mark-up di elementi HTML

- Sono preceduti e seguiti da due caratteri in parentesi angolari
- Sono normalmente accoppiati; un esempio è dato da **<p>** e **</p>**, detti rispettivamente start tag ed end tag
- Il testo tra start tag ed end tag è detto contenuto dell'elemento

`<p>`

Start tag

Testo di un paragrafo

contenuto

`</p>`

end tag

Attributi

Un elemento può essere dettagliato mediante **attributi**

- Gli attributi sono coppie “nome = valore” contenute nello start tag con una sintassi di questo tipo:

```
<tag attrib1='valore1' attrib2='valore2'>
```

I valori sono racchiusi da apici singoli o doppi. Esempio:

```
<input type='submit' value="NAME">Ok</input>
```

- Gli apici possono essere omessi se il valore non contiene spazi
- I valori «colori» sono espressi con un nome o in formato RGB

Tipi MIME

Lo standard Multipurpose Internet Mail Extensions (MIME) è nato originariamente per poter allegare data file ai messaggi di posta elettronica

- **Rappresenta il tipo di contenuto di un messaggio** (es. HTTP request)
- Classifica i tipi di contenuto sulla base di una logica a due livelli ed è largamente utilizzata in ambito web

Un tipo MIME è espresso con questa sintassi:

`tipo/sottotipo`

Esempi:

`text/plain: testo semplice`

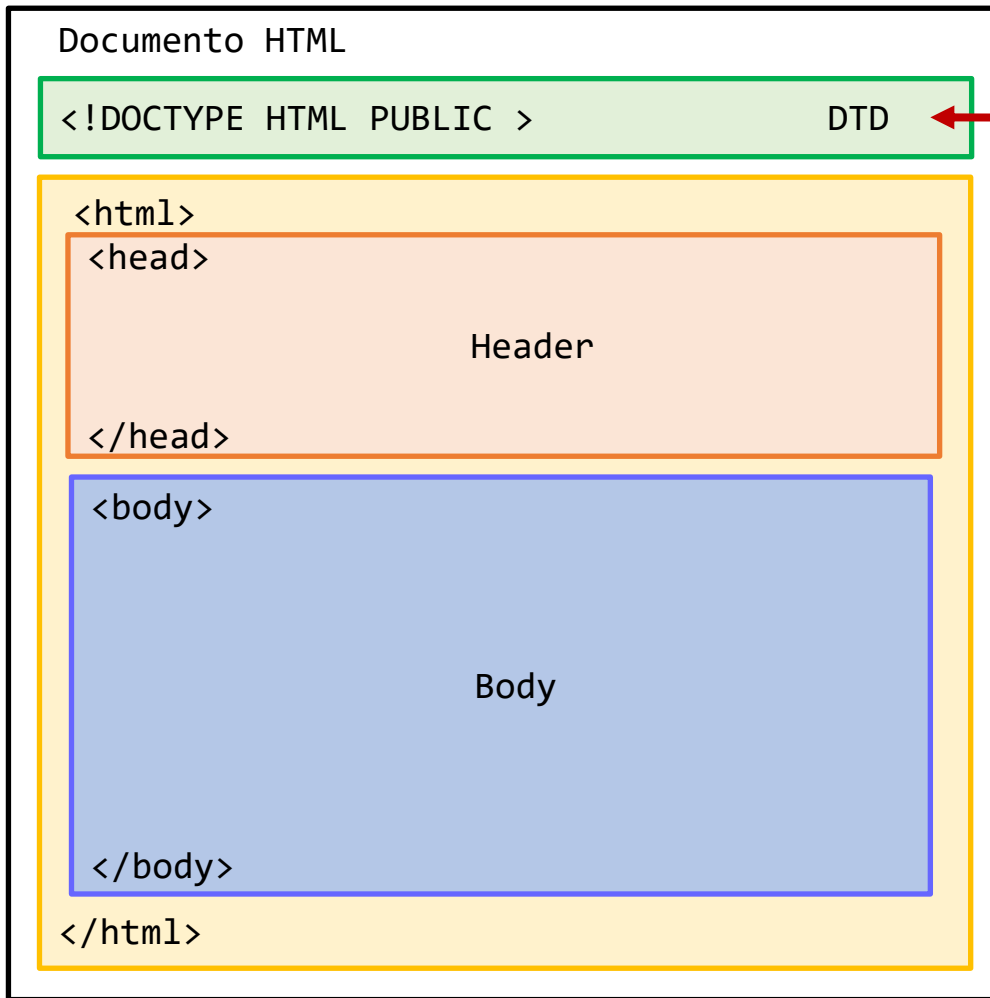
`text/html: testo HTML`

Commenti

È possibile inserire commenti in qualunque punto all'interno di una pagina HTML con la seguente sintassi:

```
<!-- Questo è un testo di commento -->
```

Struttura Documento HTML



DTD: Document Type Definition

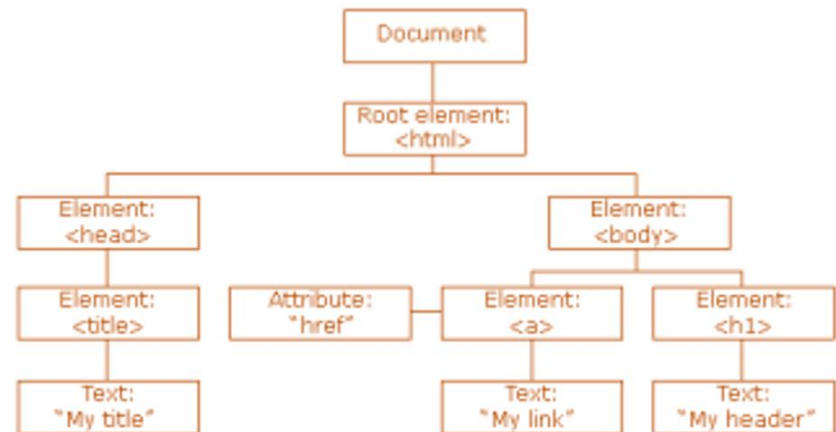
Serve al browser per identificare le regole di interpretazione e visualizzazione da applicare al documento

DOM

Una pagina HTML può essere rappresentata come una struttura ad albero chiamata **Document Object Model** (DOM)

Quando un browser riceve una pagina HTML, ne fa il parsing e costruisce la struttura ad albero del DOM

```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <a href="...">MyLink</a>
    <h1>My header</h1>
  </body>
</html>
```



Header

L'header è identificato dal tag `<head>`, contiene **elementi non visualizzati** dal browser (informazioni di servizio)

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="description" content="Documentation about HTML">
  <meta name="keywords" content="HTML, tags, commands">
  <title>Impariamo l'HTML</title>
  <link href="style.css" rel="stylesheet" type="text/css">
</head>
```

Body

Il tag `<body>` delimita il corpo del documento. Contiene la parte che viene mostrata dal browser

```
<body>
<h1>Titolo</h1>
<p>Questo è un paragrafo completo di un
documento.</p>
<p>Un altro paragrafo<br>con un a capo</p>
<hr>
<p>Esempio di lista puntata, la lista della
spesa:</p>
<ul>
  <li>Pane</li>
  <li>Latte</li>
  <li>Prosciutto</li>
  <li>Formaggio</li>
</ul>
</body>
```

Titolo

Questo è un paragrafo completo di un documento

Un altro paragrafo con un a capo

Esempio di lista puntata, la lista della spesa:

- Pane
- Latte
- Prosciutto
- Formaggio

Elementi del Body

- **Intestazioni:** titoli organizzati in gerarchia
- **Strutture di testo:** paragrafi, testo indentato, ecc
- **Aspetto del testo:** grassetto, corsivo, ecc
- **Elenchi e liste:** numerati, puntati
- **Tabelle**
- **Form** (moduli elettronici): campi di inserimento, checkbox, menu a tendina, bottoni, ecc
- **Collegamenti ipertestuali e ancore**
- **Immagini e contenuti multimediali** (audio, video, animazioni, ecc.)
- **Contenuti interattivi:** script, applicazioni esterne

Heading

I tag `<h1>`, `<h2>` ... `<h6>` servono per definire dei titoli di importanza decrescente (`<h1>` è il più importante)

- Sono previste 6 grandezze; i titoli appaiono in grassetto e lasciano una riga vuota prima e dopo di sé
- Ammettono attributi di allineamento:
`<h1 align = left | center | right | justify>`

```
<h1>Titolo 1 </h1>  
<h2>Titolo 2 </h2>  
<h3>Titolo 3 </h3>  
<h4>Titolo 4 </h4>  
<h5>Titolo 5 </h5>  
<h6>Titolo 6 </h6>
```

Titolo 1
Titolo 2
Titolo 3
Titolo 4
Titolo 5
Titolo 6

Paragrafi

Il **paragrafo** è l'unità di base entro cui suddividere un testo

- Il tag `<p>` lascia una riga vuota prima della sua apertura e dopo la sua chiusura
- Per andar a capo all'interno di un paragrafo si usa l'elemento `
`

Esempio: due paragrafi

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

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

Vengono visualizzati così:

paragrafo1

paragrafo2

Stili di Testo

Nella terminologia tipografica lo "stile di un testo" indica le possibili varianti di forma di un carattere: tondo (normale), neretto (grassetto), corsivo

HTML consente di definire lo stile di un frammento di testo, combinando fra loro anche più stili

<code><tt>...</tt></code>	Carattere monospaziato
<code><i>...</i></code>	Corsivo
<code>...</code>	Grassetto
<code><u>...</u></code>	Sottolineato (deprecato)
<code><s>...</s></code>	Testo barrato (deprecato)

Tabelle

Il tag `<table>` racchiude la tabella

- **align** = “{left | center | right}” allineamento della tabella rispetto alla pagina
- **width**=“n | n%” larghezza della tabella (anche in percentuale rispetto alla pagina)
- **bgcolor**=“#xxxxxx” colore di sfondo della tabella
- **border**=“n” spessore dei bordi della tabella (0 = tabella senza bordi);
- cellpadding, cellspacing.

Esempio di Tabella

```
<table border="1" >
  <caption align="top">
    <em>A test table with merged
    cells</em></caption>
  <tr>
    <th rowspan="2"></th>
    <th colspan="2">Average</th>
    <th rowspan="2">Red<br>eyes</th>
  </tr>
  <tr><th>height</th><th>weight</th></tr>
  <tr><th>Males</th><td>1.9</td><td>0.003</td>
  <td> 40%</td></tr>
  <tr><th>Females</th><td>1.7</td><td>0.002</t
  d><td>43%</td></tr>
</table>
```

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Link Iperestuali

Il link è il costrutto di base di un ipertesto

- È una connessione fra una risorsa Web ed un'altra. Un link è costituito da due estremi - detti ancore (anchor) - e da una direzione di percorrenza

Link = source anchor → destination anchor

L'ancora di origine è un elemento contenuto nella pagina di partenza, mentre l'ancora di destinazione è una qualsiasi risorsa web (immagine, video, ecc)

Ancore

Le ancore si esprimono utilizzando il tag `<a>`

- Quelle di origine sono caratterizzate da un attributo, denominato `href`, che contiene l'indirizzo di destinazione (è un URL)
- Quelle di destinazione sono invece caratterizzate dall'attributo `name`
- L'esempio più semplice di link è quello che collega due elementi all'interno di uno stesso documento

Link Interni

```
<p>
  <a href="#section1">Introduzione</a><br>
  <a href="#section2">Concetti di base</a><br>
  <a href="#section2.1">Definizione del
  problema</a><br>
...
</p>
<h2><a name="section1">Introduzione</a></h2>
...sezione 1...
<h2><a name="section2">Concetti di base</a></h2>
...sezione 2...
<h3><a name="section2.1">Definizione del
problema</a></h3>
...sezione 2.1...
```

Sommario

[Introduzione](#)

[Concetti di base](#)

[Definizione del problema](#)

...

Introduzione

...sezione 1...

Concetti di base

...sezione 2...

Definizione del problema

...sezione 2.1...

Link Esterni

Il caso più comune è quello di un link ad un altro documento (pagina HTML) o in generale ad un'altra risorsa (es. un'immagine)

In questo caso il primo link non specifica un'ancora e quindi si “salta” all'inizio del documento chapter2.html

```
<body>
...
<p>Per maggiori informazioni leggete il
<a href="chapter2.html">capitolo 2</a>.
Guardate anche questa
<a href="../images/forest.gif">mappa
della foresta incantata.</a></p>
...
</body>
```

```
...
Per maggiori informazioni leggete il
capitolo 2. Guardate anche questa
mappa della foresta incantata.
```

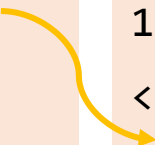
```
...
```

Link Completo

Il caso più completo è quello di un link ad un punto preciso di un documento (ancora di destinazione)

```
...  
<p>Per maggiori informazioni leggete  
il  
<a href="chapter2.html#section2 ">  
secondo paragrafo del capitolo  
2</a>.  
Guardate anche questa  
<a href=" ../images/forest.gif">  
mappa della foresta  
incantata.</a></p>  
...
```

```
...  
<h1>Capitolo 2</h1>  
<h2><a name="section1">Paragrafo  
1</a></h2>  
<p>Testo del primo paragrafo...</p>  
<h2><a name="section2">Paragrafo  
2</a></h2>  
<p>Testo del secondo  
paragrafo...</p>  
...
```



Form

Un **form** è una sezione di documento HTML che **contiene elementi di controllo** che l'utente può utilizzare **per inserire dati o in generale per interagire**

I dati inseriti possono essere poi inoltrati al server dove un agente può processarli

- Bottoni di azione
- Checkbox, caselle di spunta
- Radio Button bottoni mutuamente esclusivi
- Liste di selezione, lista di opzioni
- Caselle di inserimento di testo
- Oggetti nascosti (elementi valorizzati ma invisibili)

Tag <form>

Il tag **<form>** racchiude tutti gli elementi del modulo, è un elemento di tipo blocco

- **action = uri** **URI** dell'agente che riceverà i dati del form:
- **name = text** specifica il nome del form.
- **method = {get | post}** specifica il modo in cui i dati vengono inviati.
- **enctype = content-type**, se il metodo è POST specifica il content type usato per la codifica dei dati
- **default** application/x-www-form-urlencoded

```
<form action="http://site.com/bin/adduser" method="post">  
    ...form contents...  
</form>
```

Elementi di Input

La maggior parte dei controlli viene definita mediante il tag `<input>`. L'attributo **type** stabilisce il tipo di controllo:

- **text**: casella di testo monoriga
- **password**: come text ma il testo non è leggibile (****)
- **file**: controllo che consente di caricare un file
- **checkbox**: casella di spunta
- **radio**: radio button
- **submit**: bottone per trasmettere il contenuto del form
- **image**: bottone di submit sotto forma di immagine
- **reset**: bottone che riporta tutti i campi al valore iniziale
- **button**: bottone di azione
- **hidden**: campo nascosto

Input Text

Text è un campo per l'inserimento di testo su una sola riga

- **name = text** nome del controllo
- **value = text** eventuale valore iniziale
- **size = n** lunghezza del campo (numero di caratteri)
- **maxlength = n** massima lunghezza del testo (numero di caratteri)

```
<form action="http://site.com/bin/adduser"
method="post">
  <p>
    Nome: <input type="text" name="firstname">
  </p>
</form>
```

Nome:

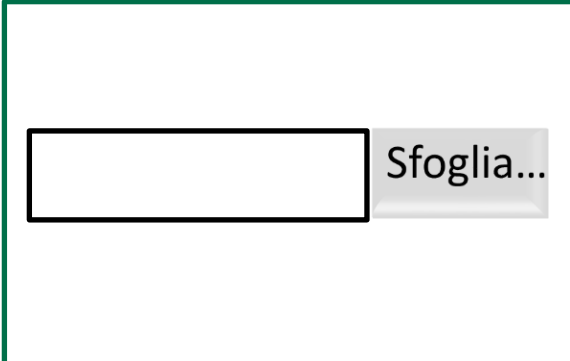
Input File

Consente di fare l'upload di un file selezionandolo nel filesystem del client.

- **name = text** specifica il nome del controllo;
- **value = content-type** lista di MIME types per l'upload;

Richiede una codifica particolare per il form (multipart/form-data) perché le informazioni trasmesse con il POST contengono tipologie di dati diverse

```
<form action="http://site.com/bin/adduser"
method="post"
enctype="multipart/form-data" >
  <p>
    <input type="file" name="attach">
  </p>
</form>
```



[illegible]

Liste di Opzioni

Il tag **<select>** permette di costruire liste di opzioni

- Per definire le singole opzioni si usa il tag **<option>** ricorrendo all'attributo **value** si può attribuire il valore.
- Con l'attributo **selected** si può indicare una scelta predefinita: **selected="selected"**

```
<form action="http://site.com/bin/adduser"
method="post">
<select name="provincia" >
<option value="BO"
selected="selected">Bologna</option>
<option value="MO">Modena</option>
<option value="PR">Parma</option>
<option value="PC">Rimini</option>
</select>
</form>
```



The image shows a visual representation of the HTML form. It consists of two identical dropdown menus side-by-side. The left menu has 'Bologna' selected. The right menu is open, showing a list of options: 'Bologna' (highlighted in green), 'Modena', 'Parma', and 'Rimini'.

CSS

Cascading Style Sheets (CSS) hanno lo scopo di separare contenuto e presentazione nelle pagine Web

Il linguaggio HTML serve a definire il contenuto senza indicare come rappresentarlo

I CSS servono a definire **come il contenuto deve essere presentato** all'utente

Vantaggi CSS

- Lo stesso contenuto diventa riusabile in più contesti
- Basta cambiare i CSS e può essere presentato correttamente in modo ottimale su dispositivi diversi
- Si può dividere il lavoro fra chi gestisce il contenuto e chi si occupa della parte grafica

CSS e HTML

```
<html>
```

```
<head>...</head>
```

```
<body>
```

```
<h1>title</h1>
```

```
<div>
```

```
<p> uno </p>
```

```
<p> due </p>
```

```
</div>
```

```
<p> tre
```

```
<a href="link.html">link</a>
```

```
</p>
```

```
</body>
```

```
</html>
```

Un documento HTML può essere visto come un insieme di blocchi (contenitori) sui quali si può agire con stili diversi. Ogni tag HTML definisce un blocco.

Sintassi CSS

- CSS si basa su regole
- Una regola è una dichiarazione su un aspetto [stilistico] di uno o più elementi HTML (Selettore + Dichiarazione/i)
- Un foglio di stile è un insieme di una o più regole che si applicano a un documento HTML

Sintassi CSS

Un'espressione come `H1 { color: blue }` prende il nome di regola CSS

Selettore: H1;

Dichiarazione: color: blue;

La sintassi generale si può quindi esprimere così:

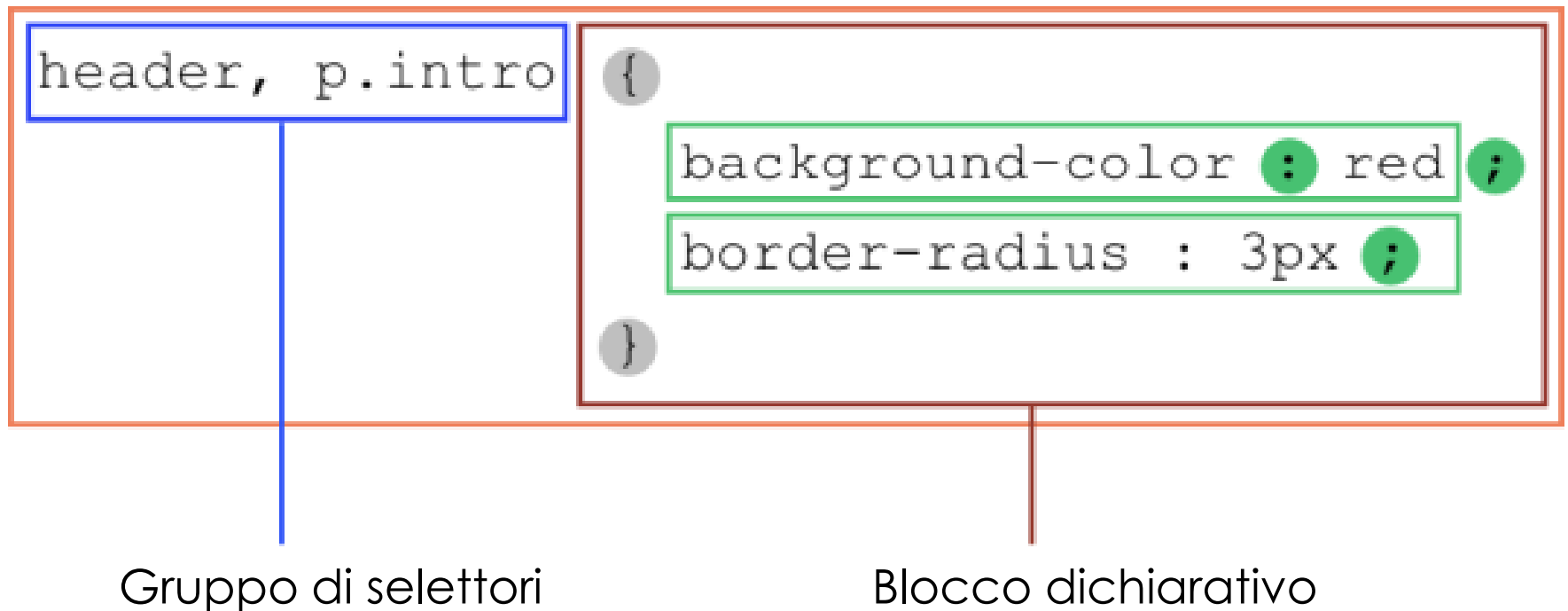
```
selettore { proprietà: valore }
```

O più in generale:

```
selettore { proprietà1 : valore1;
```

```
proprietà2 : valore2, valore3; }
```


Sintassi Generale



Hello World – No CSS

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Hello World</TITLE>
  </HEAD>
  <BODY>
    <H1>Hello World!</H1>
    <p>Usiamo i CSS</p>
  </BODY>
</HTML>
```

Hello World

Hello World!
Usiamo i CSS

Hello World – CSS

```
BODY { color: red }  
H1 { color: blue }
```

hello.css

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">  
<HTML>  
  <HEAD>  
    <TITLE>Hello World</TITLE>  
    <LINK rel="stylesheet" href="hello.css"  
      type="text/css">  
  </HEAD>  
  <BODY>  
    <H1>Hello World!</H1>  
    <p>Usiamo i CSS</p>  
  </BODY>  
</HTML>
```

Hello World!

Usiamo i CSS