

# Contents

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Storia . . . . .	2
1.2	Gli ipertesti . . . . .	2
1.3	WWW come sistema ipertestuale . . . . .	3
1.4	Elementi del Web . . . . .	3
<b>2</b>	<b>URI e URL</b>	<b>3</b>
2.1	URI . . . . .	3
2.1.1	Problematiche fondamentali . . . . .	3
2.1.2	Cosa sono . . . . .	4
2.1.3	Sintassi . . . . .	4
2.2	Specializzazioni di URI: URN e URL . . . . .	4
2.3	URN . . . . .	4
2.4	URL . . . . .	4
2.4.1	Compobenti di un URL . . . . .	4
2.5	URI opache e gerarchiche . . . . .	5
2.5.1	Operazioni sulle URI gerarchiche . . . . .	5

# Tecnologie Web T

Giuseppe Bumma

September 25, 2023

## 1 Introduzione

Il World Wide Web (WWW) è stato proposto nel 1989 da Tim Berners-Lee, ricercatore di fisica al CERN di Ginevra. L'idea alla base del progetto era quella di fornire strumenti adatti a condividere:

- documenti statici
- in forma ipertestuale
- disponibili su rete Internet tramite protocollo semplice e leggero

Si volevano rimpiazzare i sistemi di condivisione di documenti basati su protocolli più vecchi come FTP e Gopher.

### 1.1 Storia

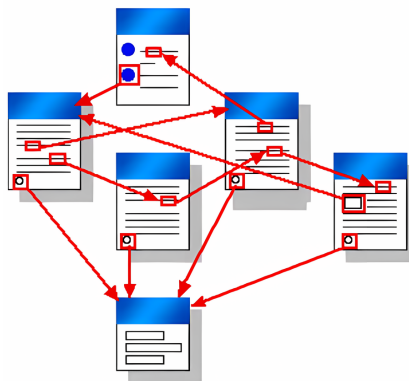
- Nel marzo del 1989 Tim Berners-Lee elaborò una proposta
- Il 12 novembre 1990 assieme a Robert Cailliau presentò una proposta più formale per un sistema ipertestuale basato su un'architettura client-server
- Il 6 agosto 1991 Berners-Lee mise on-line su Internet il primo sito Web. Inizialmente fu utilizzato solo dalla comunità scientifica
- Il 30 aprile 1993 il CERN decise di rendere pubblica la tecnologia alla base del Web

L'HTTP (**HyperText Transfer Protocol**) è un protocollo che sta alla base dell'internet e che permette lo scambio di informazioni in architetture client-server. Ci sono state varie implementazioni del protocollo HTTP

- HTTP/1.0, implementato nel 1991 e proposto come Request for Comment RFC 1945 a Internet Engineering Task Force IETF nel 1996
- HTTP/1.1, presentato come RFC 2068 nel 1997 e aggiornato/approvato nel 1999 come RFC 2616
- HTTP/2 (origin. chiamato HTTP/2.0), basato su SPDY, sviluppato dal Working Group Hypertext Transfer Protocol (httpbis) di IETF
- HTTP/2 pubblicato come RFC 7540 a Maggio 2015, 63% circa del traffico secondo le ultime statistiche

### 1.2 Gli ipertesti

Un **ipertesto** (hypertext) è un insieme di documenti messi in relazione tra loro tramite collegamenti monodirezionali (hyperlink o più semplicemente link). Può essere visto come una rete (un grafo) e i documenti ne costituiscono i nodi.



Attraverso un link possiamo passare da un punto di un documento ad un altro qualunque dei documenti del grafo. La caratteristica principale di un ipertesto è che la lettura può svolgersi in maniera non lineare: qualsiasi documento della rete può essere il successivo.

Se si prendono in considerazione non solo testi ma elementi multimediali (immagini suoni, video) si parla di ipermedia.

### 1.3 WWW come sistema ipertestuale

Idea (e motivazione di successo) di Berners-Lee è stata quella di mettere insieme le idee di ipertesto e rete Internet in modo efficace. World Wide Web è in pratica un ipertesto distribuito sulla rete in cui i documenti, chiamati anche pagine, risiedono su server geograficamente distribuiti (World Wide) e costituiscono una ragnatela virtuale (Web). Da un qualunque documento è possibile “saltare” ad un altro indipendentemente da dove questo si trovi.

Per realizzare questo ipertesto planetario abbiamo bisogno di tre elementi concettuali:

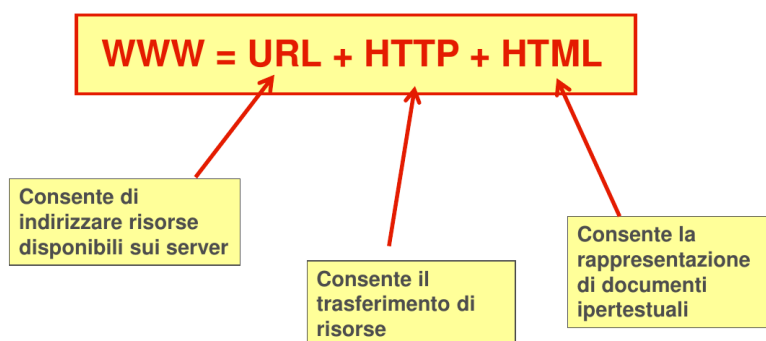
- un meccanismo per localizzare un documento
- un protocollo per accedere alle risorse che costituiscono il documento e trasferirle al client
- un linguaggio per descrivere i documenti ipertestuali (usato per costruire le pagine)

e di due elementi fisici:

- un server in grado di erogare le risorse che costituiscono i documenti
- Un client in grado di rappresentare/visualizzare i documenti e di consentire la navigazione da un documento all'altro

### 1.4 Elementi del Web

In estrema sintesi nella sua visione iniziale il Web può essere rappresentato con la “formula”:



Il Web segue un modello Client/Server: i client **attivi**, detti Web Browser utilizzano il protocollo HTTP per connettersi ai server; i server **passivi**, detti Web o HTTP Server rimangono in ascolto di eventuali connessioni di nuovi client, utilizzano il protocollo HTTP per interagire con i client e forniscono ai client le pagine Web che questi richiedono.

## 2 URI e URL

### 2.1 URI

#### 2.1.1 Problematiche fondamentali

L'URL (primo componente della formula del Web) fa riferimento a tre questioni principali:

- come identificare il server in grado di fornirci un elemento dell'ipertesto
- come identifichiamo la risorsa a cui vogliamo accedere?
- quali meccanismi (ad es. in termini di protocollo) possiamo utilizzare per accedere alla risorsa?

La risposta a tutte queste domande sono gli URI.

### 2.1.2 Cosa sono

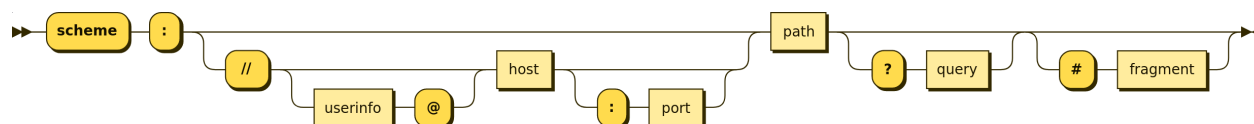
Gli URI (Uniform Resource Identifier) forniscono un meccanismo semplice ed estensibile per identificare una risorsa. Con il termine risorsa intendiamo qualunque entità abbia una identità: un documento, un'immagine, un servizio, una collezione di altre risorse.

Un URI è un **concetto generale**, quindi non fa riferimento solo a risorse alle quali si può accedere con protocollo HTTP o a entità disponibili in rete. Si dice che è un **mapping** concettuale a una identità, e non si riferisce a una particolare versione dell'entità esistente in un dato momento. Il mapping può rimanere inalterato anche se cambia il contenuto della risorsa.

### 2.1.3 Sintassi

Gli URI rispettano una sintassi standard, semplice e regolare:

URI = scheme ":" ["/" authority] path ["?" query] ["#" fragment]



## 2.2 Specializzazioni di URI: URN e URL

Esistono due specializzazioni del concetto di URI:

- Uniform Resource Name (URN): identifica una risorsa per mezzo di un “nome” che deve essere globalmente unico e **restare valido anche se la risorsa diventa non disponibile** o cessa di esistere
- Uniform Resource Locator (URL): identifica una risorsa per mezzo del suo **meccanismo di accesso primario** (es. locazione nella rete) piuttosto che sulla base del suo nome o dei suoi attributi

Applicando questi concetti ad una persona: l'URN è come identificazione basata su nome+cognome, o meglio codice fiscale, quindi io conosco quella persona ma non so dov'è, l'URL è come indirizzo di casa o numero di telefono (se univoci).

### 2.3 URN

Un URN identifica una risorsa mediante un nome in un particolare dominio di nomi (namespace), quindi consente di “parlare” di una risorsa prescindendo dalla sua ubicazione e dalle modalità con cui è possibile accedervi.

Un esempio molto noto è il codice ISBN (International Standard Book Number) che identifica a livello internazionale in modo univoco e duraturo un libro o una dizione di un libro di un determinato editore.

### 2.4 URL

Un URL tiene conto anche della modalità per accedere alla risorsa, infatti specifica il protocollo necessario per il trasferimento della risorsa stessa (tipicamente il nome dello schema corrisponde al protocollo utilizzato).

Nella sua forma più comune (schema HTTP-like) sintassi è

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

Questa forma vale per diversi protocolli di uso comune come HTTP, HTTPS, FTP, WAP, etc ...

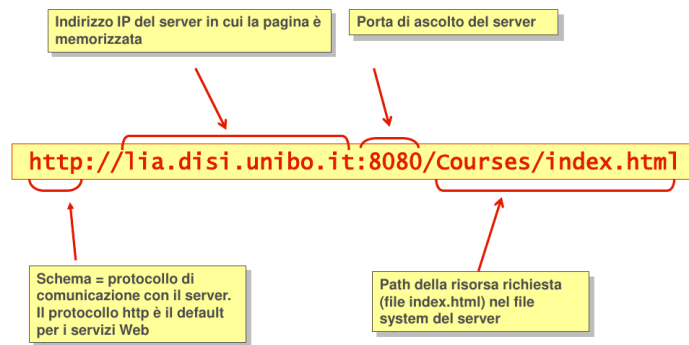
#### 2.4.1 Componenti di un URL

Vediamo ora le componenti di un indirizzo URL nel dettaglio:

- <protocol>: descrive il protocollo da utilizzare per l'accesso al server
- <username>:<password>@: credenziale per l'autenticazione (oggi giorno non si usano mai perché non è sicuro inserirle nell'URL)
- <host>: indirizzo server su cui risiede la risorsa; può essere un indirizzo logico o fisico
- <path>: percorso (*pathname*) che identifica la risorsa nel file system del server; se manca, tipicamente si accede alla home page

- **<query>**: una stringa di caratteri che consente di passare al server uno o più parametri; di solito ha questo formato:  
`parametro1=valore&parametro2=valore2...`

Esempio di URL con schema HTTP:



## 2.5 URI opache e gerarchiche

Le URI possono essere anche classificate come opache o gerarchiche

- l'URI opaca non è soggetta a ulteriori operazioni di *parsing*; esempio

`mailto:paolo.rossi@disi.unibo.it`

- L'URI gerarchica è soggetta a ulteriori operazioni di parsing, per esempio per separare l'indirizzo del server dal percorso all'interno del file system; alcuni esempi:

`http://informatica.unibo.it/`

qui non è specificato un path, quindi il server ritorna la home page identificata dal file `index.html`

`docs/guide/collections/designfaq.html#28`  
`../../../../lab/examples/ant/build.xml`

i `".."` indicano il server che sto visitando in quel momento

`file:///~/calendar`

### 2.5.1 Operazioni sulle URI gerarchiche

Di seguito si elencano le operazioni possibili su URI gerarchiche:

- **Normalizzazione**: processo di rimozione dei segmenti `"."` e `".."` (e altri caratteri speciali) dal path (N.B. la normalizzazione si applica solo a URI gerarchiche, su URI opache non ha effetto)
- **Risoluzione**: processo che a partire da una URI originaria porta all'ottenimento di una URI risultante; la URI originaria viene risolta basandosi sulla base URI
- **Relativizzazione**: processo inverso alla risoluzione

Esempio di risoluzione di una URI:

URI originaria:

`docs/guide/collections/designfaq.html#28`

Base URI:

`http://disi.unibo.it/`

Risultato:

`http://disi.unibo.it/docs/guide/collections/designfaq.html#28`