

Week 7 – PC Web



Antonia Sacchitella

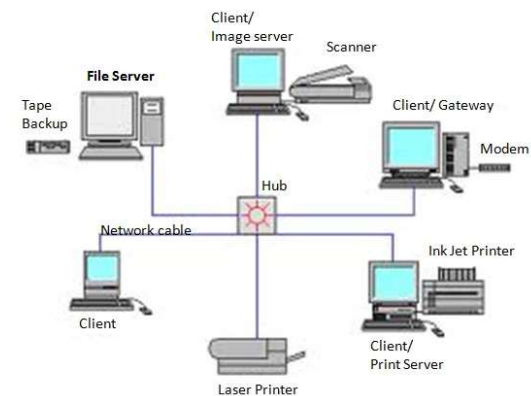
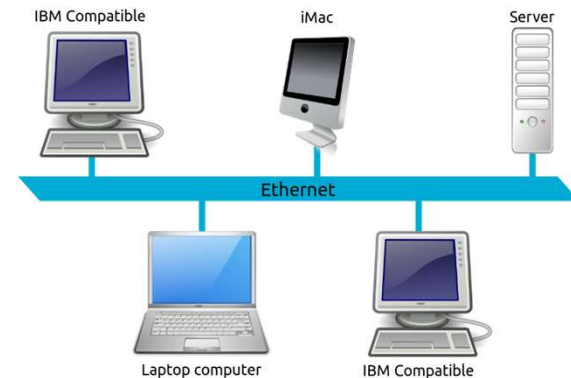
Analyst @icubedsrl

antonia.sacchitella@icubed.it



Reti di computer

- Infrastrutture che permettono di collegare computer a computer
- Cablate
- Il computer viene chiamato “Host”: ospitano l'applicazione per cui è stato necessario collegare il computer
- Trasmissione visibile a tutti gli host collegati

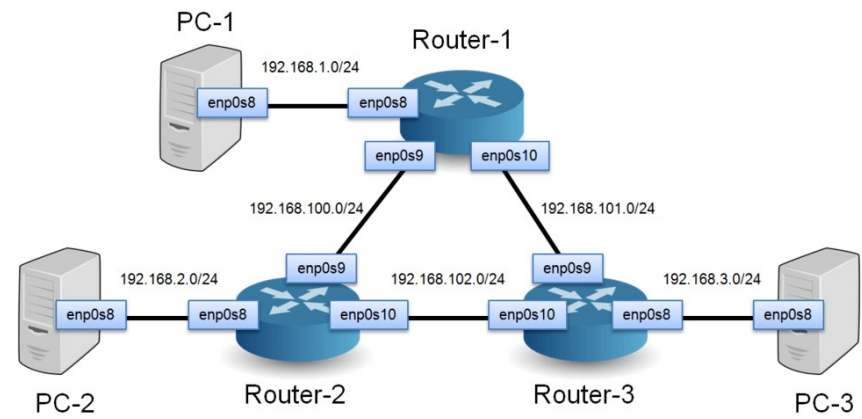


Reti di computer

Una rete di computer può essere collegata a un'altra tramite un Router

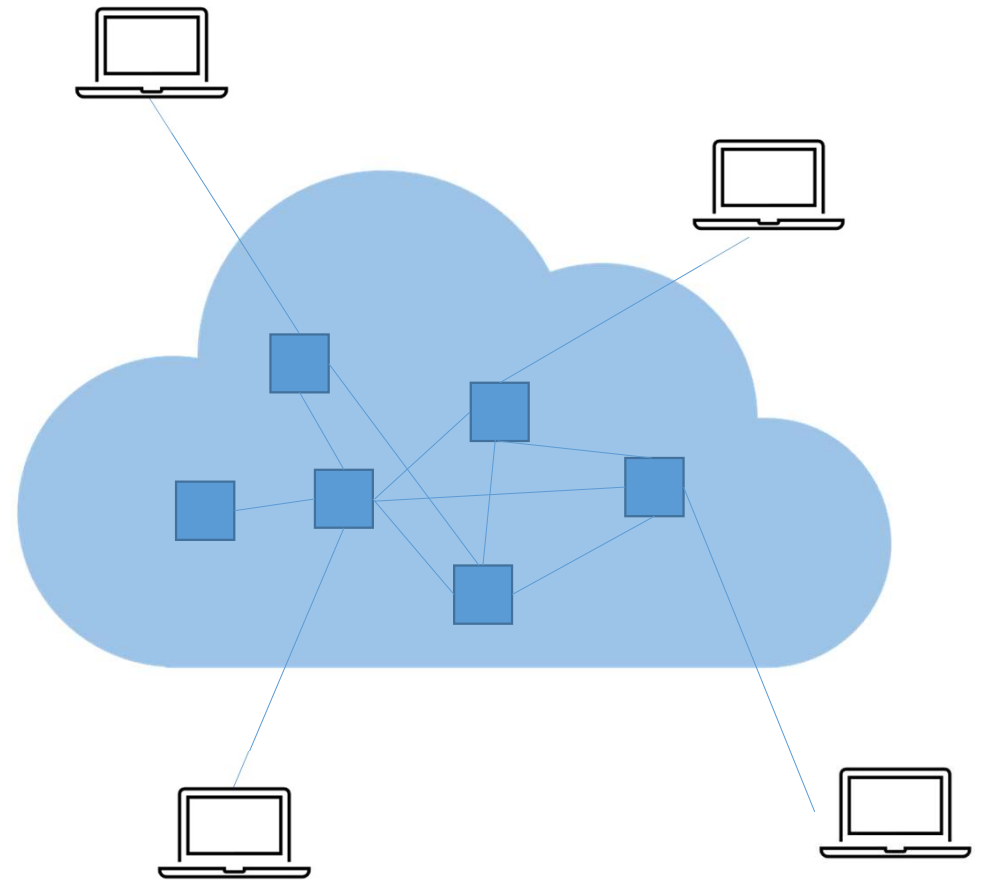
Router: instrada i messaggi tra le reti

- Sistema di comunicazione condiviso
- Conoscenza del grafo locale



Internet

- Unica rete virtuale
- Formata de numerosi Router
- Host all'esterno della rete

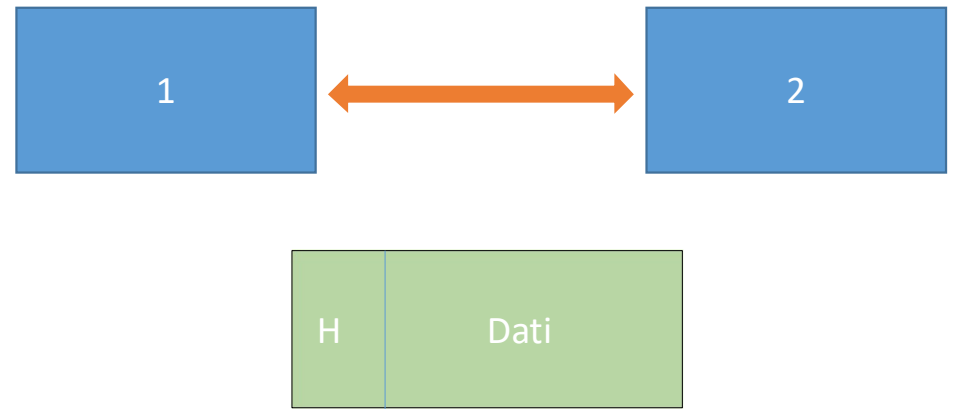


Protocolli di comunicazione

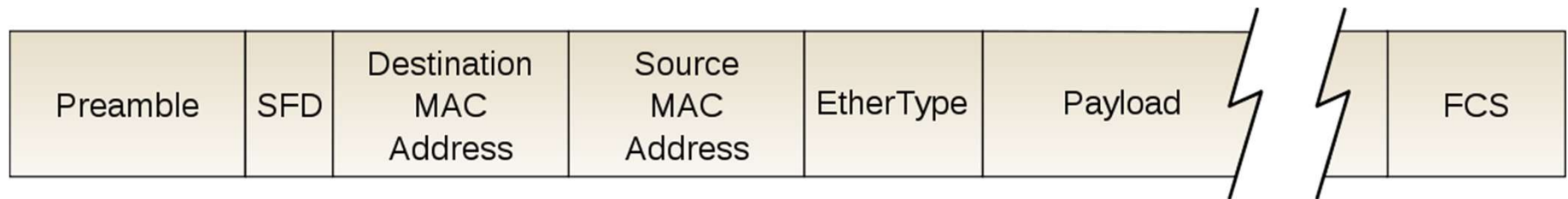
Un insieme di regole per lo scambio di messaggi tra due entità

Nel caso che le entità siano computer, il messaggio sarà diviso in:

- Header
- Dati

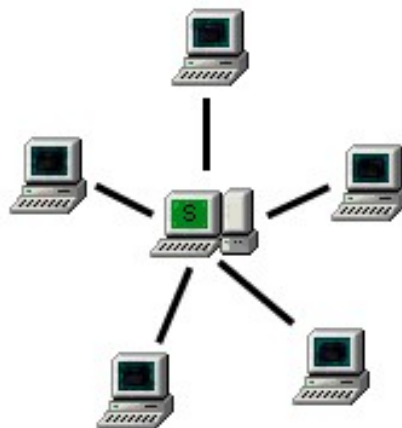


Es. Ethernet

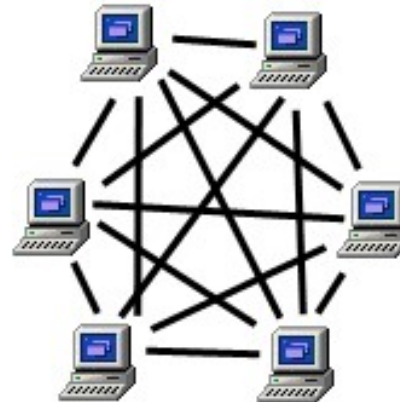


Tipologie Protocolli

Server Based Network



Peer to Peer Network



Tipi di Reti

Commutazione di circuito

- Connessione tra interlocutori è dedicata per tutta la durata della comunicazione
- Es: chiamata



Commutazione di pacchetto

- Non esiste un canale dedicato
- Il messaggio viene suddiviso in pacchetti che possono essere instradati dal router per percorsi diversi
- Il messaggio dovrà essere riassemblato



Trasmissione informazioni in Internet

Messaggio Finito

- Diviso in pacchetti
- Router li instrada su percorsi diversi

Es: Allegato

Flusso di dati

- Flusso di dati viene spezzato in pacchetti detti “datagram”
- I Datagram sono correlate dalle informazioni importanti per la trasmissione (indirizzo mittente e destinatario)

Qualità del Servizio

Ogni pacchetto ricevuto da Router viene accodato. Se la pila è piena, il pacchetto in eccesso deve essere ritrasmesso.

Ciò potrebbe comportare una qualità del servizio scarsa: il pacchetto potrebbe impiegarci troppo tempo oppure essere perso.

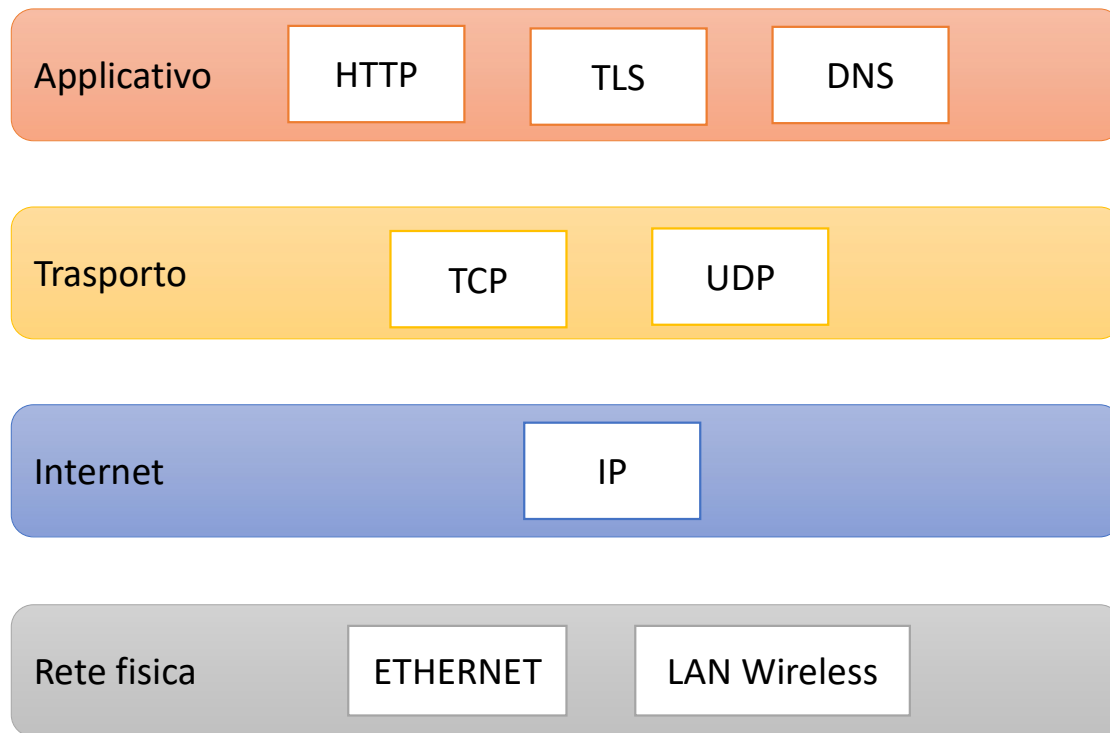
I pacchetti quindi vengono etichettati con la tipologia di trasmissione, il router decide se promuovere dei pacchetti o scartarli tramite un processo chiamato **deep packet inspection**.

Demo

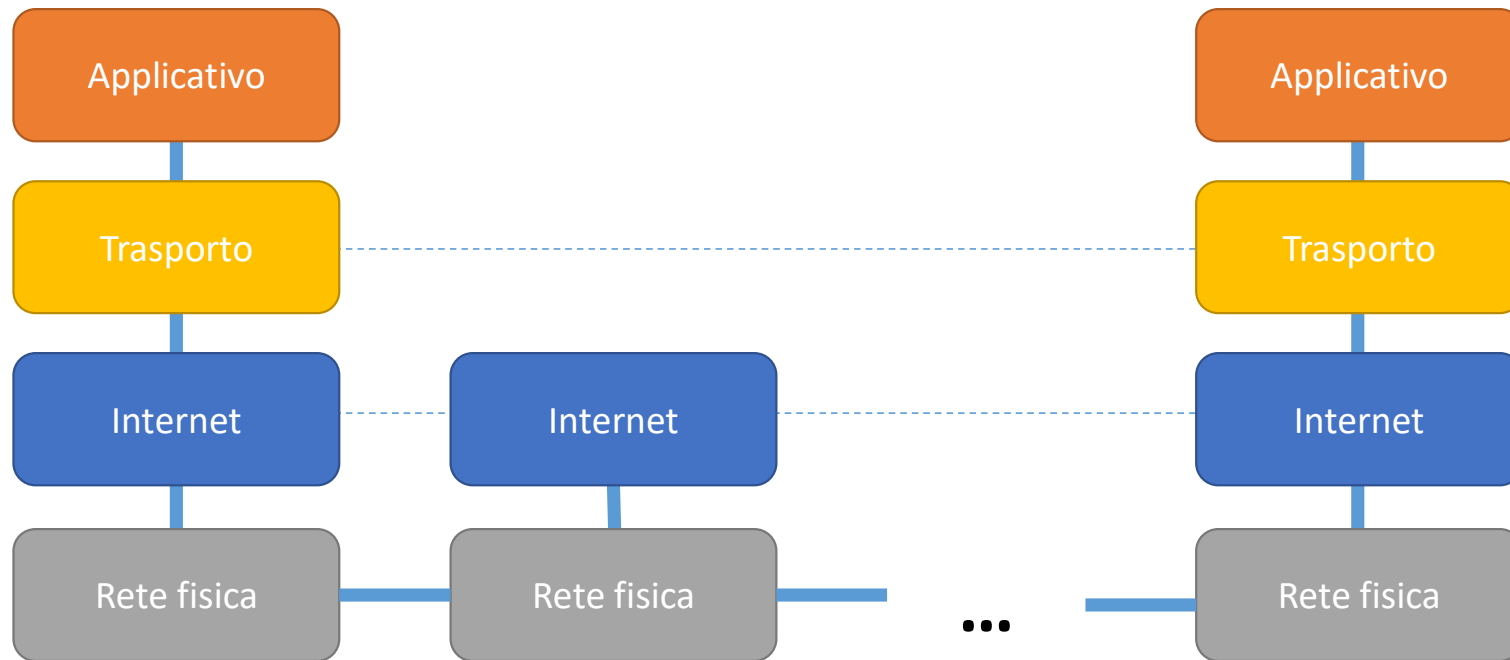
Ping e tracert



Gerarchia Protocolli



Protocolli e Trasmissione



Internet Protocol

IP: sequenza che identifica univocamente i dispositivi connessi alla rete

Assegnazione dell'IP può essere:

- Statica
- Dinamica

IPv4

32 bit: 4 interi tra 0-255

~ 4 miliardi di combinazioni

Es: 192.128.1.89

IPv6

128 bit: 8 interi fra 0 e 2^{16}

$\sim 2^{128}$

Es: ac32:11tr:0000:0002:78qe:1234: ...

Principi di Internet

- **Protocolli gerarchici**

Funzionalità di trasporto e instradamento sono separate e indipendenti dall'applicazione

- **Principio End-to-End**

Funzionalità applicative svolte ai nodi della rete, non al suo interno

- **Principio Best-Effort Delivery**

La rete non fa differenziazione dei pacchetti, quale sia il contenuto, mittente e destinatario

Domain Name System

Nella vita di tutti i giorni però non utilizziamo gli IP per accedere a un particolare servizio.

All'IP viene associato un nome simbolico.

The text "www.icubed.it" is displayed in a white, monospace-style font on a dark rectangular background.

La corrispondenza viene conservata da un Server : DNS server

Contiene localmente un certo numero di corrispondenze: si collega con altri DNS Server per ottenere l'ip corrispondente.

Domain Namespace

Livello 1: Top Level Domain

Gestito dall'organizzazione IANA

Definizione dei suffissi: .it, .com, .gov, .org ...

Livello 2: Second level Domain

Nome (Es. Microsoft)

www.support.microsoft.com

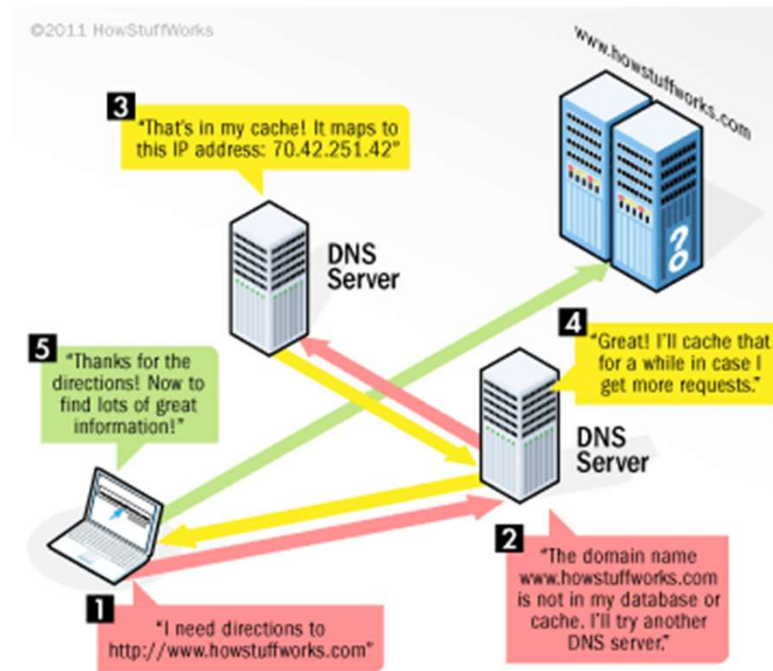
Livello 3: Host

Da qui è organizzato internamente



DNS

DNS è un Sistema composto da un database distribuito che mappa il nome simbolico all'IP



Ruoli in Internet

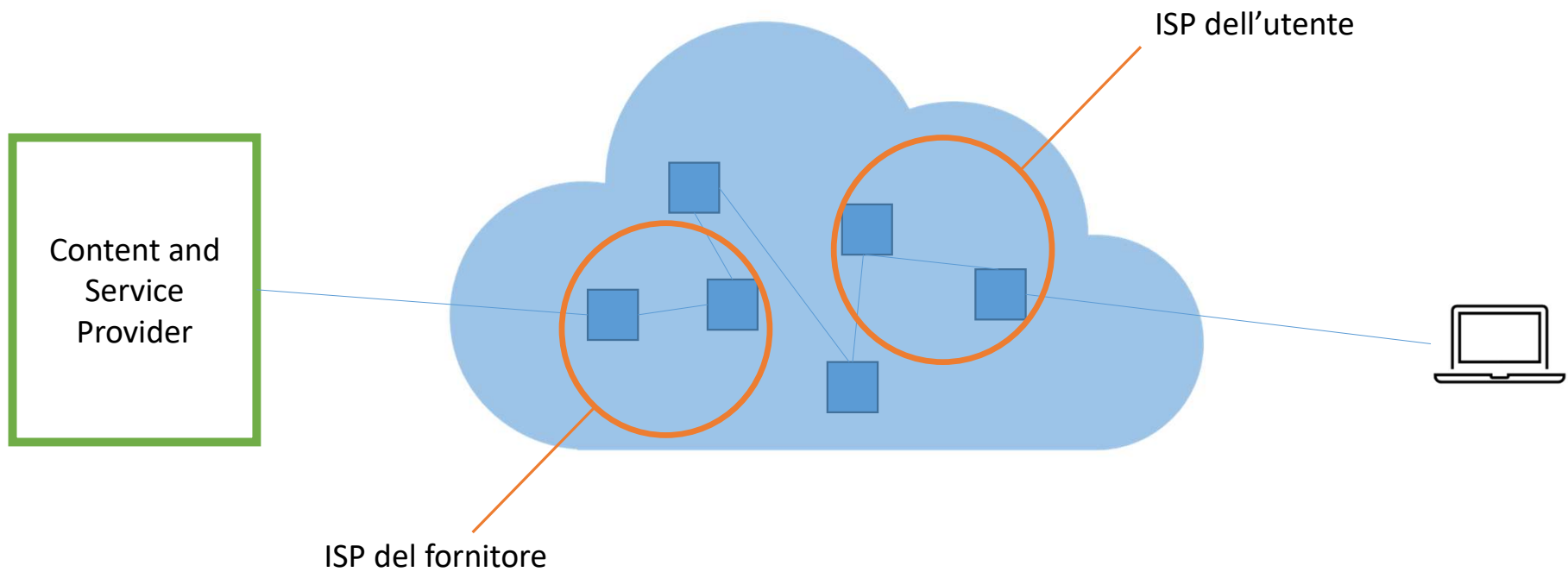
Technology Components Providers: sono i provider di dispositivi per internet. (es. dispositivi di rete)

Access Device Providers: sono i provider che producono i dispositivi di accesso ad internet.

Content and Service Providers: sono i provider che forniscono il contenuto e i servizi in Internet

Network Providers (ISP): Provider che forniscono la rete internet.

Internet Service Providers



Introduzione al Web



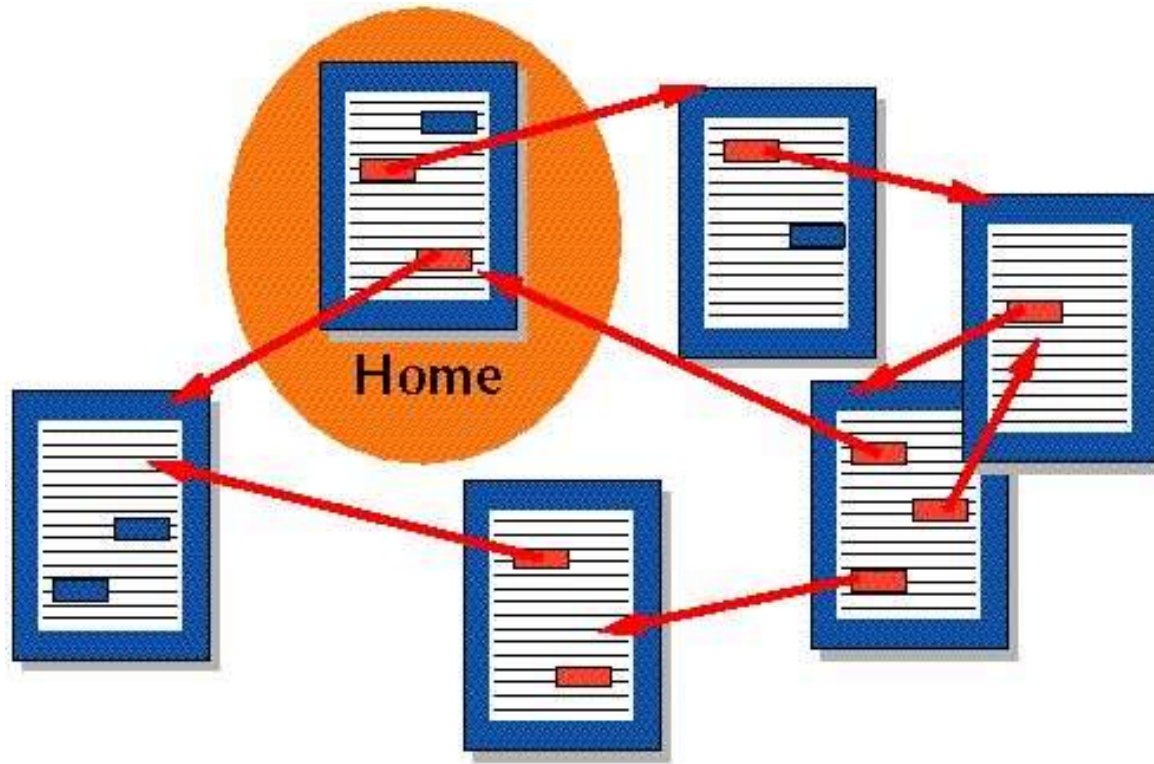
Antonia Sacchitella

Analyst @icubedsrl

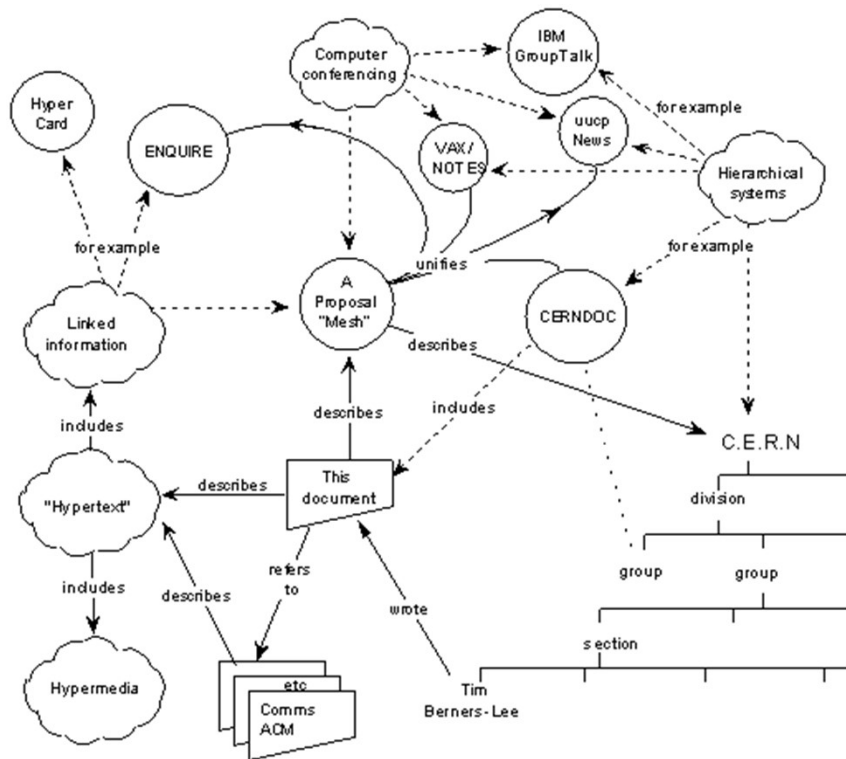
Antonia.sacchitella@icubed.it



Hypertext



World Wide Web

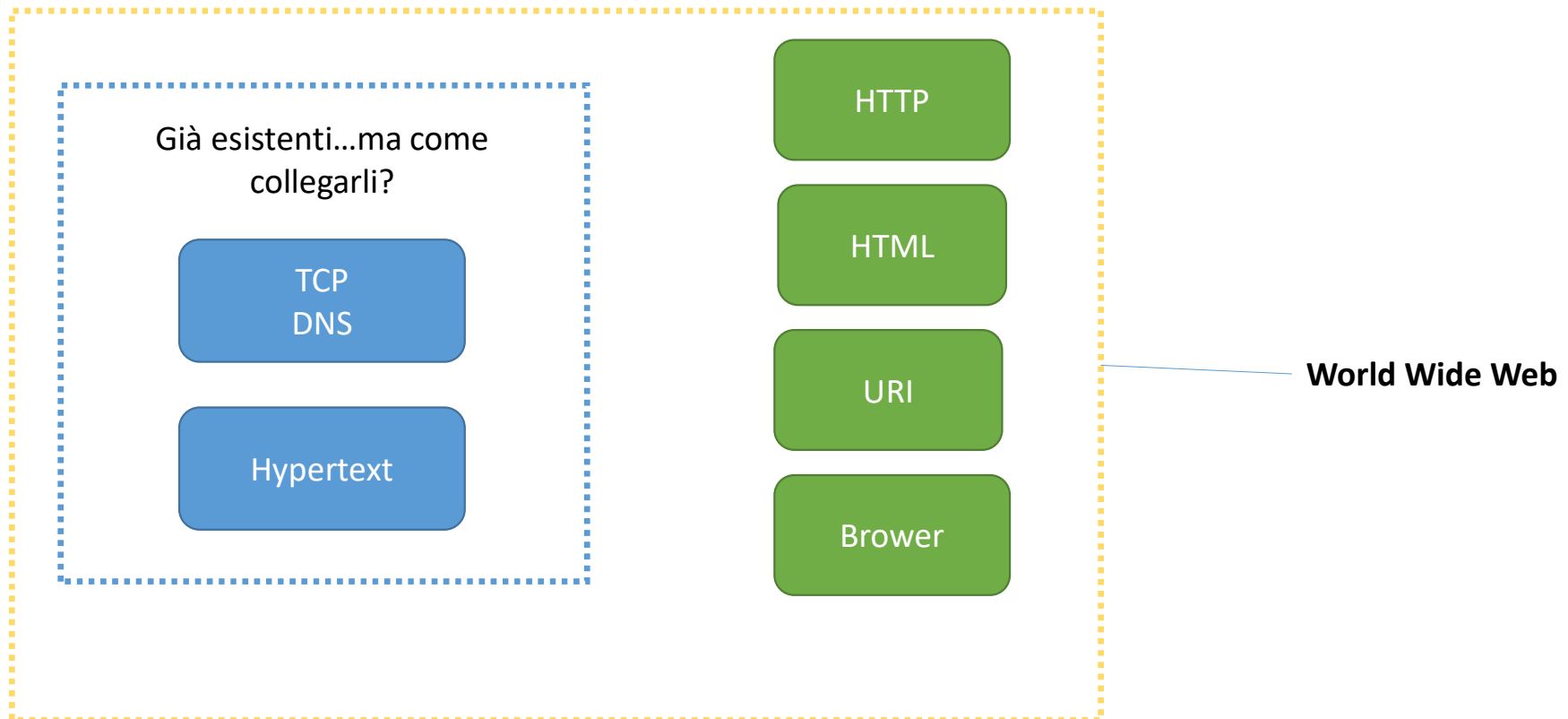


1989 – Proposta

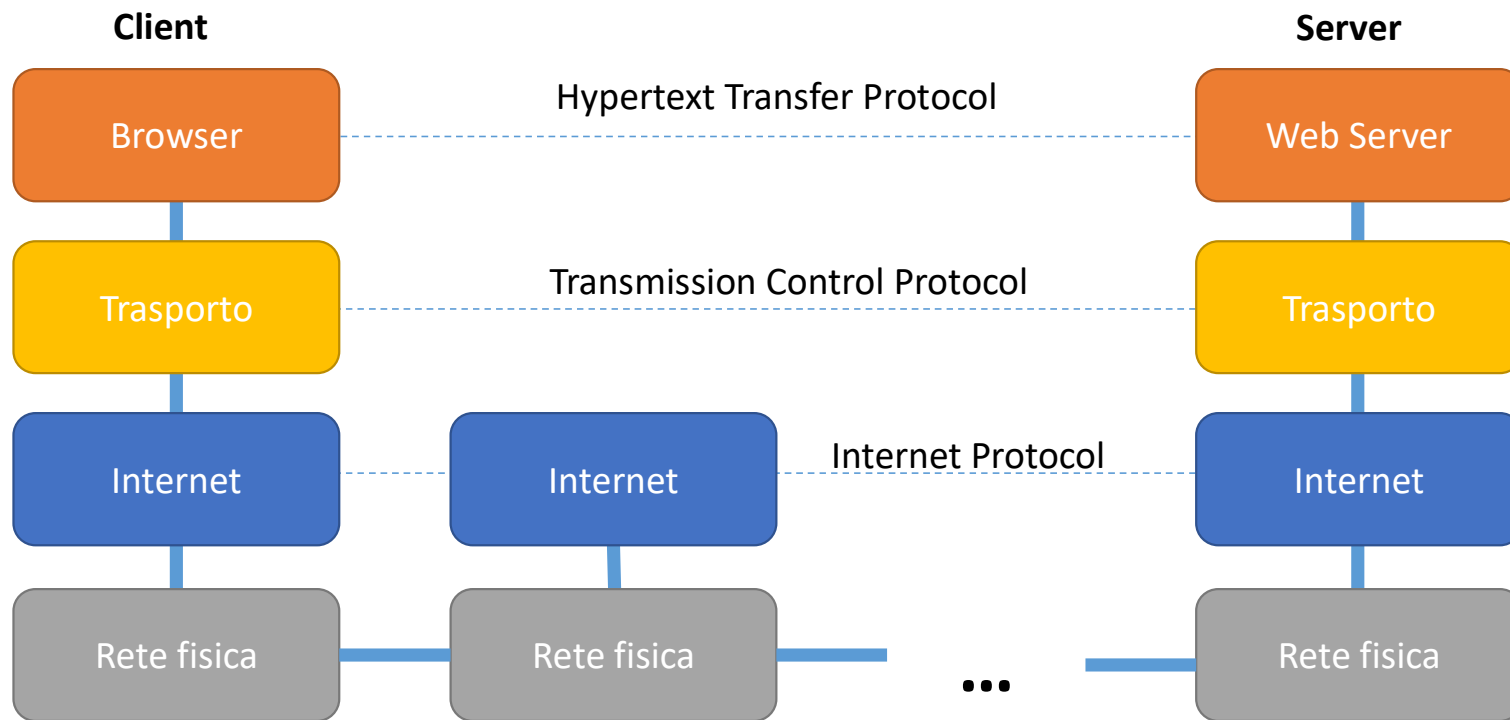
Tim Berners-Lee, CERN

- Documenti accessibili solo tramite nome simbolico (Url), oppure tramite link
- Archiviazione pagine di ipertesto collegandole tra loro
- Accesso indipendente dalla posizione del server

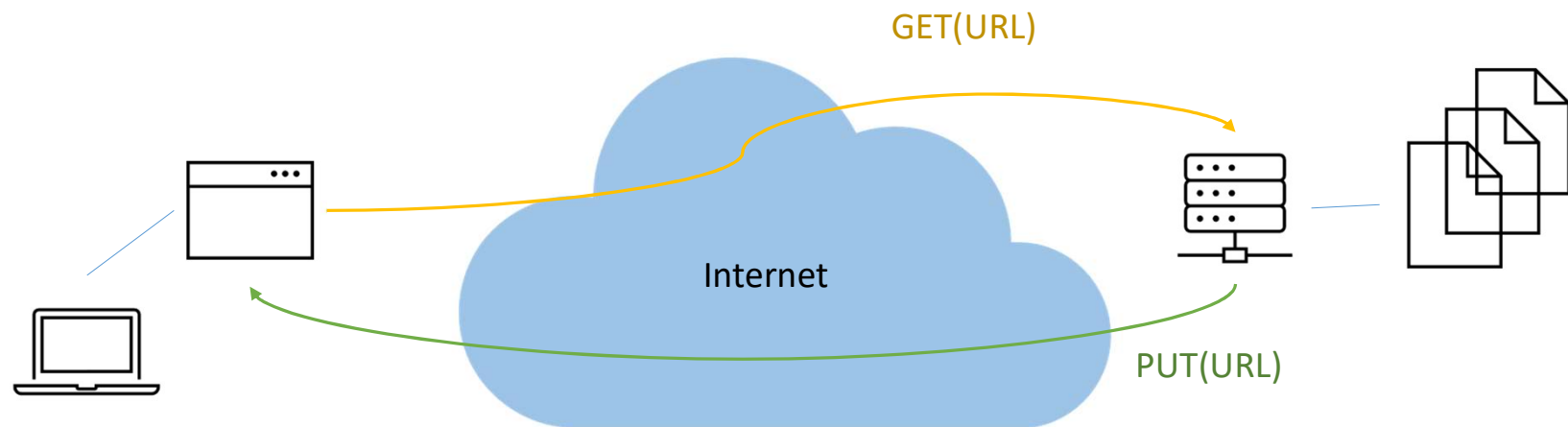
World Wide Web



Protocollo HTTP



Funzionamento HTTP



URI, URL, URN

URI: Uniform Resource Identifier

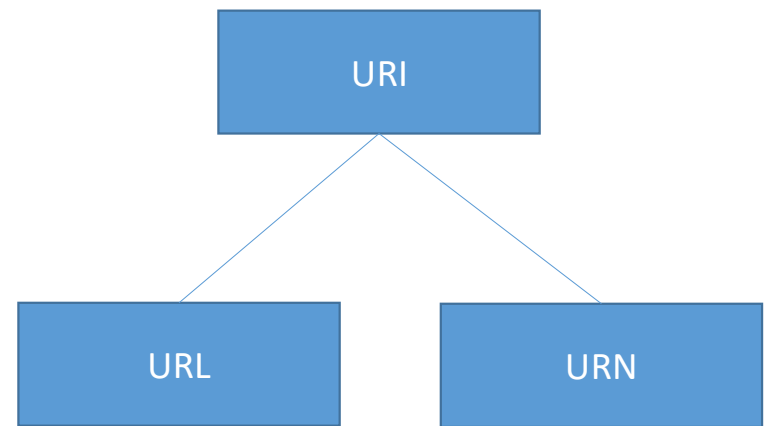
Concetto generale per identificazione
sia in web che per altro.

URL: Uniform Resource Locator

Definizione dell'indirizzo della risorsa

URN: Uniform Resource Name

Definizione del nome della risorsa



URL

<http://www.miosito.it/A/index.html>

<http://>

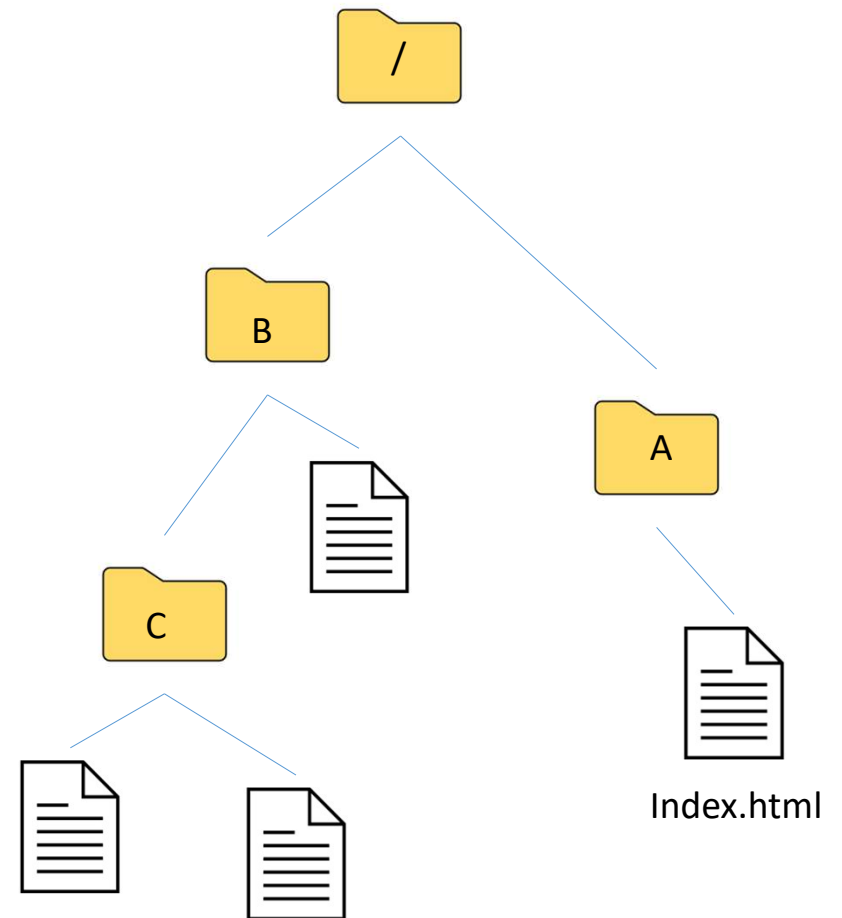
Nome dello schema

www.miosito.it

Nome del dominio

[A/index.html](http://www.miosito.it/A/index.html)

Nome file



Linguaggio HTML

Si basa sulla distinzione tra contenuto, struttura logica e modalità di presentazione delle pagine.

Contenuto della pagina è dato da :

1. Informazioni
2. Meta-informazioni

Il rendering della pagina è gestito dal browser

Programmazione e Markup

Linguaggio di programmazione:

Istruisce una macchina di calcolo

Linguaggio di Markup:

Annota un document in modo che la logica della pagina sia divisa dal contenuto

Pagine statiche e dinamiche

Pagine statiche

Pagina scritta interamente dallo sviluppatore/designer.

Pagine dinamiche

Pagina generate a partire da contenuti:

- Di base dati
- Dati dall'utente
- ...

Si usano linguaggi di scripting

XHTML e XML

XML

Un metalinguaggio per la definizione di linguaggi di markup.

Nato per necessità di avere maggiore libertà nella definizione dei tag

XHTML

Linguaggio di markup che associa alcune proprietà del XML con l'HTML.

Segue la semantica dettata dal XML per gestire pagine web

HTML5, CSS & Javascript



Antonia Sacchitella

Analyst @icubedsrl

antonia.sacchitella@icubed.it



Cos'è HTML5

HyperText Markup Language
(linguaggio di marcatura di ipertesti)

Linguaggio utilizzato per creare pagine web e altri
tipi di documenti visualizzabili in un browser

Standard internazionale le cui specifiche sono
manutenute dal World Wide Web Consortium
(**W3C**)



World Wide Web Consortium (W3C)



Organizzazione internazionale che ha lo scopo di definire gli standard web e di sviluppare al massimo le potenzialità del World Wide Web

Guidato dall'inventore del Web Tim Berners-Lee



Sito italiano <http://www.w3c.it>

Un po' di storia

L'html è uno standard che è stato sviluppato e che si va a perfezionare nel tempo:

1990: HTML tags

1993: HTML 2.0

1995: HTML 3.0

1997: HTML 4.0

1999: HTML 4.01

- 2000: XHTML 1.0

Un gruppo del W3C (WHATWG) si separa e dà vita a delle nuove specifiche che definiscono l'HTML5

- 28 ottobre 2014: Il consorzio riconosce ufficialmente che HTML5 è a tutti gli effetti uno standard Web

Riferimenti

W3C Recommendation HTML5

- www.w3.org/TR/html5

W3C Recommendation HTML5: Edition for Web Authors

- www.w3.org/TR/html5-author

WHATWG community: HTML Living Standard

- <https://html.spec.whatwg.org/multipage>

Mozilla Developer Network (*in italiano!*)

- <https://developer.mozilla.org/it/docs/Web/HTML>

Si può usare HTML5 oggi?

Certo! Tramite apposito DocType

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01



```
<!DOCTYPE html>
```

HTML 5 (Semplificato!)

I browser ignorano i tag che non riconoscono

Struttura generale

Un document HTML è una **struttura ad albero** composta da elementi e testo.

Elemento: composto da una coppia di tag.

Tag: definisce un' informazione di natura strutturale con significato semantico particolare.

Il tag può contenere sia testo che altri elementi.

L'insieme di elementi di un testo viene detto markup.

Attributi

Il tag di apertura di un document può essere dotato di attributi particolari.

Gli attribute sono proprietà in coppia chiave-valore che danno informazioni aggiuntive sulle specifiche del tag.

```
<canvas id="html5_canvas" width="200" height="100">
```


Document Object Model

Il browser interpreta l'intera pagina html. Esegue:

1. Il parsing del markup
2. La rappresentazione ad albero della struttura del documento html (DOM)

Il DOM si compone di tutti gli elementi presenti nel markup, a partire dal tag <html>.

Head e Body

<head></head>

Utilizzato per aggregare i metadati relativi alla pagina.

Riguarda soprattutto informazioni rispetto alle impostazioni della pagina

<body></body>

Rappresenta il corpo della pagina: comprende il contenuto di cui il browser deve fare rendering

DEMO

Come creare un documento HTML



Come faccio a sapere cosa posso usare oggi?

In fase di progettazione

Download attribute  Global 58.7%

When used on an anchor, this attribute signifies that the browser should download the resource the anchor points to rather than navigate to it.

[Current version](#) [Usage relative](#) [Show all](#)

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
8		38	31					4.1	
9		39	43			7.1		4.3	
10		40	44		31	8.4		4.4.4	
11	12	41	45	8	32	9	8	44	44
	13	42	46	9	33				
		43	47		34				
		44	48						

Can I Use ... ?

<http://www.caniuse.com/>

Novità HTML5

Con HTML5, dobbiamo usare

- HTML per il markup e la semantica
- CSS per lo stile
- Javascript per l'interazione

DocType semplificato

- `<!DOCTYPE html>`
Identifica la tipologia di documento che il browser andrà a renderizzare

Nuovi tag che danno un significato semantico alla pagina

Tag <main>

Dovrebbe racchiudere il contenuto principale della pagina.

Deve essere unico nell'intera pagina. Tutti i contenuti ripetuti (come loghi, caselle di ricerca, link, ecc..) non devono esservi inseriti.

```
<body>
  <header>
    <div id="logo">Rocking Stone</div>
    <nav>...</nav>
  </header>
  <main role="main">
    <h1>Guitars</h1>
    <p>The greatest guitars ever built.</p>

    <article>
      <h2>Gibson SG</h2>
      <p>...</p>
    </article>

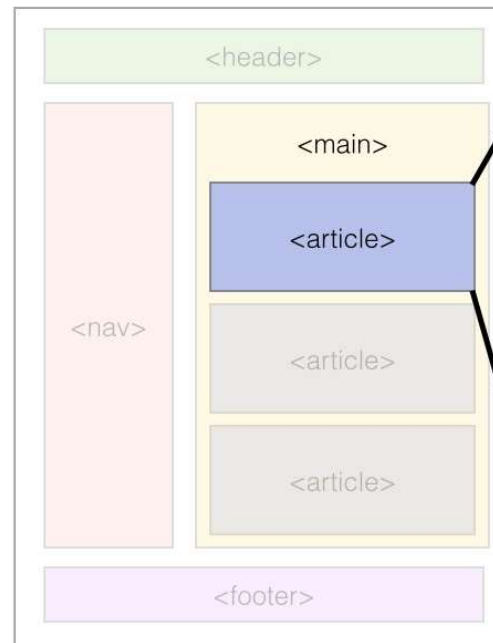
    <article>
      <h2>Fender Telecaster</h2>
      <p>...</p>
    </article>
  </main>
</body>
```

Tag <section> e <article>

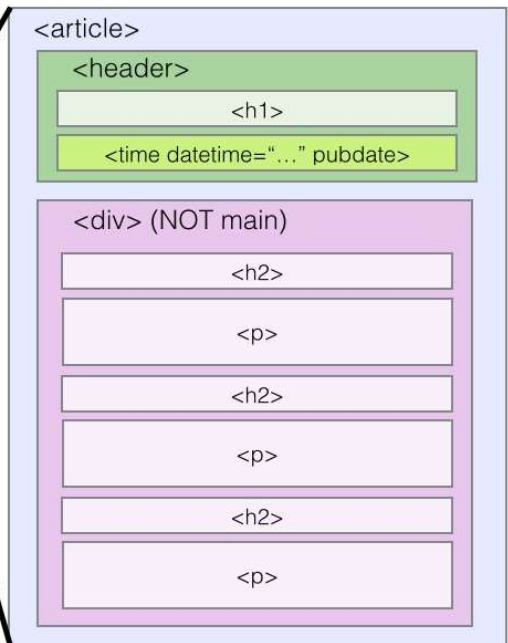
```
<article>
  <h1>How to use HTML5 Sectioning Elements</h1>
  <p>...</p>

  <section>
    <h2>The <main> Element</h2>
    <p>...</p>
  </section>
  <section>
    <h2>The <article> Element</h2>
    <p>...</p>
  </section>
  <section>
    <h2>The <section> Element</h2>
    <p>...</p>
  </section>
  ...
</article>
```

High Level Blog Index



Inside a Specific Post



Tag <aside>

Rappresenta una sezione che include un contenuto legato a quanto trattato nella pagina, ma in qualche modo distinto.

```
<aside>  
  <h3>Fonti</h3>  
  <p>W3C</p>  
</aside>
```

Tag <header>

Rappresenta un blocco di intestazione per una pagina o una sezione (es. titolo e data di pubblicazione di un articolo)

```
<header>  
  <h1>Google buys Nestz</h1>  
  <p>Posted at 11:34am 13° January 2014</p>  
</header>
```

Tag <footer>

Identifica un footer; contiene generalmente informazioni riguardo l'autore e copyright

```
<footer>
```

```
    Copyright Matt West 2014
```

```
</footer>
```

Tag <nav>

Segnala un menu di navigazione (insieme di link)

```
<nav>  
  <ul>  
    <li><a href="#chapter-one">Chapter One</a>  
    <li><a href="#chapter-two">Chapter Two</a>  
    <li><a href="#chapter-three">Chapter Three</a>  
  </ul>  
</nav>
```

Tag <figure> e <figcaption>

Rappresenta un blocco distinto dal principale, che contiene immagini, diagrammi, grafici, esempi di codice.

È possibile associare una didascalia tramite <figcaption>.

```
<figure>  
<figcaption>Figura 2.1</figcaption>  
  
</figure>
```

Formattazione Testo

Elemento	Descrizione
<bdi>	Porzione di testo che deve essere isolata
<mark>	Porzione di testo evidenziato
<ruby>	Porzione di testo con ideogrammi
<wbr>	Interruzione di linea solo se lo richiede il layout

Tag <time>

Indica la data di creazione di un particolare elemento (<article>, documento, <section>, etc)

```
<time datetime="2009-10-22T13:59:47-04:00" pubdate>  
    October 22, 2009 at 1:59pm EDT  
</time>
```

Con l'attributo **pubdate** indica che è la data di pubblicazione

DEMO

Nuovi tag



Form

Utilizzate per fornire la possibilità all'utente di inviare contenuti al server.

Il server prende in input i dati, li elabora e formula una risposta all'utente.

Il tag **<form>** racchiude il blocco di elementi che gestiscono l'input dell'utente

Struttura degli input

```
<form action="/my-url" method="post">
```

```
<label for="name-tbx">name</label>
```

```
<input type="text" name="name" id="name-tbx" />
```

```
<label for="email-tbx">email</label>
```

```
<input type="email" name="email" id="email-tbx" />
```

```
<input type="submit" value="Send!" />
```

```
</form>
```

Data type per gli input

`<input type="email" />`

`<input type="date" min="2010-01-01" max="2011-01-01" />`

`<input type="range" min="0" max="100" />`

`<input type="search" results="10" placeholder="Cerca..." />`

`<input type="tel" pattern="regex" />` (<http://html5pattern.com/Phones>)

`<input type="color" placeholder="es: #333333" />`

`<input type="number" step="1" min="-5" max="10" />`

Controllo di una form

Attributo **required**, che consente la validazione

```
<style>
    [required] { background-color: red; }
    :invalid { background-color: red; }
</style>

<input type=text required />
```

* non supportati da IE 9, supportate da IE 10

E poi...

Meter



```
<meter min="0" low="10" optimum="50" high="90" max="100"></meter>
```

Rappresenta quanto siamo vicini ad un valore massimo stabilito

Progress

80%



```
<progress max="100" value="80"></progress>
```

Rappresenta a che punto siamo di un determinato processo visto come un ideale 100%

Entrambi producono la visualizzazione di barre di avanzamento, la cui colorazione dipende dal browser e dal OS

DEMO

HTML Forms 2.0



Esercitazione n. 1

Realizzare una pagina Web statica specificando i seguenti punti:

1) HomePage

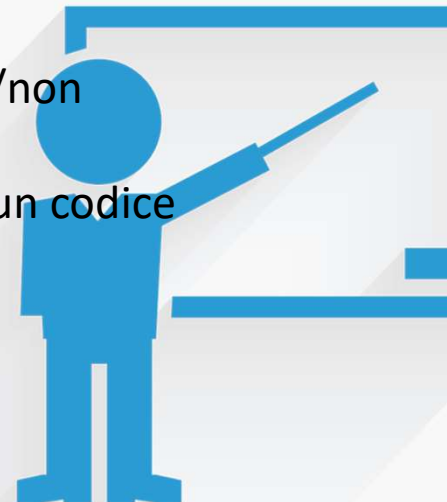
- inserire un logo
- posizionare logo e titolo della pagina in alto sulla stessa linea
- barra di navigazione verticale con link: Home, Aggiungi Libro.

2) Aggiungi libro

- creare una form per l'inserimento di titolo, autore, codice ISBN, disponibile/non disponibile.

Input di titolo e autore devono essere di tipo testo, codice ISBN deve essere un codice numerico di 13 cifre,

disponibile/non disponibile deve essere scelto tramite radio button.



CSS

- Utilizzati per stilizzare delle pagine web
- Passano in stato «Recommended» del W3C nel 1996
- Si inseriscono all'interno del tag <head> di una pagina HTML e collegano un file esterno

```
<html>  
    <head>  
        <link rel="stylesheet" type="text/css" href="style.css" >  
    </head>  
</html>
```

- Questa soluzione garantisce il massimo riuso di codice possibile potendo essere richiamato in ciascuna delle pagine web HTML desiderate.



Regole CSS

E' costituito da una sequenza di *regole*, costituite da un *selettore* e da un *blocco di dichiarazioni*, racchiuso tra parentesi graffe

```
selettore {  
    proprietà1:valore1;  
    proprietà2:valore2;  
}
```

Le proprietà CSS sono numerose, le più utilizzare sono circa 60, tra cui:

background-color: definisce il colore di sfondo di un elemento;

color: definisce il colore del testo di un elemento;

font-size: definisce la dimensione di un testo;

float: definisce un blocco *fluttuante*, spostabile su uno dei lati sx/dx del suo contenitore

margin e *padding*: definiscono lo spazio circostante agli elementi

text-align: definisce l'allineamento degli elementi, tra cui il testo.

Stile per i tag

```
body {  
    font-family: arial;  
    font-size: 15px;  
}  
p, div {  
    color: ■ red;  
    background: ■ white;  
}  
* {  
    color: ■ green;  
}
```

* Applica lo stile specificato a tutti i tag
Viene chiamato selettore universale

Stile con l'attributo class

```
.warning {  
  color: red;  
  background: white;  
}  
  
.small {  
  font-size: 70%;  
}
```

```
<p class="warning">
```

Classe di stile: definisce uno stile richiamabile da ogni tag con l'attributo class.

Permette di definire uno stile per tipologie di tag differenti.

Mescolare tag e classi

```
p, .warning{  
  font-weight: bold;  
}
```

Mescolando i selettori per tag e per classe, lo stile verrà applicato se il tag:

- è un p
- Oppure ha class warning
- Oppure è entrambi

Pseudo-Classi

Le pseudo-classi sono stati che può avere un particolare tag.

Un esempio è il tag <a> riferito ai link che ha diversi stati in base all'interazione che ha avuto con l'utente:

- Link: link visualizzato in pagina
- Hover: utente passa il mouse sopra al link
- Visited: link già stato visitato

```
a:hover{  
  color:  chocolate;  
}
```

Possono essere anche utilizzate in presenza di classi!

Pseudo-Elementi

Gli pseudo-elementi sono elementi creati dal browser per consentire una formattazione più semplice.

Pseudo Elemento	Descrizione
::first-name	Prima riga di un elemento
::first-letter	Prima lettera di un elemento
::before	Inserire del contenuto prima dell'elemento
::after	Inserire del contenuto dopo dell'elemento

DEMO

CSS



Selettori di attributi

```
input[type="button"] {  
  color: ■ red;  
  background: □ black;  
}  
input[type="checkbox"], input[type="radio"] {  
  color: □ black;  
  background: transparent;  
}  
input[disabled], input[readonly] {  
  background: ■ gray;  
}
```

Consentono di mettere uno stile in attribute.

Es: attribute delle form

Selettori Combinati

```
p strong {  
  color: ■red;  
}  
div * p {  
  background: ■yellow;  
}
```

È possibile combinare più selettori tra loro.
(spazio) / * : tutti i tag strong all'interno di
un p (anche innestati)

```
p > strong {  
  color: ■green;  
}
```

> : tutti i tag strong direttamente innestati in
p

Posizionamento

Gli elementi possono essere divisi in

- Inline: elementi in linea (img, span, a)
- Block: contenitori che delimitano spazio (div, section, p)

È possibile gestire il posizionamento degli elementi block con:

- Float : per ridisporre l'elemento con altri
- Clear: “pulire” lo spazio attorno all'elemento

Posizionamento
Relativo

Posizionamento Assoluto

Elementi potrebbero essere sovrapposti.

Le istruzioni sono:

- Position: tipo di posizionamento (static, absolute)
- Z-Index: gestione degli elementi in primo piano, secondo piano.
- Top, Left...: specifica l'allineamento

DEMO

Selettori



Esercitazione n. 2

Riprendendo l'esercitazione n. 1 aggiungere un foglio di stile CSS specificando le stesse caratteristiche.

Passando sui campi della barra di navigazione con il mouse, deve cambiare il colore di sfondo.

Gestire i colori del link visitato/non visitato. Eliminare la sottolineatura data di default.

- tabella: con i dettagli specificati nella form

CSS:

- Gestire lo stile in un file CSS esterno

- Usare diversi selettori

- cambiare sfondo alle righe della tabella, passandoci sopra. solo nel body.

Opzionale:

- Rendere la pagina responsiva

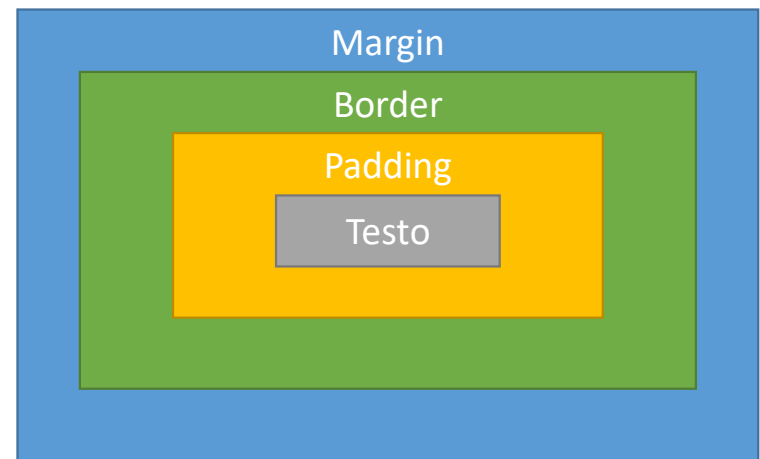
N.B. Questi sono requisiti minimi. Per il resto, libera fantasia.



Box Model

Ogni element ha una dimensione data dalla somma dei seguenti parametri:

- **Border**: bordo dell'elemento
- **Padding**: spazio tra contenuto dell'elemento e il bordo
- **Margin**: spazio tra il bordo dell'element e quelli circostanti



Gestione del Border

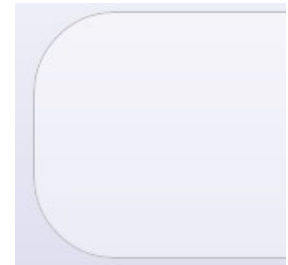
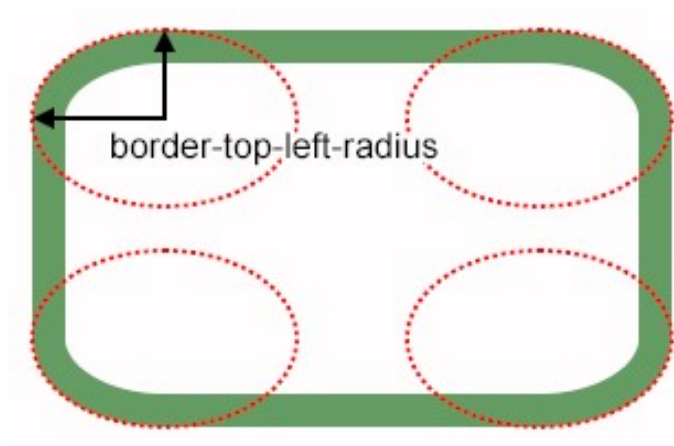
È possibile impostare le seguenti caratteristiche:

Proprietà	Descrizione
Border-width	Dimensione del bordo
Border-Style	Stile del bordo (dotted line, solid line)
Border-Color	Colore del bordo
Border Radius	Curvatura degli angoli

Bordi arrotondati

Attraverso l'attributo border-radius

```
.border{  
  border-radius:50px;  
}
```



Disposizione degli elementi

La disposizione degli elementi viene gestita tramite tre proprietà:

- Float
- Visibility: visibilità dell'elemento (hidden per nascondere)
- Display: gestisce rendering dell'elemento (inline, block, inline-block, none)

Media Query

Utilizzate per cambiare lo stile e adattare il contenuto in base al dispositivo che lo utilizza.

L'idea è creare un CSS per ogni tipo di visualizzazione che si intende supportare creando delle regole in base alle quali sarà il browser a decider quale CSS utilizzare.

Tali regole includono la definizione del tipo di device e delle sue caratteristiche.

DEMO

CSS



Javascript

Javascript è essenziale per HTML5: è la parte "funzionale" delle nostre app
Quasi tutti i browser di nuova generazione hanno un engine che migliorano notevolmente le performance del Javascript rispetto ai browser del passato

Javascript è un linguaggio di **scripting**

- Non è necessaria compilazione
- È semplice testo facilmente leggibile

Javascript è un linguaggio **interpretato***

- È eseguito direttamente dal browser
- Ha funzioni limitate = maggiore sicurezza

Javascript interagisce con gli **elementi HTML** della pagina

- Buone capacità di interattività e modellazione

* Le ultime iterazioni dei browser fanno una compilazione "al volo"



Inserimento del codice nella pagina

Direttamente come codice nell'HTML

- `<script><!-- --></script>`
- Semplice codice “**occasionale**”

Referenziando un file esterno

- `<script src="javascript.js"></script>`
- **Libreria di funzioni** di uso ricorrente nel progetto

Attenzione all'**ordine di esecuzione** degli script

- Il codice inserito nel tag `<head>` viene eseguito prima degli altri

Commenti, testi, apici

Il codice può essere **commentato**:

- `/* commento di più righe */`
- `// commento riga singola`

I testi possono essere racchiusi tra **apici singole** (' ') o **doppie** (" ")

- `alert('javascript!'); alert("javascript!");`

I **caratteri speciali** devono essere anteposti da \

- `alert("javascript \= \'scripting\'");`

Tipi di base

Tipo	Descrizione
Object	Tipo base di tutti gli oggetti
Date	Rappresenta una data
Boolean	Rappresenta un booleano
Number	Rappresenta un numero
String	Rappresenta una stringa
Array	Lista di oggetti
Undefined	Rappresenta un valore non esistente

Var, Let o Const?

Var, let e Cost permettono di dichiarare una variabile.

Var : definisce la variabile sia a livello temporaneo che a livello globale in maniera dipendente alla sua dichiarazione

Let : definisce la variabile in maniera coerente al suo contesto di funzionalità.

Const : definisce una variabile il cui valore non può essere cambiato.

L'uso di var è deprecato. Si utilizza soprattutto let!

Funzioni

Blocco di codice eseguibile

- `function nomeFunzione(parametri) {
 //istruzioni
}`

Chiamata **diretta**

- `nomeFunzione(parametri)`

Chiamata da **link**

- `<a href="javascript:nomeFunzione(parametri)"..`

Le funzioni sono **Case Sensitive**

Attenzione al posizionamento del codice

Eventi

Direttamente nel **markup**

- `<button onclick="Nomefunzione(parametri)" ..`

Da **codice**

- `Oggetto.evento = handler`

Principali funzioni e proprietà

typeof: Verifica del tipo di una variabile

window: accesso alla pagina

- open: popup
- opener: accesso alla pagina “chiamante”
- history: storico di navigazione
- location: spostamenti di pagina
- navigator: informazioni del browser dell'utente

document: accesso al DOM della pagina

- getElementById
- getElementByName

alert: messaggi di avviso

Javascript – Principali utilizzi nell'HTML

Cercare elementi per **Id**:

- `document.getElementById('section1');`

Cercare elementi per classe:

- `document.getElementsByClassName('section')`

Cercare elementi per nome **tag**:

- `document.getElementsByTagName('div');`

Cercare elementi tramite selettori CSS (Selectors API):

- `document.querySelectorAll("ul li:nth-child(odd)");`
- `document.querySelectorAll("table.test > tr > td");`

Espressioni condizionali

```
If (variabile == valore){  
    //istruzioni  
}
```

== uguale

!= diverso

> maggiore di

>= maggiore uguale di

< minore di

<= minore uguale di

Operatori logici: &&, ||, !(negazione)

Else

Operatore ternario: <espr1> ? <espr2> :<espr3>

Caratteri speciali per le stringhe

`\n` : nuova riga

`\t` : tab orizzontale

`\b` : backspace

`\r` : ritorno a capo

`\\` : commento

`\'` : apice singolo

`\"` : apice doppio

DEMO

Javascript



Esercitazione n. 3

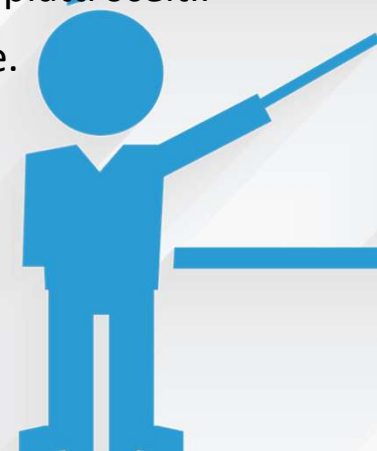
Realizzare una pagina Web di un ristorante.

La pagina Web è suddivisa in:

- Home: con alcune immagini ed informazioni sul ristorante
- Pagina di prenotazione: in questa pagina è possibile inserire i propri dati (nome, cognome, email e data di prenotazione). Nella pagina è presente un bottone che cliccato mostra un pop-up con la frase «Hai prenotato correttamente»
- Pagina Menu. Contiene una tabella con tre colonne (nome del piatto, descrizione e prezzo). Utilizzare una funzione javascript che calcola il conto sulla base dei piatti scelti.
- Pagina di contatto: contiene tutte le informazioni di contatto del ristorante.

Personalizzare la pagina Web con un CSS specifico.

Realizzare le operazioni “dinamiche” con un file javascript specifico.



Storage

Un'applicazione richiede la persistenza di alcuni dati:

- Parametri di configurazione
- Preferenze dell'utente
- Informazioni di accesso ...

Questi possono essere:

1. Mantenuti nel server
2. Salvati come cookie



Local Storage o Session Storage

Gestione dello Storage

Metodo	Descrizione
Length	Numero totale di elementi memorizzati
Key(index)	Ottenere una chiave in base alla posizione nell'indice
getItem(key)	Ottenere un valore memorizzato data la chiave corrispondente
setItem(key, value)	Salvare un nuovo element con la corrispettiva chiave
removeItem(key)	Rimuovere element tramite la chiave
Clear()	Rimuovere tutti gli elementi precedentemente memorizzati

Demo

WebStorage



Bootstrap

Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web.

Contiene modelli di progettazione basati su HTML e CSS per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.

Utilizzare Bootstrap nel proprio progetto

Specificare le informazioni CSS

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO17+AMvyTG2x"
      crossorigin="anonymous">
```

E javascript

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
      integrity="sha384-IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p"
      crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"
      integrity="sha384-Atwg2Pkwv9vp0ygtn1JAoJH0nYbwNjLPhwyoVbhoPwBhjQPR5VtM2+xf0Uwh9KtT"
      crossorigin="anonymous"></script>
```

nel proprio progetto

Esercitazione n. 4

Modificate l'aspetto della pagina Web realizzata nell'esercitazione 3 utilizzando il framework Bootstrap.

Aggiungere all'interattività dell'utente un metodo che in fase di prenotazione salvi sullo storage locale le informazioni di nome, cognome ed email.



© 2021 iCubed Srl

La diffusione di questo materiale per scopi differenti da quelli per cui se ne è venuti in possesso è vietata.

iCubed s.r.l. • Piazza Durante, 8 – 20131, Milano
• Phone: +39 02 57501057 • P.IVA 07284390965

