# Abilità informatiche

A.A. 2023/2024

03b - Internet e World Wide Web

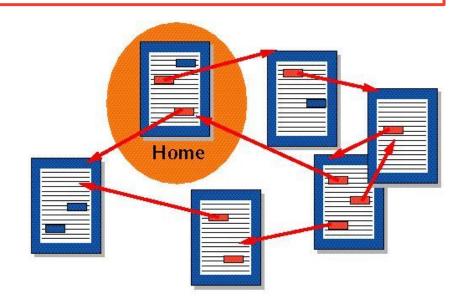
## Sebastian Barzaghi

<u>sebastian.barzaghi2@unibo.it</u> https://orcid.org/0000-0002-0799-1527

Riassunto della lezione precedente (03a)

# Cos'è un "ipertesto"?

Un modo non lineare di presentazione dell'informazione tramite un insieme di unità informative (dette nodi), connesse tra loro da collegamenti (detti link)



Fonte: Robert Cailliau, CC BY-SA 3.0 <a href="https://creativecommons.org/licenses/by-sa/3.0">https://creativecommons.org/licenses/by-sa/3.0</a>, via Wikimedia Commons.

# Elementi principali

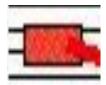
## Nodo

unità minima di informazione, dotata di autonomia e completezza



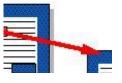
## **Ancora**

frammento di un nodo da cui un link viene orientato verso un altro frammento di un nodo



## Link

connessione tra una coppia di ancore



## Cos'è Internet?



Una rete di comunicazione tra dispositivi distribuiti a livello globale

- TCP
- IP
- DNS
- ..



Fonte: <a href="https://rubenverborgh.github.io/WebFundamentals/">https://rubenverborgh.github.io/WebFundamentals/</a>

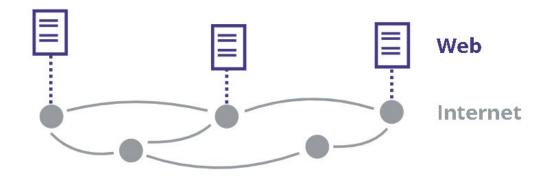
## Cos'è il World Wide Web?



Un sistema documentale ipertestuale distribuito su Internet



- HTTP
- HTML
- URL
- ...

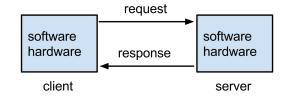


Fonte: <a href="https://rubenverborgh.github.io/WebFundamentals/">https://rubenverborgh.github.io/WebFundamentals/</a>

## Modello Client - Server



Architettura di rete in cui un agente computer o software - (il client) accede ai servizi o alle risorse di un altro agente - computer o software -(il server) attraverso una rete



Fonte: Lubaochuan, CC BY-SA 4.0

 $<\! https://creative commons.org/licenses/by-sa/4.0>, via$ 

Wikimedia Commons.

https://upload.wikimedia.org/wikipedia/commons/6/6

9/Client-server\_model.svq

## Cos'è un client?



Un computer o un software che accede a un servizio fornito da un server

Consente agli utenti di interagire facilmente con un servizio o una risorsa remota (es. sito Web, un server di posta elettronica o un sistema di archiviazione cloud)



Fonte: Raysonho @ Open Grid Scheduler / Grid Engine, Public domain, via Wikimedia Commons.

https://upload.wikimedia.org/wikipedia/commons/4/40/PCExpanion.jpg

## Cos'è un server?



Computer o software che fornisce servizi o risorse ai client, attraverso una rete

Progettato per eseguire specifiche funzioni di elaborazione e gestione delle risorse

→ Es. un server Web ospita un sito Web e lo serve ai client che richiedono di accedere e visualizzare il sito



Victor Grigas, CC BY-SA 3.0 <a href="https://creativecommons.org/licenses/by-sa/3.0">https://creativecommons.org/licenses/by-sa/3.0</a>, via Wikimedia Commons.

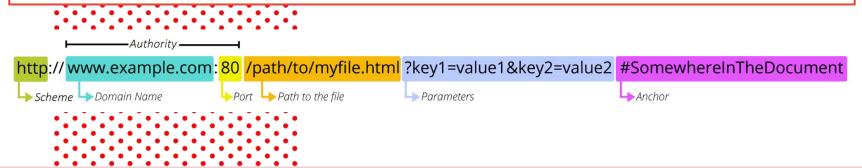
https://upload.wikimedia.org/wikipedia/commons/6/69/ Wikimedia\_Foundation\_Servers-8055\_35.jpg

## Cos'è un Uniform Resource Locator?



Meccanismo usato dai browser per recuperare risorse pubblicate sul Web

Una stringa di testo che *identifica* (~codice fiscale) e *localizza* (~indirizzo di casa) una risorsa web



# Tecnologie a supporto dei client



Protocollo HTTP

Uno o più formati di rappresentazione (es. HTML)

Tecnologie di rete: TCP, IP, DNS

# 3.2.2 HTTP

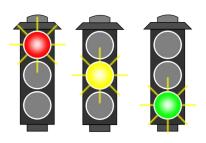
Definizione Elementi Esempi

## Cos'è un protocollo?

Per comunicare, gli elementi all'interno delle reti devono seguire **regole comuni** 

Insieme di regole e di messaggi che governano la comunicazione tra due entità

La definizione di ogni protocollo consiste nel fornire un insieme di regole non ambigue, definendo i messaggi che possono essere scambiati tra entità, il loro significato e le azioni da intraprendere in ogni situazione (~ semafori)



## Cos'è un protocollo?



Deve essere espresso in un particolare linguaggio, un insieme di segnali e/o simboli comprensibili alle entità al fine di avviare e gestire la comunicazione

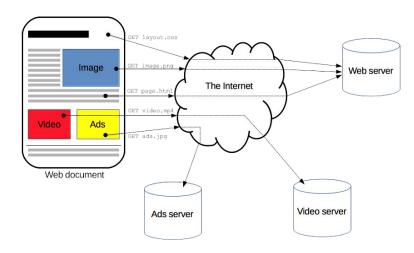
- Una sintassi da seguire per costruire i messaggi
- Delle regole interpretative del messaggio, per definire la semantica dei messaggi
- Dei meccanismi per **sincronizzare** la comunicazione
- Dei meccanismi per correggere e/o gestire eventuali errori che possono intercorrere nello scambio dei messaggi

## Cos'è il protocollo HTTP?



Per comunicare, gli elementi all'interno delle reti devono seguire regole comuni

Protocollo di comunicazione utilizzato per il trasferimento di dati tra un client e un server su una rete

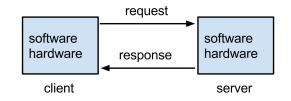


Fonte:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

## Cos'è il protocollo HTTP?

HTTP standardizza come client e server comunicano tramite il **metodo richiesta-risposta**, in cui il client invia una **richiesta** al server e il server risponde con una **risposta** contenente i dati richiesti



Fonte: Lubaochuan, CC BY-SA 4.0

<a href="https://creativecommons.org/licenses/by-sa/4.0">https://creativecommons.org/licenses/by-sa/4.0</a>, via

Wikimedia Commons.

https://upload.wikimedia.org/wikipedia/commons/6/6

9/Client-server\_model.svq



Permette lo scambio di dati tra un server e un client

Esistono due tipi di messaggi HTTP:

- richiesta: messaggio mandato dal client per richiedere una determinata azione da parte del server
- risposta: messaggio mandato dal server come risposta alla richiesta del client





Una **testa** composta da:

- una riga descrittiva (di richiesta o di risposta)
- intestazioni (o header)
   opzionali che specificano la richiesta o descrivono il corpo del messaggio

## PUT /create page HTTP/1.1

Host: localhost:8000 Connection: keep-alive

Upgrade-Insecure-Requests: 1
Content-Type: text/html

Content-Length: 345

Body line 1 Body line 2

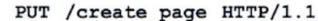
...

#### Fonte:



Una *testa* composta da:

- una riga descrittiva (di richiesta o di risposta)
- intestazioni (o header)
   opzionali che specificano la richiesta o descrivono il corpo del messaggio



Host: localhost:8000 Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: text/html

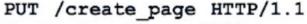
Content-Length: 345

Body line 1 Body line 2

...

Fonte:

Una riga vuota di separazione tra testa e corpo



Host: localhost:8000 Connection: keep-alive

Upgrade-Insecure-Requests: 1

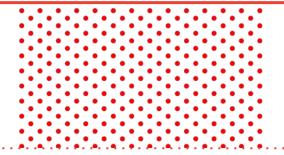
Content-Type: text/html

Content-Length: 345

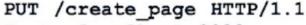
Body line 1 Body line 2

\*\*\*

#### Fonte:



Un **corpo** (o **body** o **payload**) con il contenuto del messaggio



Host: localhost:8000 Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: text/html

Content-Length: 345

Body line 1 Body line 2

Fonte:

## Richiesta HTTP



- testa:
  - riga di richiesta: <u>metodo</u>, percorso, versione di HTTP
  - intestazioni: nome del client, lingua, codifica, ecc.
- riga vuota
- corpo



## Requests

```
POST / HTTP/1.1

Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;...)... Firef
Accept: text/html,application/xhtml+xml,...,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary-Content-Length: 345
```

-12656974 (more data)

#### Fonte:



## **GET**

Permette di richiedere informazioni sulla risorsa definita dall'URL specificato – per esempio come conseguenza di un click su un collegamento ipertestuale

GET /users HTTP/1.1 Host: api.example.com

Un client invia una richiesta GET al server per ottenere la lista degli utenti registrati nel sistema



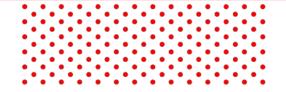
## **POST**

Permette di creare sul server web contattato la risorsa specificata dall'URL utilizzato nella richiesta, e di associare informazioni incluse nella richiesta

```
POST /users HTTP/1.1
Host: api.example.com
Content-Type: application/json

{
    "name": "John Doe",
    "email":
"j.doe@example.com",
    "password": "password123"
```

Un client invia una richiesta POST al server per creare un nuovo utente nel sistema. La richiesta include i dati dell'utente da creare nel corpo della richiesta (ad esempio, nome, email, password).



## DELETE

Permette di rimuovere dal server web contattato tutte le informazioni relative alla risorsa specificata nella richiesta tramite l'URL DELETE /users/123 HTTP/1.1 Host: api.example.com

Un client invia una richiesta DELETE al server per rimuovere un utente specifico dal sistema. L'ID dell'utente da eliminare è specificato nell'URL.



## **PUT**

Permette di specificare informazioni aggiuntive, incluse nella richiesta, ad una risorsa esistente che già risiede sul server

```
PUT /users/123 HTTP/1.1
Host: api.example.com
Content-Type: application/json

{
    "name": "John Doe",
    "email": "john.doe@example.com",
    "password": "nuovapassword456"
}
```

Un client invia una richiesta PUT al server per aggiornare le informazioni di un utente specifico nel sistema. La richiesta include i nuovi dati dell'utente nel corpo della richiesta, mentre l'ID dell'utente da aggiornare è specificato nell'URL.



- testa:
  - riga di risposta: versione di protocollo, <u>codice di stato</u>, messaggio di stato
  - intestazioni: nome del client, lingua, codifica, ecc.
- riga vuota
- corpo

## Responses

# HTTP/1.1 403 Forbidden Server: Apache Content-Type: text/html; charset=iso-8859-1 Date: Wed, 10 Aug 2016 09:23:25 GMT Keep-Alive: timeout=5, max=1000 Connection: Keep-Alive Age: 3464

Date: Wed, 10 Aug 2016 09:46:25 GMT

X-Cache-Info: caching Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)

#### Fonte:



1XX (es. 100 "Continue")

Risposta informativa preliminare o di diagnostica (molto raro, di solito non usato) HTTP/1.1 100 Continue

Content-Length: 0

Connection: keep-alive

Il server comunica che ha ricevuto l'intestazione della richiesta e che il client può procedere a inviare il corpo della richiesta



2XX (es. 200 "OK")

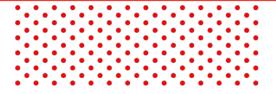
La richiesta è stata ricevuta, compresa, accettata e elaborata con successo dal server

```
"message": "Richiesta completata
con successo.",
    "data": {
        "user_id": 123,
        "name": "John Doe",
        "email":
"john.doe@example.com"
    }
}
```

Content-Type: application/json

HTTP/1.1 200 OK

Il server comunica che la richiesta è stata completata con successo e che i dati richiesti sono inclusi nella risposta



3XX (es. 301 "Moved permanently")

Il client deve effettuare ulteriori azioni per completare la richiesta HTTP/1.1 301 Moved Permanently Location:

https://www.new-example.com

Il server comunica che la risorsa richiesta è stata spostata permanentemente in un nuovo URI, e il client dovrebbe aggiornare il proprio URL di riferimento



4XX (es. 404 "Not Found")

Errori che dipendono dalla richiesta del client (spesso derivano da errori nell'URL della richiesta)

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "Risorsa non trovata",
    "message": "La pagina richiesta
non esiste."
}
```

Il server comunica che la risorsa richiesta non è stata trovata sul server



5XX (es. 500 "Internal Server Error")

Errori che dipendono dal server durante il tentativo di elaborare la richiesta del client

HTTP/1.1 500 Internal Server Error

Content-Type: text/plain

Si è verificato un errore interno

del--server. Il server comunica che un errore generico lato server gli impedisce di completare la richiesta del client

# 3.2.3 HTML

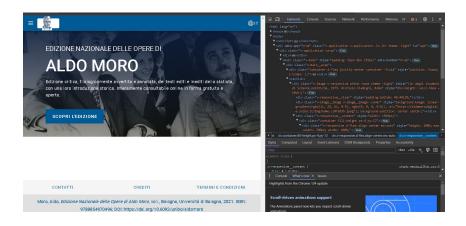
Definizione Elementi Esempi

## Pagina Web



Documento ipertestuale pubblicato sul World Wide Web e visualizzato dall'utente tramite un client (browser, app, ecc.)





Consiste in un documento HTML e file correlati che il client richiede ad uno o più server per elaborarli e generarne la visualizzazione sullo schermo del dispositivo

## Tecnologie Web



Tecniche e strumenti continuano ad evolvere (es. framework), ma in sostanza le tecnologie di visualizzazione che stanno alla base di una pagina Web sono le stesse



## Cos'è l'Hypertext Markup Language?



Linguaggio di marcatura che definisce la struttura e la semantica del contenuto di una pagina web



Consiste in una serie di **elementi** che vengono utilizzati per racchiudere parti del contenuto della pagina per farle apparire e/o comportare in una certa maniera

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
My first paragraph.
</body>
</html>
```

# **My First Heading**

My first paragraph.

# Cos'è Cascading Style Sheets?



Linguaggio per definire lo stile delle pagine Web



Basato sul concetto di *selettore* per individuare specifici elementi HTML all'interno della pagina Web e applicargli un certo tipo di formattazione

#### Codice HTML

```
Sono un paragrafo centrato e con il testo rosso.
Anche io!
E io pure!
```

#### Codice CSS

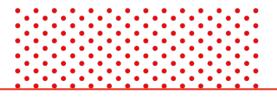
```
text-align: center;
color: red;
```

Sono un paragrafo centrato e con il testo rosso.

Anche io!

E io pure!

# Cos'è JavaScript?



Linguaggio di programmazione usato per aggiungere interattività alle pagine Web



Esecuzione di determinate azioni in seguito allo scatenarsi di specifici eventi



#### Codice HTML

```
cbody>
Un paragrafo.
<button type="button" onclick="myFunction()">Clicca
qui</button>
</body>
```

#### Codice JavaScript

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Oh no, hai cliccato!";
}
```



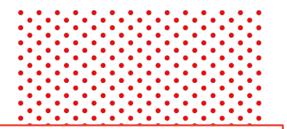
Un paragrafo.

Oh no, hai cliccato!

Clicca qui

Clicca qui

#### Marcatura di un testo



Annotazione del testo per definire esplicitamente i ruoli strutturali e semantici delle parti di cui è costituito

#### libro

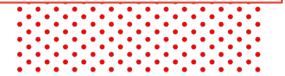
#### capitolo

#### paragrafo

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

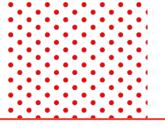
paragrafo

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

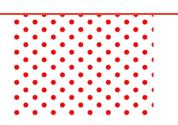


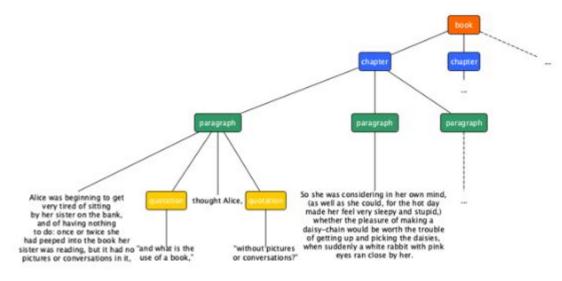
Fonte: <a href="https://basic-inf.github.io/2022-2023/chapters/08.pdf">https://basic-inf.github.io/2022-2023/chapters/08.pdf</a>

### Marcatura di un testo



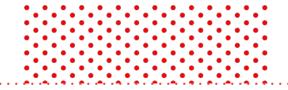
La marcatura trasforma il testo in una struttura ad albero





Fonte: https://basic-inf.github.io/2022-2023/chapters/08.pdf

#### Marcatura di un testo



Diversi linguaggi di marcatura (es. SGML, XML)

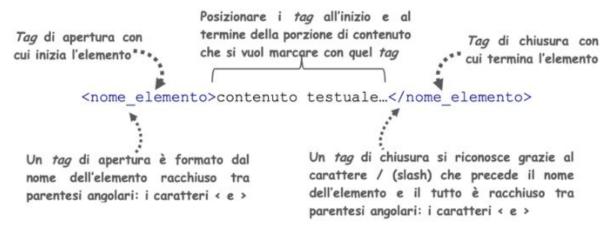


Metalinguaggi → definiscono regole sintattiche per specificare la marcatura, ma non impongono alcun vocabolario

```
libro>
   <capitolo>
       <paragrafo>
       </paragrafo>
       <paragrafo>
       </paragrafo>
   </capitolo>
</libro>
```

### Elemento

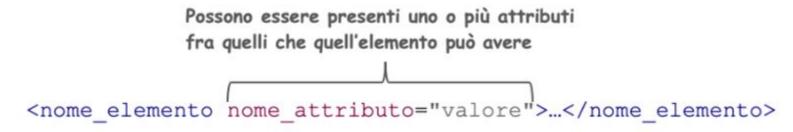
Termine informativo che esprime la semantica della porzione di testo a cui si riferisce



Fonte: <a href="https://basic-inf.github.io/2022-2023/chapters/08.pdf">https://basic-inf.github.io/2022-2023/chapters/08.pdf</a>

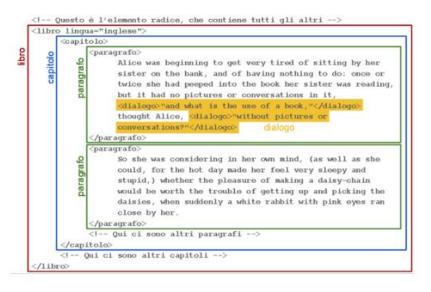
#### Attributo

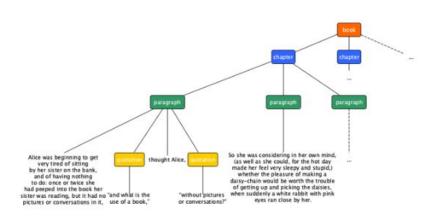
Informazione aggiuntiva che si riferisce all'elemento a cui viene assegnata, sotto forma di coppia chiave-valore inserita nel marcatore di apertura



Fonte: https://basic-inf.github.io/2022-2023/chapters/08.pdf

### Marcatura di un testo: esempio





Fonte: https://basic-inf.github.io/2022-2023/chapters/08.pdf

#### Introduzione a HTML

Linguaggio che segue una sintassi simile a XML e che inoltre utilizza uno specifico vocabolario costituito da elementi ed attributi per identificare i vari ruoli strutturali e semantici di una pagina Web



Il contenuto di un elemento può essere testo, altri elementi, o entrambi



```
<h1> Questo è un titolo. </h1>
Questo è un paragrafo.

<img src="immagine.jpg">
</body>
```

### Struttura base



- <!DOCTYPE>: informazione per il client riguardante il tipo di documento che si deve aspettare
- <html>: documento HTML
- <head>: intestazione con metadati del documento
- <body>: contenuto del documento

```
k!DOCTYPE html>
<html lang="en">
    (head)
        <!--INTESTAZIONE-->
    </head>
    <body>
        <!--CONTENUTO-->
    </body>
</html>
```

# HTML: esempio

### Titolo (visibile in pagina)

Questo è un paragrafo

Questo è un paragrafo con un link

#### <head>



# Sezione non visualizzata contenente metadati della pagina Web

- <title>
- <meta name="description"/>
- <meta name="keywords"/>
- <meta name="author"/>
- <link>
- .

```
chead
   <title>Titolo del documento</title>
   <meta charset="UTF-8">
   <meta name="description" content="Free Web tutorials">
   <meta name="keywords" content="HTML, CSS, JavaScript">
   <meta name="author" content="John Doe">
   k rel="stylesheet" href="style.css">
 </head>
   <h1>Titolo più grande</h1>
   <h2>Titolo un po' più piccolo</h2>
   <h3>Titolo ancora più piccolo(/h3>
   Questo è un paragrafo.
   <a href="https://www.example.org/">Questo è un link</a>
   <img src="immagine.jpg" alt="Un'immagine.">
   Elemento di una lista non ordinata
   Elemento di una lista non ordinata
   Elemento di una lista ordinata(/li>
   Elemento di una lista ordinata
   (/01>
 </body>
</html>
```

### <body>

Sezione visualizzata della pagina Web con il contenuto vero e proprio

- <h1>, <h2>, <h3>, <h4>, <h5>,<h6>
- <a>>
- <img>
- , <<ul>
- <|i><
- .,

```
<title>Titolo del documento</title>
 <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
 <meta name="keywords" content="HTML, CSS, JavaScript">
 <meta name="author" content="John Doe">
  k rel="stylesheet" href="style.css">
</head>
<body>
 <h1>Titolo più grande</h1>
 <h2>Titolo un po' più piccolo</h2>
  <h3>Titolo ancora più piccolo</h3>
  Questo è un paragrafo.
 <a href="https://www.example.org/">Questo è un link</a>
 (img src="immagine.jpg" alt="Un'immagine.")
 cul>
 Elemento di una lista non ordinata
 Elemento di una lista non ordinata
 (/ul>
 cols
 Elemento di una lista ordinata
 Elemento di una lista ordinata
 (/ol>
</body>
```

#### **Attributi**

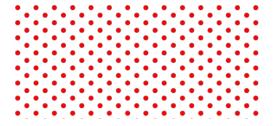


Specificano le funzioni degli elementi, le loro tipologie, memorizzano dati, arricchiscono il significato del contenuto, ecc.

- href: destinazione del collegamento
- **src**: fonte della risorsa
- title: testo aggiuntivo a comparsa
- **alt**: testo alternativo

```
<a href="https://www.example.org/" title="Link di esempio">Questo è un link</a>
<img src="immagine.jpg" alt="Testo alternativo di un'immagine" title="Immagine di esempio">
```

### Esempio: immagini



<img src="https://upload.wikimedia.org/wikipedia/common
s/thumb/6/64/Piet\_Mondrian%2C\_1908-</pre>

10%2C\_Evening%3B\_Red\_Tree\_%28Avond%3B\_De\_rode\_boom%29%2 C\_oil\_on\_canvas%2C\_70\_x\_99\_cm%2C\_Gemeentemuseum\_Den\_Haa g.jpg/1280px-Piet\_Mondrian%2C\_1908-

10%2C\_Evening%3B\_Red\_Tree\_%28Avond%3B\_De\_rode\_boom%29%2 C\_oil\_on\_canvas%2C\_70\_x\_99\_cm%2C\_Gemeentemuseum\_Den\_Haa g.jpg" alt="Testo alternativo di un'immagine" title="Immagine di

esempio" width="380px" height="200px">







### Esempio: link

- Link interno: # + identificativo di un altro elemento nella stessa pagina
- Link esterno: URL completo della pagina a cui il collegamento rimanda

<a href="#titolo1">Questo è un link
interno che manda a un titolo nella
stessa pagina</a>

<a href="https://www.example.org/">Qu esto è un link esterno che manda ad un'altra pagina</a>



Questo è un link interno che manda a un titolo nella stessa pagina

Questo è un link esterno che manda ad un'altra pagina

# 3.2.4 Tecnologie di rete

TCP IP DNS

### Come fa il client a trovare il server?



URL e HTTP forniscono le istruzioni necessarie ad ottenere una rappresentazione della risorsa cercata (=copia della pagina HTML che visualizziamo)

- 1. Il client va a cercare uno degli indirizzi IP del dominio usando il DNS
- 2. Il client richiede al server / <percorso>? <parametri># <ancora>
- 3. Il server restituisce un messaggio con la copia della risorsa richiesta

http://<dominio>/<percorso>?<parametri>#<ancora>

### Cos'è un indirizzo Internet Protocol?

Pur usando l'URL per accedere ad una risorsa, in realtà il server web che la ospita non è direttamente raggiungibile usando il suo nome ma attraverso un indirizzo specifico



Un indirizzo IP è un identificativo numerico assegnato a ogni dispositivo connesso a Internet



### Il tuo IP è 154.47.16.57

Operatore	Datacamp Limited
Posizione	111411, Bogotá, Bogota D.C., Colombia (CO)
Hostname	unn-154-47-16-57.datapacket.com
Browser	Chrome 124.0.0.0
Sistema Operativo	Windows
Mobile	No

### Cos'è un indirizzo Internet Protocol?



Una serie di numeri separati da punti, utilizzata per instradare il traffico di rete tra i dispositivi



Essenziali per la comunicazione su Internet, poiché consentono ai dispositivi di identificarsi e di scambiarsi informazioni tra di loro

### Il tuo IP è 154.47.16.57

Operatore	Datacamp Limited
Posizione	111411, Bogotá, Bogota D.C., Colombia (CO)
Hostname	unn-154-47-16-57.datapacket.com
Browser	Chrome 124.0.0.0
Sistema Operativo	Windows
Mobile	No



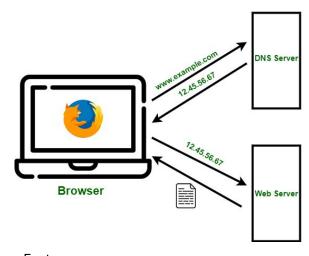
# Cos'è il Domain Name System?



Quando un utente inserisce un URL o clicca su un link, il browser deve scoprire l'indirizzo IP associato a quel dominio per stabilire una connessione

Sistema di nomenclatura utilizzato per tradurre i nomi di dominio degli indirizzi Web in indirizzi IP numerici



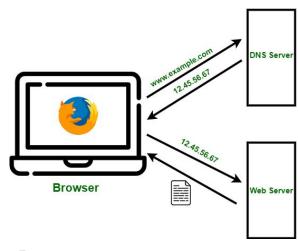


Fonte: https://www.geeksforgeeks.org/working-of-domain-name-system-dns-server/

# Cos'è il Domain Name System?

~ elenco telefonico per Internet, dove i nomi di dominio come "google.com" vengono tradotti in indirizzi IP numerici come "172.217.168.78" che i computer possono usare per connettersi ai server Web corrispondenti

Trovare i siti web desiderati utilizzando nomi di dominio, invece di usare gli indirizzi IP numerici



Fonte: <a href="https://www.geeksforgeeks.org/working-of-d">https://www.geeksforgeeks.org/working-of-d</a> <a href="https://www.geeksforgeeks.org/working-of-d">omain-name-system-dns-server/</a>

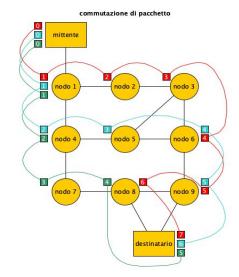
### Cos'è il Transmission Control Protocol?



Protocollo che garantisce una trasmissione affidabile dei dati attraverso una rete

#### Commutazione di pacchetto

Suddivisione dei dati in pacchetti, aggiunta di un'intestazione che contiene informazioni sui dati, e garanzia che i pacchetti siano consegnati senza errori e nell'ordine corretto



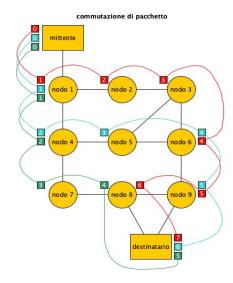
#### Fonte:

https://basic-inf.github.io/2022-2 023/chapters/06.pdf

### Cos'è il Transmission Control Protocol?

→ gli indirizzi IP vengono utilizzati per instradare i pacchetti di dati attraverso la rete da un dispositivo all'altro

Insieme, TCP e IP costituiscono il **protocollo TCP/IP**, che è diventato lo standard de facto per la comunicazione di rete su Internet



#### Fonte:

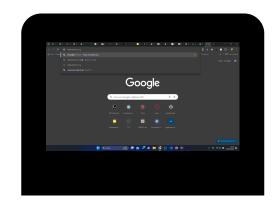
https://basic-inf.github.io/2022-2 023/chapters/06.pdf



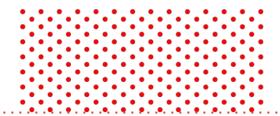
Il Web ~ una strada; ad un'estremità c' è il client (casa), all'altra il server (un negozio)

Inserisci un URL / clicchi su un link durante la navigazione tramite browser (~ cammini fino al negozio)









Il browser si collega al server DNS e trova l'indirizzo reale del server su cui risiede il sito web (~ trovi l'indirizzo del negozio)



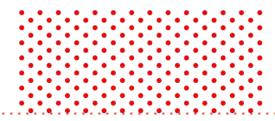




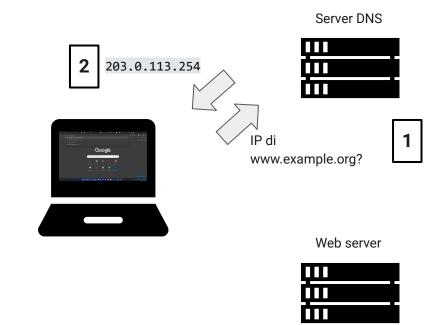
Web server

Server DNS



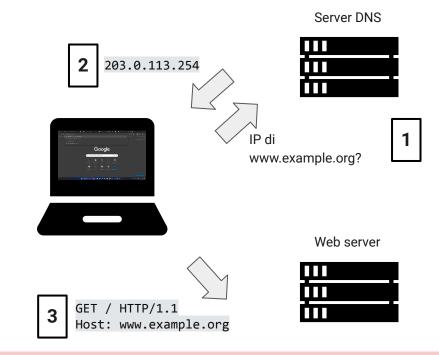


Il browser si collega al server DNS e trova l'indirizzo reale del server su cui risiede il sito web (~ trovi l'indirizzo del negozio)

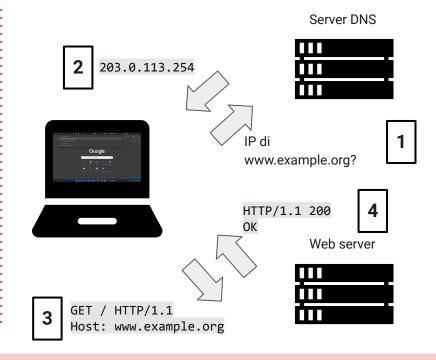




Il browser invia un messaggio di richiesta HTTP al server, chiedendogli di inviare una copia del sito web al client (~ andare al negozio e ordinare i tuoi prodotti)

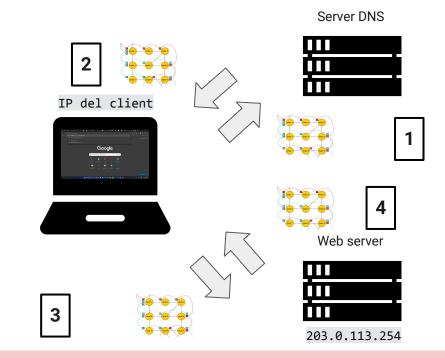


Se il server approva la richiesta del client, invia al client un messaggio "200 OK", che significa "Certo, puoi guardare quel sito web! Eccolo qui", e inizia quindi a inviare i file del sito web al browser sotto forma di pacchetti di dati (~ il negozio ti consegna i tuoi beni, e li riporti a casa)



utilizzando TCP/IP

Questo messaggio e tutti i dati inviati tra il client e il server vengono trasmessi attraverso la tua connessione Internet



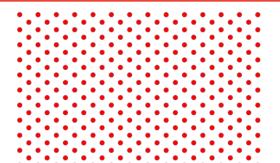


Il browser assembla i pacchetti in una pagina web completa e la visualizza a schermo (~ i prodotti arrivano alla tua porta)





# Quiz 3



https://forms.gle/pVgLYnisLf AUTCem8



# Abilità informatiche

A.A. 2023/2024

03b - Fine

### Sebastian Barzaghi

<u>sebastian.barzaghi2@unibo.it</u> <u>https://orcid.org/0000-0002-0799-1527</u>