

tecnologie web

Matteo

January 25, 2026

1 uri

URi (uniform resource identifier): stringa di caratteri che identifica in modo univoco una risorsa su internet esistono due tipi di URI:

- URL (uniform resource locator): indica l'indirizzo di una risorsa su internet
- URN (uniform resource name): identifica una risorsa in modo univoco senza fornire il suo indirizzo

gli uRL sono piu immediati da usare ma piu soggetti a cambiamenti gli uRN sono piu stabili ma meno intuitivi

Nella visione moderna, la distinzione tra locator e name è secondaria rispetto al concetto di schemi: ogni URI appartiene ad uno schema (la parte della stringa che precede i due punti).

gli uri devono essere:

- trascrivibili
- Fornire identificazione, non interazione
- essere organizzati gerarchicamente

`{URI=schema://authority/path?query#fragment}`

schema = protocollo tcp usato.

l'authority è una parte specifica dell'URI che identifica chi controlla o ospita la risorsa.

La parte path è la parte identificativa della risorsa all'interno dello spazio di nomi identificato dallo schema e (se esistente) dalla authority.

query: fornisce parametri aggiuntivi per identificare o interagire con la risorsa.

fragment: identifica una parte specifica della risorsa o un punto di interesse all'interno della risorsa stessa.

i caratteri degli uri possono essere:

- caratteri riservati: hanno un significato speciale all'interno dell'URI
- caratteri non riservati: possono essere usati liberamente
- caratteri escaped: usati per rappresentare caratteri speciali o non consentiti

Una **route** è un'associazione della parte path di un URI ad una risorsa gestita o restituita da un server web.

esiste managed route: il server associa ogni URI ad una risorsa o attraverso il file system locale (risorse statiche) oppure generate attraverso una computazione (risorse dinamiche)

File-system route: il server associa la radice della parte path ad una directory del file system locale e ogni filename valido all'interno di quella directory genera un URI corretto e funzionante.

un uri assoluto contiene tutte le informazioni necessarie per localizzare una risorsa, mentre un uri relativo(uri reference) fornisce un percorso in relazione a un altro URI di base.

risoluzione degli URI relativi: il processo di conversione di un URI relativo in un URI assoluto utilizzando un URI di base come riferimento.

esistono diversi schemi

http e https di gran lunga i più usati schema file: per accedere a file locali schema data: per incorporare dati direttamente all'interno di un URI ftp: per trasferire file tra computer su una rete

2 codifica caratteri

esistenza di tante lingue diverse crea difficoltà

per rappresentare i caratteri servono regole: ordine, contiguità, raggruppamento in gruppi logici

gli **shift** servono per cambiare il set di caratteri in uso, i codici liberi sono errori di trasmissione, poi ci sono i codici di controllo.

ascii definisce 128 caratteri, usa 8 bit per rappresentare ogni carattere, il primo bit è di controllo, per questo sono 128 e non 256.

ci sono inoltre 33 caratteri di controllo non stampabili.

code page: estensioni di ASCII per supportare caratteri aggiuntivi. così vengono aggiunte le altre lingue.

negli anni 90 due commissioni diverse hanno cercato di creare uno standard universale per la codifica dei caratteri: iso/iec e unicode, ora sono stati unificati.

unicode contiene ora tutti i caratteri di tutti gli alfabeti, si cerca di essere efficienti e che ogni carattere abbia uno scopo.

inoltre quando possibile si cerca di usare caratteri preesistenti per evitare problemi di compatibilità.

iso ha 2 schemi di codifica: ucs 2 e ucs 4 ucs 2 usa 2 byte per carattere, ucs 4 usa 4 byte per carattere. visto che ucs 4 ha 4 miliardi di combinazioni, fu inventato utf-8: usa da 1 a 4 byte per carattere, e è compatibile con ascii.

i sistemi big endian e little endian gestiscono l'ordine dei byte in modo diverso. quindi per evitare ambiguità si usa il BOM(byte order mark) all'inizio del file. Zero-Width No-Break Space (ZWNBS), un carattere che può essere usato in qualunque contesto di whitespace (cioè ovunque tranne in mezzo alle parole) senza modificare il significato dei testi, questo indica l'ordine dei byte.

3 markup

i dati esistono come valori isolati(25), come coppie chiave valore o etichette(eta:25), e come record, o raccolte di etichette(nome:matteo, eta:25, ecc...) poi li puoi rappresentare in molti modi: alberi, tabelle, liste ecc....

Testi sono molto soggettivi La teoria del markup descrive i testi come un albero ordinato ed etichettato di elementi e nodi di testo.

HTML:Attuale formato di punta per i documenti di testo sul Web Il testo è organizzato in elementi racchiusi all'interno di tag. C'è un elenco di tag che costituisce il vocabolario HTML. Utilizza linguaggi aggiuntivi (ad esempio, CSS e Javascript per tipografia, layout e interattività sofisticati

XML piu incentrato alla struttura dei dati rispetto ad HTML Permette di definire nuovi tag e strutture di dati personalizzate.

esiste il formato binario per rappresentare i dati in modo piu efficiente, non e leggibile dall'uomo. il formato leggibile si invece.

il parsing e il processo di conversione di un documento in un albero di oggetti in memoria.

Il markup interno inserisce istruzioni di presentazione all'interno del testo, in mezzo alle parole. Il markup esterno prevede due blocchi di informazioni: il contenuto e il markup, separati e collegati da indirizione.

esistono anche diversi scopi di markup:

- presentazione: come visualizzare il testo
- puntazione: struttura logica del testo
- procedurale: istruzione per gli effetti.
- descrittivo: informazioni sui dati a livello strutturale
- referenziale: collegamenti tra parti del testo o tra testi diversi
- metamarkup: informazioni sul markup stesso

3.1 linguaggi di markup

TROFF/NROFF: uno dei primi linguaggi di markup, usato per la stampa di documenti tecnici.

TEX: complesso ma permette macro. poi vi si ricavo LaTeX, piu semplice.

lingauggi wiki: usato per creare pagine web in modo semplice.

markdown: linguaggio di markup leggero, facile da leggere e scrivere, usato per documentazione e blog.

json: formato di dati leggero e facile da leggere, usato per lo scambio di dati tra applicazioni web.

yaml: superset di json, piu leggibile per gli umani, usato per file di configurazione.

sgml: standard per definire linguaggi di markup personalizzati, usato in applicazioni specifiche. sgml percmette di definire una grammatica per un linguaggio di markup.

in un documento sgml ci sono 3 parti:

- declaration: definisce la grammatica del documento

- doctype: specifica il tipo di documento e la sua struttura
- document instance: il contenuto effettivo del documento

XML è un sottoinsieme di SGML, ma più semplice e più facile da implementare. come sgml XML è composto da:

- elementi: unità di base del documento, racchiuse tra tag di apertura e chiusura
- attributi: forniscono informazioni aggiuntive sugli elementi
- entità: rappresentano caratteri speciali o sequenze di caratteri
- PCDATA: testo contenuto all'interno degli elementi
- commenti: note o spiegazioni all'interno del documento
- processing instructions: istruzioni per il processore XML

I documenti XML possono essere validi o ben formati. un documento è ben formato se segue le regole sintattiche di XML anche senza essere valido. un documento è valido se segue uno schema o una definizione di tipo di documento (DTD).

4 HTML

HTML ora diventato motore di presentazione di applicazioni web complesse.

c'è inoltre ancora confusione tra le varie versioni

In HTML 4.01 e XHTML 1.0, ci sono molte pagine web scritte non conformemente agli standard ma comunque leggibili altre non funzionano. quindi ci sono due modalità di rendering:

- modalità strict: il browser cerca di seguire gli standard il più possibile
- modalità quirks: il browser cerca di emulare i comportamenti non standard

HTML5: versione più recente di HTML, introduce nuovi elementi e attributi per migliorare la semantica e l'accessibilità dei documenti web. ora è il living standard cioè in continuo sviluppo.

Gli elementi **inline** non iniziano su una nuova riga, possono essere **fontstyle** (forniscono informazioni di rendering) o **phrase**

poi ci sono gli elementi **block-level** o tag di blocco che iniziano su una nuova riga e occupano tutta la riga.

ci sono poi le liste

e poi ci sono gli elementi generici come **div** e **span** che non hanno significato semantico ma sono usati per raggruppare altri elementi e fornirgli caratteristiche dettate da altri attributi.

<header> : rappresenta l'intestazione di una sezione o di un documento.

<footer> : rappresenta il piè di pagina di una sezione o di un documento.

HTML 5 riprende da XHTML 2.0 anche le "liste di navigazione" ossia particolari sezioni dedicate a raggruppare link alla pagina corrente (o a sezioni di) o ad altre pagine. Si usa l'elemento **<nav>** molto spesso in combinazione con **<header>** e **<footer>**.

<a href> o **<a name>** per creare collegamenti ipertestuali.

immagini con ``.

Attraverso l'attributo `srcset` è possibile indicare più risorse per la stessa immagine, da usare alternativamente. `<figure>` e `<figcaption>` per raggruppare immagini e didascalie.

tag di embedding come `<audio>`, `<video>`, e `<canvas>` per inserire contenuti multimediali e grafici interattivi.

Con i FORM si utilizzano le pagine HTML per inserire valori che vengono poi elaborati sul server. I FORM sono legati ad applicazioni server-side

si usa `<form +attributi>` per creare un modulo HTML. cio permette all'utente di interagire.

ci sono diversi tipi di attributi associati all'input:

- `required`: campo obbligatorio
- `placeholder`: testo segnaposto
- `readonly`: campo di sola lettura
- `list`: l'input deve essere uno dei valori in una lista predefinita.

Sono attributi globali quelli definiti su tutti gli elementi del linguaggio HTML. ad esempio `style`, o `id`

attributi data-: Attributi personalizzati che possono essere utilizzati da applicazioni e script Javascript senza inquinare lo spazio dei nomi.

attributi ARIA: Attributi che migliorano l'accessibilità dei contenuti web per utenti con disabilità.

colori si possono definire tramite codice RGB o nome del colore.

si usano tre tipi di lunghezza : pixel, percentuali(rispetto al contenitore), multi-lunghezze.

HTML non tiene in conto maiuscole e minuscole nei tag e negli attributi, XHTML sì.

in HTML i whitespace multipli vengono ridotti a uno solo, in XHTML vengono preservati.

i tag di head:

- `<title>`: specifica il titolo del documento
- `<meta>`: fornisce metadati come charset,
- `<link>`: collega risorse esterne come fogli di stile
- `<style>`: contiene regole CSS interne
- `<script>`: include codice JavaScript
- `<base>`: specifica l'URL di base per i collegamenti relativi

elementi come `canvas`, `video` e `audio` permettono di creare contenuti multimediali interattivi direttamente nelle pagine web senza bisogno di plugin esterni. le interazioni possono essere anche complesse, come giochi o applicazioni grafiche.

`canvas` ad esempio permette di disegnare direttamente sulla pagina.

`audio` e `video` permettono di caricare video e audio in diversi formati.

i web components permettono di creare elementi HTML personalizzati riutilizzabili con funzionalità incapsulate.

5 CSS

CSS (Cascading Style Sheets) è un linguaggio di stile usato per descrivere la presentazione di un documento scritto in HTML o XML. La parola chiave è cascading: è prevista ed incoraggiata la presenza di fogli di stile multipli, che agiscono uno dopo l'altro, in cascata, per indicare le caratteristiche tipografiche e di layout di un documento HTML

il sistema a cascata stabilisce l'ordine di priorità tra più regole di stile applicate allo stesso elemento contemporaneamente.

HTML prevede l'uso di CSS in tre modi:

- inline: direttamente nell'elemento HTML tramite l'attributo style
- internal: all'interno di un elemento `<style>` nella sezione `<head>`
- external: tramite un file CSS esterno collegato con l'elemento `<link>`

innoltre uno stile può essere applicato a:

- elementi specifici
- tutti gli elementi di un certo tipo
- elementi con una certa classe o ID

id assume un valore univoco su tutto il documento, in modo da identificare quello specifico elemento tra tutti gli altri

class assume un valore qualunque e può essere usato per raggruppare elementi simili.

una proprietà è un attributo di stile specifico che può essere applicato a un elemento HTML, come colore, dimensione del carattere, margini, ecc.

uno statment indica una proprietà e il suo valore, ad esempio `color: red;`.

un selettore è una parte di codice CSS che specifica a quali elementi HTML si applicano le regole di stile definite.

una regola CSS è composta da un selettore e un blocco di dichiarazioni che definiscono le proprietà di stile da applicare agli elementi selezionati.

alcune lunghezze sono assolute, altre relative.

il **device pixel** è l'unità di misura più usata per il web. è la minima unità di misura visualizzabile su uno schermo(dipende anche dalla densità dello schermo).

La visualizzazione di un documento con CSS avviene identificando lo spazio di visualizzazione di ciascun elemento.

ogni elemento ha un box rettangolare che ne definisce lo spazio occupato.

il box model è creato in relazione alle altre ed è definito da **flusso** e **posizione**.

i flussi sono gestiti implicitamente o dalla proprietà di display.

la scatola(box) è composta da:

- margin: spazio esterno al bordo che separa le scatole le une dalle altre.
- border: bordo della scatola
- padding: spazio interno tra il bordo e il contenuto
- content: area che contiene il contenuto effettivo dell'elemento

queste parti della scatola possono essere modificate tramite css.

è possibile annidare i selettori dentro un contenitore con &

Il canvas è l'area virtuale di posizionamento degli elementi del DOM via CSS. E' un piano cartesiano infinito in larghezza e altezza, con l'asse delle x da sinistra a destra e l'asse delle y dall'alto in basso

Il viewport è la parte del canvas che è attualmente visibile attraverso lo schermo o la finestra e può muoversi sul canvas in seguito a scrolling. pixel è l'unità di misura di riferimento delle canvas. esistono pixel la cui dimensione varia in base alla densità dello schermo(device pixel) e pixel la cui misura è fissata indipendentemente dalla densità(css pixel).

viewport width e viewport height: larghezza e altezza della finestra di visualizzazione attuale misurate con la percentuale dell'area visibile.

Una lunghezza flessibile (flex) è una dimensione con unità fr che rappresenta una frazione dello spazio rimasto nel contenitore.

diversi tipi di posizionamento:

- statico: posizione predefinita, segue il flusso normale del documento
- relativo: posizione relativa alla sua posizione originale
- assoluto: posizione relativa al primo antenato posizionato
- fisso: posizione relativa alla finestra del browser, rimane fissa durante lo scrolling
- sticky: si comporta come relativo fino a quando non raggiunge una certa posizione, poi si comporta come fisso

zindex: proprietà che determina l'ordine di sovrapposizione degli elementi posizionati. Un elemento con un valore di **zindex** più alto sarà visualizzato sopra gli elementi con valori più bassi.

overflow: proprietà che controlla come gestire il contenuto che supera le dimensioni del suo contenitore. Può essere impostato su valori come visible, hidden, scroll o auto.

La proprietà display gestisce la natura e l'organizzazione della scatola rispetto al contesto (mondo esterno) e contenuto (mondo interno)

Il layout è l'organizzazione spaziale delle componenti strutturali più importanti di una pagina web. raramente viene usato il posizionamento naturale, sono state inventate tecniche di layout più complesse.

<http://www.fabioitali.it/TW/2025/layout/>

con grid si definisce una griglia bidimensionale per posizionare gli elementi in righe e colonne.

Con display: flex; si può assegnare ad un elemento un'organizzazione visuale a contenuti flessibili e che si distribuiscono armonicamente nello spazio disponibile.

ereditarietà: molte proprietà CSS vengono ereditate dagli elementi figli dai loro genitori, facilitando la gestione degli stili comuni.

La keyword `!important` può essere aggiunta in fondo a qualunque statement e aumenta la priorità dello statement anche rispetto a statement successivi attivabili in cascata

L'ordine con cui vengono considerate le varie regole non è dato solo dall'ordine in cui sono posti nei fogli di stile. Infatti, viene applicato un algoritmo di ordinamento sulle dichiarazioni secondo alcuni principi (dal più al meno importante):

- media-type (ad esempio, schermo vs stampa)
- dichiarazioni con `!important`
- origine di una dichiarazione (utente vs autore vs user-agent)
- la specificità del selettore della dichiarazione
- l'ordine delle dichiarazioni (l'ultima ha la precedenza)

anche la cascata non è ordinata in modo assoluto, ma dipende dal contesto:

1. dichiarazioni dell'user agent
2. dichiarazioni dell'utente
3. dichiarazioni dell'autore
4. dichiarazioni importanti dell'autore (`!important`)
5. dichiarazioni importanti dell'utente (`!important`)

la scatola può essere trasformata una volta generata con `transform: function(parameters)`

Le regole precedute da un `@`, comunemente chiamate *at rules*, servono per specificare ambiti o meta-regole del foglio di stile ad esempio `@import` per importare regole da un altro foglio di stile

Le *media queries* servono per specificare delle regole particolari che vengono attivate nel caso in cui il supporto usato per visualizzare la pagina Web soddisfa particolari vincoli. Tra le varie espressioni utilizzabili, quelle più comuni permettono di realizzare dei vincoli sulla larghezza, altezza e colore supportati dal dispositivo.

Il linguaggio per creare *media queries* è complesso, ed è composto di regole separate, collegate da operatori come `and`, `not`, `only`.

`@keyframes` è un blocco di regole per specificare lo stato iniziale (`from`), lo stato finale (`to`) ed eventuali stati intermedi (percentuali) di una o più proprietà per fare animazioni.

`@layer` organizza le regole CSS in livelli (*layers*) con priorità definite, permettendo di gestire la cascata in modo più strutturato e modulare.

@container permette di applicare stili condizionali basati sulle dimensioni del contenitore di un elemento, piuttosto che sulle dimensioni della finestra del browser, facilitando il design reattivo.

proprietà con valori customizzati iniziano sempre con –

5.1 framework CSS

forniscono librerie di stili predefiniti e componenti riutilizzabili per facilitare lo sviluppo web.

sono molto facili da usare e gestiscono molte differenze tra browser, ma ne risulta un look standardizzato e poco personalizzabile.

twitter bootstrap: uno dei framework più popolari, offre una vasta gamma di componenti e stili predefiniti.

Bootstrap definisce una griglia virtuale divisa in dodicesimi

fornisce anche numerosi effetti grafici come navbar, tabulatori, finestre ecc....

6 semantic web

semantic web: struttura comune che consente di condividere e riutilizzare dati tra applicazioni, aziende e comunità
ciatton link: permette di creare collegamenti semantic web
stack: illustra l'architettura del web semantico
diverse classi sono organizzate in modo tassonomico attraverso un modello chiamato RDF.

RDF (resource description framework): serve a fare affermazioni (statement) sulle risorse nella forma di triple (soggetto-predicato-oggetto)

un grafo RDF è un insieme di triple RDF
le risorse sono rappresentate come nodi.

il modello a triple è semplice e minimalista. il modello RDF inoltre è modulare:

- la gestione delle informazioni può essere parallelizzata
- informazioni parziali sono comunque valide

svantaggi:

- il modello di dati RDF è costituito da elementi di dati piccoli e frammentati, quindi un database di medie dimensioni risulta in miliardi di triple
- limitazione delle relazioni n-arie, non ci sono modi semplici di descriverli
- limitata possibilità di attribuire informazioni alle triple stesse

reificazione: prendo una tripla e gli do un identificativo per farla diventare parte di un'altra tripla

microformati: sono ad esempio embedding di triple RDF all'interno di ambienti ospiti
importante l'aspetto della serializzazione. manca il reasoning e la generazione di nuove informazioni

con RDF si possono introdurre le inferenze attraverso altre classi e proprietà

il modello tabellare è il migliore che abbiamo nel complesso, ma è inefficiente per le relazioni M-N
il modello ad albero ha senso per le relazioni di tipo 1-N
meno per M-N