# tecnologie web

Matteo

January 29, 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
- essre 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 possonoessere:

- 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: l 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 piu usati shema 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 cartatteri

esistenza di tante lingue diverse crea difficolta

per rappresentare i caratteri servono regole: ordine, contigiuta, 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 e 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. cosi 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 efficenti e che ogni carattere aabia uno scopo.

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

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 e compatibile con ascii.

i sistemi big endian e little endian gestiscono l'ordine dei byte in modo diverso. quindi per evitare ambiguita si usa il BOM(byte order mark) all'inizio del file. Zero-Width No-Break Space (ZWNBSP), 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 indirezione.

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 e un sottoinsieme di SGML, ma piu semplice e piu facile da implementare.
come sgml XML ecomposto 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 e ben formato se segue le regole sintattiche di XML anche senza essere vaalido. un documento e 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 modalita di rendering:

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

HTML5: versione piu recente di HTML, introduce nuovi elementi e attributi per migliorare la semantica e l'accessibilità dei documenti web. ora è il living standard cioe 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 `<img + attributo>`.

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.

clori 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 caaricare 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 puo essere usato per raggruppare elementi simili.

una proprieta 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 proprieta e il suo valore, ad esempio `color: red;`.

un selettore e 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** e l'unita di misura piu usata per il web. è la minima unita di misura visualizzabile su uno schermo(dipende anche dalla densita 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 e creato in relazione alle altre ed e definito da **flusso** e **posizione**.

i flussi sono gestiti impplicitamente o dalla proprieta di display.

la scatola(box) e 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 e l'unita 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 e fissata indipendenti 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 a 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 piu complesse.
http://www.fabiovitali.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. dichiarazione dell'user angent
2. dichiarazioni dell'utente
3. dichiarazioni dell'autore
4. dichiarazioni importanti dell'autore(!important
5. dichiarazioni importanti dell'utente(!important)

la scatola puo 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 organizze 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.

proprieta 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 piu 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 javascript

javascript associa eventi a funzioni(attributi) che vengono eseguite quando l'evento(click,mouse-over ecc....) si verifica.

inserendo istruzioni javascript nei valori dell'attributo si creano chiamate callback.

2 eventi particolari **load**(nella window ) e **DOMContentLoaded**(nel document) che indicano il caricamento della pagina.

i servizi serverside sono invece associati a uri.

NOde fornisce il modo di asssociare chiamte callback a uri.

gli script possono essere eseguiti sia in maniera sincrona che asincrona.

in maniera asincrona lo script puo essere sia caricato associato a un evento, al completamento di un'operazione di rete o dopo un timer.

HTML visualizza l'output degli script in 4 modi diversi:

- sulla finestra del browser
- scrivendo sulla console
- in una finestra di alert
- modificando il DOM del doc visualizzato

lo script puo essere inserito in 3 modi:

- direttamente dento l'attributo di un evento
- all'nterno di un tag `<script>`
- in un file esterno puntato dal tag `<script`

javascript e molto flessibile per i tipi di dati utilizzabili, 3 importanti: numeri, booleani, stringhe. importanti anche i null e undefined.

poi ci sono gli **object**: tipi di dati strutturat di cui fanno parte anche gli array.

3 modi di definire una variabile:

- var: scope di funzione
- let: scope di blocco
- const: scope di blocco, non puo essere riassegnata

le funzioni javascript sono blocchi di istruzioni dotati di un nome e facoltativamente di paramentri.

possono ma non sono obbligate a restituire un valore di ritorno. Le funzioni non sono tipate, i valori di ritorno sì (come al solito).

Se manca un parametro, non restituisce errore ma assume che il parametro sia undefined.

gli **object**: sono strutture liste di proprieta(coppie chiave-valore) non ordinate. il valore di una proprieta puo essere un object

si puo accedere alle proprieta di un object in due modi: alert(object.proprieta) o alert(object["proprieta"])

Un array è un object in cui le chiavi sono numeri interi assegnati automaticamente, per distinguerle usi le quadre anziche le graffe.

gli array hanno diverse proprieta uniche come pop o push.

JavaScript predefinisce alcuni oggetti utili per raccogliere insieme i metodi più appropriati per certi tipi di dati. es(object, Array, String, Math, Date, RegExp, ecc...)

l'oggetto string fornisce metodi per manipolare le stringhe di testo.

JSON (JavaScript Object Notation) è un formato dati derivato dalla notazione usata da JS per gli oggetti. Viene usato per lo scambio di dati tra applicazioni web e server.

solo valori string, number, boolean, array o object Anche i nomi delle proprietà sono tra virgolette

Tutti i linguaggi di programmazione più importanti oggi accettano e scrivono dati in JSON

JSON e un singoletto che supporta solo 2 metodi:

- stringify: converte un oggetto JS in una stringa JSON
- parse: converte una stringa JSON in un oggetto JS

Una data è un oggetto che esprime un giorno e un orario rappresentandolo come il numero di millisecondi trascorsi dalla mezzanotte del 1 gennaio 1970 e la data in questione. Poiché in realtà è un numero, questo permette di fare operazioni aritmetiche e confronti numerici.

MATH contiene funzioni matematiche e costanti predefinite.

regexp è un oggetto che rappresenta espressioni regolari, usato per cercare e manipolare stringhe di testo.

**oggetti principali del browser**:

- window: rappresenta la finestra del browser

- navigator: fornisce informazioni sul client
- location: rappresenta l'URL corrente
- history: gestisce la cronologia di navigazione
- document: rappresenta il documento HTML caricato

il document è il più importante, rappresenta la pagina web caricata. ed ha proprietà e metodi per accedere ad ogni elemento della gerarchia (es. title, body, forms, images ecc....)

Inoltre document rappresenta l'oggetto DOMDocument del DOM del documento visualizzato

## 6.1 DOM

il DOM (Document Object Model) è una rappresentazione ad albero della struttura di un documento HTML o XML, che consente agli script di accedere e manipolare il contenuto, la struttura e lo stile del documento in modo dinamico. il modo in cui il DOM viene costruito dal parsing fu deciso dal WHATWG

Il Document Object Model è un interfaccia di programmazione (API) per documenti sia HTML sia XML. Definisce la struttura logica dei documenti ed il modo in cui si accede e si manipola un documento.

Il core del DOM definisce alcune classi fondamentali per i documenti HTML e XML, e ne specifica proprietà e metodi. La classe principale di DOM è DOMNode, di cui la maggior parte delle altre classi è una sottoclasse. le classi principali sono:

- DOMDocument: rappresenta l'intero documento, specifia metodi per accedere agli elementi principali come head e body
- DOMElement: rappresenta un elemento HTML o XML, specifica metodi per accedere agli attributi e ai figli dell'elemento
- DOMAttr: rappresenta un attributo di un elemento
- DOMText: rappresenta il testo all'interno di un elemento

DOMnode sprcifica metodi per navigare e manipolare l'albero del documento, come appendChild, removeChild, getElementsByTagName ecc....

javascript interagisce con il DOM tramite l'oggetto document, che rappresenta il documento caricato nel browser, e fornisce questi metodi per accedere e manipolare gli elementi del DOM.

il DOM per HTML permette di leggere e scrivere interi elementi, innerHTML legge e scrive il contenuto di un sottoalbero esclusi i tag radice.

outerHTML legge e scrive l'intero elemento incluso il tag radice.

il DOM non ha metodi veri e propri per la nvigazione, bisogna usare le proprieta children come(firstChild, lastChild, parentNode, nextSibling, previousSibling ecc....)

ci sono metodii per accedere a nodi specifi come getElementById, getElementsByClassName, getElementsByTagName ecc....

fu poi anche inventato querySelector e querySelectorAll che usano selettori CSS per trovare elementi.

## 6.2 javascript avanzato

I valori undefined, null e NaN sono tre keyword diverse ma concettualmente affini per anomalie del codice.

undefined e il valore predefinito di una variabile non inizializzata. null indica l'assenza intenzionale di un valore. NaN (Not a Number) è un valore speciale che indica un risultato matematico non definito o non rappresentabile.

i valori falsy sono valori che vengono considerati false in un contesto booleano. I valori falsy in JavaScript sono: false, 0, meno 0, 0n (BigInt zero), "", null, undefined e NaN. tutti gli altri sono chiamati truthy.

—— e `&&` non fanno casting ma valutano direttamente i valori.

In pratica, moltissimi programmatori lo usano come verifica della presenza e istanziazione di una variabile o la disponibilità di una libreria o un servizio.

in javascript le funzioni sono oggetti quasi come tutte le altre varibili, possonon essere assegnate a variabili o passate come parametri di funzione. si possono anche restituire funzioni da altre funzioni. o passare funzioni anonime.

molte funzioni specifiche per gli array come array.sort o array.filter.

javascript è object oriented In un linguaggio object oriented tradizionale, la classe è un template sulla base del quale vengono istanziati gli oggetti del programma, specificando i membri (stati dell'oggetto, valori) e i metodi (comportamenti dell'oggetto, funzioni).

**this**: parola chiave che si riferisce all'oggetto corrente in cui il codice è in esecuzione, che sia una funzione, un metodo o un costruttore.

i linguaggi object-oriented sono divisi in due categorie principali: class-based e prototype-based.

javascript e prototype based: gli oggetti possono ereditare direttamente da altri oggetti tramite il prototipo, senza la necessità di definire classi.

Ogni oggetto in JavaScript è autonomo e si possono aggiungere tutti i metodi/proprietà che si vuole senza modificare gli altri.

Per aggiungere proprietà/metodi condivisi da molti oggetti debbo usare l'oggetto prototype.

javascript ha 4 tipi di scope:

- globale: variabili dichiarate fuori da qualsiasi funzione
- di funzione: variabili dichiarate all'interno di una funzione
- di blocco: variabili dichiarate con let o const all'interno di un blocco
- di modulo: variabili dichiarate all'interno di un modulo

JavaScript non ha protezione dei membri privati di un oggetto, ma sono tutti accessibili e manipolabili.

C'è un quinto scope, detto closure, che è lo scope della funzione all'interno della quale viene definita un'altra funzione.

Ad esempio, una funzione che restituisce una funzione ha uno scope che è sempre accessibile alla funzione interna, ma non dal mondo esterno. è usato per creare variabili private.

Una function expression immediatamente invocata (IIFE) è una funzione anonima creata ed immediatamente invocata. serve per creare oggetti privati non ripetibili.

## 6.3 DOM avanzato

Xpath è un linguaggio di query per selezionare nodi in un documento XML o HTML basato sulla struttura ad albero del DOM.

Gli assi identificano la direzione rispetto alla struttura del documento in cui andare a cercare l'oggetto da restituire rispetto al nodo contesto (NC). Gli assi principali sono ad esempio child, parent, descendant, ancestor, sibling ecc.... alcuni possono essere abbreviati.

i metodi classici per accedere ad elementi del DOM sono 3:

- getElementById: restituisce l'elemento con l'ID specificato
- getElementsByClassName: restituisce una raccolta di elementi con la classe specificata
- getElementsByTagName: restituisce una raccolta di elementi con il tag specificato

**Jquery**: framework javascript che semplifica la manipolazione del DOM, la gestione degli eventi e le operazioni AJAX. jQuery ha una peculiarità sintattica, l'uso della keyword `$` per ogni comando della libreria

jQuery definisce allora una funzione `$()` che è un alias del costruttore jQuery(), e ha come parametro un selettore CSS.

slide 12 19-javascript.

il successo di jquery ha portato WHATWG a creare getElementsByClassName, querySelector e querySelectorAll.

AJAX (Asynchronous JavaScript and XML) è una tecnica di sviluppo web che consente di aggiornare parti di una pagina web in modo asincrono, senza dover ricaricare l'intera pagina.

non e un linguaggio ma un insieme di tecnologie:

- HTML e CSS per la presentazione
- DOM per la manipolazione dinamica del contenuto
- XMLHttpRequest per la comunicazione asincrona con il server
- JavaScript per orchestrare tutto

migliora l'esperienza utente rendendo le applicazioni web più reattive e interattive. ma può aumentare la complessità del codice e richiede una gestione attenta delle richieste asincrone.

XMLhttpREquest(XHR) è la grande differenza tra AJAX e le tecniche precedenti di comunicazione client-server.

sono state poi create alternative come funzioni jquery, librerie axios e fetch

javascript nella programmazione ha una filosofia di asincronia basata su eventi e callback.

perche l'asincronia funzioni correttteamente ci devono essere 4 condizioni:

- corretta esecuzione del codice
- tenere il browser libero

- flusso unico dell'algoritmo
- usare variabili locali

Una promessa è un oggetto che, si promette, tra un po' conterrà un valore. La promessa è creata dalla funzione chiamante e mantenuta dalla funzione chiamata. Qui nella versione semplice con un solo database,

le promesse risolvono piuttosto bene i 4 problemi.

funzionano anche le funzioni generator/yield per gestire l'asincronia in modo piu lineare. sono funzioni che possono essere messe in pausa e riprese, permettendo di scrivere codice asincrono in modo più simile a quello sincrono.

le funzioni async/await sono un modo più moderno e leggibile per gestire l'asincronia in JavaScript. Una funzione dichiarata con la parola chiave async restituisce una promessa, e all'interno di essa si possono usare le parole chiave await per attendere il completamento di altre promesse.

se hai molte chiamate indipendenti promise.all permette di eseguirle in parallelo e attendere che tutte siano completate.

Introdotto nel 2023. Non ancora molto diffusa, ma... Mescola generatori, iteratori e await/async. E' un loop in cui ogni iterazione è controllata da una funzione asincrona: se restituisce un valore si va avanti, mentre se dà null si esce dal loop:

## 6.4 modularizzazione in javascript

modularizzazione= suddividere il codice in moduli indipendenti e riutilizzabili

javascript non l'ha avuto per molto tempo, ma ora esistono due sistemi principali:

- CommonJS: usato principalmente in Node.js, utilizza require() per importare moduli e module.exports per esportarli.
- ES6 Modules: standard ufficiale di JavaScript, utilizza import e export per gestire i moduli.

i moduli aiutano a mantenere il codice organizzato, facilitano la manutenzione e promuovono il riutilizzo del codice.

## 6.5 template

le pagine web dianmiche spesso usano template per generare contenuti HTML in modo efficiente.

cioe creaano prima pagine con placeholder che vengono poi riempiti con dati dinamici al momento del rendering.

Ricorrere ad un linguaggio di programmazione per generare codice HTML però crea alcuni stupidi problemi sintattici solitamente ignorati dai linguaggi di programmazione tradizionali

l'interpolazione di stringhe è una tecnica che consente di inserire variabili o espressioni all'interno di stringhe di testo in modo semplice e leggibile. si usa sopratutto con i template literals in JavaScript, che utilizzano backtick (‘) e la sintassi `$espressione` per l'interpolazione.

si possono usare anche framework come handlebrars o mustache per creare template piu complessi.

o anche angular, react o vue che usano template per creare interfacce utente dinamiche.

le nuove applicaioni stanno sempre piu strette alla sintassi tradizionale.

Un componente è un insieme anche complesso di codice dotato di senso compiuto, autonomo, autosufficiente e incapsulato. Il nuovo modello di web vede le applicazioni come costruzioni di molteplici componenti indipendenti che scambiano tra loro informazioni.

ci sono molti framework, anche HTML ha creato i web components per creare componenti riutilizzabili con funzionalità incapsulate.

cioe puoi definire nuovi elementi personalizzati, creare un DOM shadow per incapsulare lo stile e il comportamento del componente, e riutilizzare questi componenti in diverse parti dell'applicazione o in progetti diversi.

## 6.6 routing

Con routing intendiamo l'assegnazione di un URI diverso ad ogni stato dell'applicazione web (cioè ad ogni selezione di contenuto differente).

Con l'approccio LAMP il problema del routing è molto semplice: ad ogni pagina HTML e/o ogni script server-side è automaticamente associato un URI (file system routing) e non ci sono complicazioni di sorta.

window.location contiene l'URI corrente della pagina web caricata nel browser.

window.history fornisce metodi per navigare nella cronologia del browser, come back(), forward() e go().

## 6.7 binding

il binding e il collegamento tra dati e interfaccia utente in un'applicazione web.

puo essere unidirezionale o bidirezionale.

# 7 semantic web

semantic web:scrtuttura comune che consente di condividere e riutilizzare dati tra applicazioni, aziende e comunita ciattion link: permette di creare collegamenti semantic web stack: illustra l'architettura del web semantico divverese classi sono oragnizzate 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 garfo RDF e un insieme di triple RDFle risorse sono rappresentate come nodi.

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

- la gesione delle informazioni puo essere parallelizzata
- informazioni parziali sono comunque valide

svantaggi:

- il modello di dati RDF e 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 possibilita di attribuire informazioni alle triple stesse

reificazione: prendo una tripla e gli do un identificativo per farla diventare parte di un alòtra 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 cllassi eproprieta

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