

ECMA SCRIPT

ECMAScript è lo standard ufficiale su cui si basa JavaScript. È come il "manuale delle regole" che definisce come deve funzionare il linguaggio JavaScript.

Caratteristiche principali:

- Definisce sintassi e funzionalità base di JavaScript
- Viene aggiornato periodicamente (ES6, ES2015, ES2016...)
- Introdotto per standardizzare JavaScript tra diversi browser

EVENT LOOP

L'event loop in JavaScript è un meccanismo fondamentale che permette di gestire l'esecuzione asincrona del codice in modo efficiente. Ecco una spiegazione semplice:

Quando scrivi codice, alcune operazioni possono richiedere tempo (come chiamate API, timer). Invece di bloccare l'esecuzione del programma, l'event loop si occupa di gestire queste operazioni in modo intelligente.

Funzionamento base:

1. Il codice sincrono viene eseguito normalmente
2. Quando c'è un'operazione asincrona (tipo setTimeout, fetch, ecc.), viene "messa in attesa"
3. JavaScript continua a eseguire altro codice senza fermarsi
4. Quando l'operazione asincrona è pronta, la sua callback viene messa in una coda
5. L'event loop preleva le callback dalla task queue e le esegue quando lo call stack è vuoto

DOM

Il DOM (Document Object Model) è essenzialmente la rappresentazione ad albero di una pagina web, trasformando ogni elemento HTML in un oggetto che JavaScript può manipolare.

- Ogni tag HTML diventa un "nodo"
- Questi nodi hanno relazioni gerarchiche (padre, figlio, fratello)

Tramite JavaScript puoi:

1. Modificare contenuti
2. Cambiare stili
3. Aggiungere/rimuovere elementi
4. Gestire eventi

JSON e AJAX

JSON (JavaScript Object Notation) è un formato leggero per scambiare dati, simile a un oggetto JavaScript, ma sotto forma di testo.

Struttura base:

- Formato chiave-valore
- Supporta stringhe, numeri, booleani, array, oggetti, null
- Facile da leggere per umani e macchine

AJAX (Asynchronous JavaScript And XML) è una tecnica per inviare e ricevere dati dal server senza ricaricare l'intera pagina web.

Funziona così:

- Permette di fare richieste al server in background
 - Aggiorna solo parti specifiche di una pagina
 - Rende le web app più veloci e interattive
1. Si verifica un evento in una pagina web, ad esempio la pagina viene caricata o si fa clic su un pulsante
 2. Un oggetto XMLHttpRequest viene creato da JavaScript
 3. L'oggetto XMLHttpRequest invia una richiesta a un server web
 4. Il server elabora la richiesta
 5. Il server invia una risposta alla pagina web
 6. La risposta viene letta da JavaScript
 7. L'azione appropriata, come un aggiornamento della pagina, viene eseguita da JavaScript

OOP - prototype

Il prototype in JavaScript è un meccanismo interno di ereditarietà che permette agli oggetti di condividere proprietà e metodi attraverso una catena di oggetti collegati. È un sistema che consente la condivisione efficiente di funzionalità tra istanze, dove ogni oggetto ha un

riferimento al proprio prototipo da cui può ereditare caratteristiche. Questo meccanismo funziona come un sistema di "eredità" in cui gli oggetti possono accedere e utilizzare metodi e proprietà definiti nei loro prototipi, creando una catena di ereditarietà che si risale fino all'oggetto prototipo base.

BUILD TOOL

Un build tool come Webpack è uno strumento che automatizza il processo di gestione e ottimizzazione delle risorse di un'applicazione web, come JavaScript, CSS e immagini. Webpack raccoglie i file, li trasforma, li combina e li minifica per migliorare le performance e facilitarne l'uso in produzione.

PROMISE

Una Promise in JavaScript è un oggetto che rappresenta un'operazione asincrona che può essere in tre stati:

- Pending (in attesa): stato iniziale
- Fulfilled (completata): operazione riuscita
- Rejected (fallita): operazione non riuscita

Una Promise è un oggetto che rappresenta il completamento (o fallimento) di un'operazione asincrona e il suo valore risultante. È un modo per gestire operazioni che non avvengono immediatamente, permettendo di specificare cosa fare quando l'operazione ha successo o fallisce.

Una Promise è come una "promessa" che in futuro ti darà un risultato. È come se dicessi: "Ti prometto che ti farò sapere l'esito di qualcosa, ma non immediatamente"

ERROR HANDLING

1. Riprova: prova a eseguire di nuovo un'operazione che non è riuscita
2. Fallback: usa un valore che è un fallback predefinito
3. Valore di errore: restituisce un valore di errore ma continua l'esecuzione
4. Propaga: passa l'errore verso l'alto
5. Log: scrivi un avviso o un errore
6. Termina: termina l'applicazione

7. Ignora: ignora l'errore e continua l'esecuzione

BROWSER API

Le Browser API sono funzionalità integrate nei browser web che permettono di interagire con vari aspetti del web e del dispositivo.

Esempi principali:

- Geolocalizzazione
- Storage (localStorage)
- Notifiche
- Riproduzione audio/video
- Manipolazione del DOM

Sono come "strumenti predefiniti" che i browser mettono a disposizione degli sviluppatori per creare applicazioni web più ricche e interattive, senza dover sviluppare tutto da zero

Typescript

TypeScript è un superset di JavaScript che aggiunge il supporto per i tipi statici. Permette di scrivere codice più sicuro e leggibile, riducendo gli errori grazie alla verifica dei tipi durante la compilazione. TypeScript si compila poi in JavaScript, quindi può essere eseguito ovunque venga supportato JavaScript.