Vue Advance

Computed Properties

- Assumono valori dinamici in base ad altre property del nostro oggetto Vue;
- Possono essere utilizzate come normali proprietà del campo data;

Usare le computed properties conviene perché usa la cache, e i nuovi valori vengono calcolati solo se e quando le dipendenze dovessero cambiare.

Utilizzare le computed piuttosto che i metodi, evita di generare i valori di ritorno ad ogni accesso, così da rigenerarli solo se e quando le dipendenze dovessero cambiare (caching).

Vengono utilizzate per calcolare il valore di una proprietà in base ad alcune altre condizioni.

Watchers

Si può usare questa opzione per attivare una funzione ogni volta che una proprietà cambia.

Computed Properties	Watchers
Restituiscono un valore, sono sincrone e non hanno effetti collaterali	Hanno effetti collaterali
Creano nuove proprietà reattive	Chiamano solo funzioni
Possono reagire ai cambiamenti di più oggetti	Possono guardare solo un oggetto alla volta
Sono memorizzati nella cache, quindi si ricalcolano solo quando le cose cambiano	Vengono eseguiti ogni volta che un oggetto cambia
Vengono valutate solo quando è necessario per essere utilizzate	

Componenti

I componenti ci consentono di suddividere l'interfaccia utente in parti indipendenti e riutilizzabili e di pensare a ogni parte isolatamente, per farlo possiamo:

- utilizzando un custom tag: <hello-world />
- utilizzando il tag component e l'attributo is: <component :is="HelloWorld" />

Possiamo registrarli:

- Globalmente: ma ciò impedisce di rimuovere i componenti inutilizzati e rende le relazioni di dipendenza meno esplicite.

- Localmente: rende disponibile il componente solo all'interno del componente in cui è registrato

Props

Rappresentano attributi custom che permettono di configurare un componente dall'esterno, definite a priori, è possibile definire il loro tipo o l'eventualità.

Passate ad un componente in maniera statica o dinamica.

Vue implementa il cosiddetto one-way data flow, le prop passate ad un componente possono essere modificate dall'esterno, ma non dal componente stesso, in quanto il flusso di dati è monodirezionale, dall'alto al basso.

Risolvibile grazie agli eventi custom, è possibile per un componente comunicare informazioni verso l'esterno, con l'opzione emits un componente può definire a priori gli eventi che è in grado di emettere e generare eventi.

Il cosiddetto Prop Drilling è quando dobbiamo passare una prop da un componente A ad uno C e per farlo lo passiamo a B che è in mezzo.

Possiamo evitarlo con il Provide - Inject, dove A usa l'opzione provide per fornire dati a tutti i suoi discendenti e C usa inject per ottenere dati forniti tramite provide da qualsiasi componente ascendente e utilizzarli come normali properties.

Nel progetto abbiamo preferito usare gli eventbus perché i componenti non hanno una gerarchia diretta.

Slots

Come le props ma per passare codice HTML fra componenti.

Composition API

Usate al posto delle Options API (dividendo il codice nei blocchi data, computed, methods, ecc.). I blocchi data, computed e methods sono stati rimpiazzati dalla funzione setup(), che restituisce un oggetto contenente tutte le variabili a cui il template deve avere accesso, esiste ref che è grossomodo un equivalente di data.

Rispetto alle options API c'è maggior libertà nell'organizzazione del codice, miglior supporto a TypeScript e codice più compatto.

Aggiungendo l'attributo setup al tag <script> il contenuto del tag diventi quello della funzione setup().

Node.js

Una piattaforma software cross-platform che permette di creare il proprio Web server.

Web application con connessioni real-time, two-way, dove non solo il client, ma anche il server può iniziare una connessione, permettendo di trasferire dati liberamente.

Usa un modello I/O non bloccante e ad eventi, che lo rende un framework leggero ed efficiente, le richieste vengono tradotte in eventi (ognuno dei quali è associato a un event handler) e inserite in una coda. C'è un event loop single-threaded che controlla se ci sono nuovi eventi nella coda.

Node.js è capace di gestire un elevato numero di connessioni simultanee. Non congeniale per applicazioni CPU-intensive.

Node.js incorpora Node Package Manager (NPM), il più grande ecosistema di librerie open source al mondo.

PHP

PHP è un linguaggio di scripting, interpretato, originariamente concepito per la programmazione di pagine Web dinamiche lato server.

L'interprete PHP pre processa tutti i file con estensione PHP, che sono sostanzialmente file contenenti HTML, all'interno dei quali è presente del codice PHP, contenuto all'interno dei delimitatori <?php ... ?>. Ma è possibile anche creare file con estensione .php per poi includerli nel file principale.

La sintassi PHP è di tipo C-like:

C	PHP
Tipizzazioni molto forte	**Tipizzazione debole**
Supporta i puntatori	**Non supporta i puntatori**
Tipi di dati numerici supportati: int8_t, unint8_t, int16_t, uint16_t, int32_t, uint32_t, int64_t, uint64_t, int, unsigned int, signed char, unsigned char, short, unsigned short, long, unsigned long, long long, unsigned long long	Tipi di dati numerici supportati: 32-bit signed int, 64-bit signed int (long integer)
Tipi di dati float supportati: single precision float, doublelong double	Tipi di dati float supportati: double, precision, float
array di dimensioni fisse	**Array di dimensione variabile**
Non supporta array associativi	Supporta array associativi

Cookies

Sono un meccanismo di supporto alla gestione delle sessioni basato sull'idea che sia il client a mantenere lo stato di precedenti connessioni, e ad inviarlo al server

di pertinenza ogni volta che richiede un documento.

Il termine cookie indica un blocco di dati opaco lasciato dal server in consegna ad un richiedente per poter ristabilire in seguito il suo diritto alla risorsa richiesta .

Alla prima richiesta di uno user-agent, il server fornisce la risposta ed un header aggiuntivo, il cookie, con dati arbitrari e con la specifica di usarlo per ogni successiva richiesta. Il server associa a questi dati le informazioni sulla transazione. Ogni volta che lo user-agent accederà a questo sito, rifornirà i dati opachi del cookie che permettono al server di ri-identificare il richiedente, e creare così un profilo specifico.

Tipi

- Tecnici: usati dal gestore del sito per mettere in opera alcune funzioni o rendere più agile la navigazione;
- Analitici: usati dal gestore del sito per raccogliere alcune informazioni in forma aggregata sugli utenti;
- Di profilazione: sono usati dal gestore del sito per raccogliere dati personali sui visitatori [solo questi è necessario accettare].

JavaScript Advanced

API HTML5

WebStorage

Le API WebStorage consentono di salvare e recuperare dati localmente al browser, dati memorizzati in coppie chiave-valore.

Due oggetti principali:

- localStorage: i dati non hanno scadenza e non vengono cancellati alla chiusura del browser.
- sessionStorage: a differenza di localStorage, i dati sono mantenuti solo per la sessione corrente e cancellati quando il browser viene chiuso.

A differenza dei cookies:

- Limite di memoria più ampio 10MB contro i 4MB dei Cookie.
- Non ha una data di scadenza.
- Non è possibile gestirli lato server.

WebWorker

Permettono di eseguire codice Javascript in background.

Drag and Drop

Le API Drag and Drop consentono di effettuare operazioni di drag and drop all'interno della pagina.

Geolocation

L'API Geolocation consente di ottenere la posizione geografica dell'utente.

IndexedDB

Le API IndexedDB consentono di creare e manipolare un database all'interno del browser, consentono di memorizzare quantità di dati strutturati più grande rispetto alle API WebStorage.

API JavaScript

WebSocket

Le WebSocket consentono di instaurare un canale di comunicazione full-duplex attraverso una singola connessione TCP.

WebForms

Sono API per la validazione dei campi di input.

WebHistory

Web History fornisce un accesso facilitato all'oggetto windows.history, che contiene gli URL visitati dall'utente.

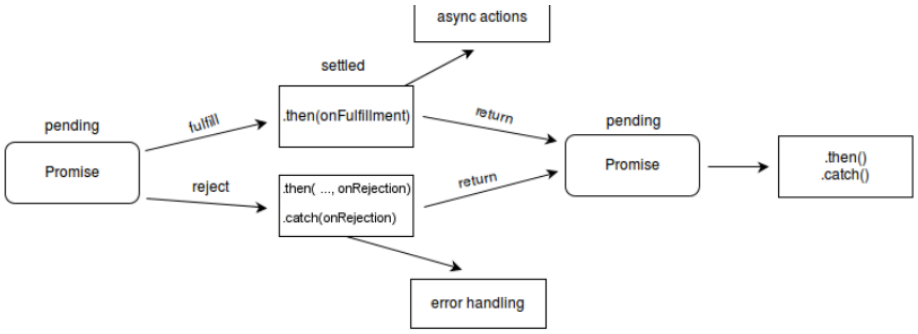
ECMAScript

Specifica tecnica di un linguaggio di scripting, standardizzata e mantenuta da ECMA International nell'ECMA262 ed ISO/IEC 16262, la sua implementazione più conosciuta è Javascript.

Le versioni vanno da ES2 a ES6.

L'ultima versione aggiunge Promise che consente di gestire l'eventuale completamento, o fallimento, di un'operazione asincrona.

In una promessa, il valore non è necessariamente noto al momento della creazione e la promessa consente di associare handler sia in caso di successo che di errore di un'operazione asincrona; invece di restituire il valore in modo sincrono, l'idea è quella di restituire la promessa di fornire il valore in un momento futuro.



Un'altra aggiunta sono le async/await, uno zucchero sintattico che semplifica l'utilizzo delle funzioni asincrone e, di conseguenza, anche delle Promise.

Mobile

Alcune definizioni:

Consiste in tecniche/tecnologie che permettono a utenti in movimento di utilizzare device portatili, di eseguire applicazioni e di connettersi ad applicazioni remote.

È una tecnologia che consente la trasmissione di dati, voce e video tramite un computer o qualsiasi altro dispositivo abilitato wireless senza dover essere collegato ad un collegamento fisico fisso.

È un termine generico che si riferisce a una varietà di dispositivi che permettono alle persone di accedere a dati e informazioni ovunque si trovino.

Nel mobile bisogna rispettare alcune caratteristiche, come:

Portabilità: ridurre la dimensione dei dispositivi in modo da creare computer che potessero essere portati in giro con relativa facilità.

Miniaturizzazione: creare componenti hardware nuovi e molto piccoli che permettano l'utilizzo del dispositivo personale durante uno spostamento.

Connettività: sviluppare dispositivi e applicazioni che permettessero all'utente di essere online e di comunicare wireless durante gli spostamenti.

Bluetooth Low Energy (BLE) è una tecnologia di rete personale senza fili che mira a nuove applicazioni nel settore sanitario, fitness, beacon (trasmettitori hardware che trasmettono il loro identificatore ai dispositivi elettronici portatili vicini.), sicurezza e intrattenimento domestico.

Convergenza : integrare tutti i dispositivi esistenti in un unico dispositivo ibrido in grado di fare tutto.

Divergenza: ogni dispositivo è pensato per svolgere una specifica funzione.

Altri attori possono essere:

Mobile communication: infrastruttura creata per garantire una comunicazione continua e affidabile per lo scambio di dati e voce utilizzando reti wireless.

Hardware mobile: dispositivo di elaborazione portatile con la capacità di recuperare ed elaborare i dati.

Software mobile: è il programma software che è stato sviluppato specificamente per essere eseguito su hardware mobile.

Alcuni problemi da affrontare nel mobile sono:

- I dispositivi mobile sono «resource-constrained».
- La connettività mobile è altamente variabile in termini di prestazioni e affidabilità.
- I dispositivi mobili sono intrinsecamente meno sicuri, inoltre nuove problematiche dovute alla privacy.

Sviluppo mobile

Conviene scegliere nativo o ibrido? Dipende da:

- Che tipo di app deve essere sviluppata;
- Esigenze degli utenti.

Native

Applicazioni sviluppate appositamente per un sistema operativo, utilizzando il suo linguaggio di sviluppo.

Vantaggi:

- High performance
- Sicurezza
- Personalizzazione (UX)
- Forte compatibilità con altre app
- Assoluta compatibilità con API fornite dal OS

Svantaggi:

- Costo
- Tempistiche

Vantaggi	Svantaggi
Le app native sono **più veloci** delle app Web	Le app native sono più **costose** da sviluppare rispetto alle app Web
Le app native possono **accedere** a **risorse** di sistema/dispositivo come un GPS o una fotocamera	La progettazione e la creazione dell'app nativa per diverse piattaforme come iOS e Android è **costosa** e richiede tempo
Queste app possono funzionare **senza** una **connessione** Internet	La **manutenzione** e **l'aggiornamento** costante delle app native comportano costi maggiori
Queste app hanno maggiore **sicurezza** rispetto alle app Web, poiché le app native devono essere approvate dall'App Store	Può essere difficile avere un'app mobile nativa approvata dall'**App Store**
Le app native sono più «**facili**» da sviluppare grazie alla disponibilità di strumenti per sviluppatori, elementi di interfaccia e SDK	

(Responsive) Web app

È possibile accedere alle web app tramite il browser del dispositivo mobile, non è necessario scaricare e installare l'app su un dispositivo mobile per usarle, sviluppata come pagine Web e può essere utilizzata sulla maggior parte dei dispositivi in grado di navigare sul web.

Vantaggi	Svantaggi
Le app Web funzionano nel **browser**, quindi non è necessario installarle o scaricarle	Le app Web **non** funzionano **senza** una **connessione** Internet.
Le app Web sono **facili da gestire**, poiché hanno una base di codice comune indipendentemente dal sistema operativo	Le app Web funzionano più **lentamente** delle app mobili
Le app Web sono più **facili** e **veloci** da **creare** rispetto alle app mobili native	È **difficile** scoprire le app Web poiché non sono ospitate in un database specifico come un app store
Non è richiesta **l'approvazione** dell'App Store, hai solo bisogno di un browser	Le app Web presentano **rischi maggiori** e scarsa qualità e non vi è alcuna garanzia di sicurezza poiché le app Web non devono essere approvate dagli app store

Queste applicazioni Web sono utilizzabili in qualsiasi dimensione del riquadro di visualizzazione del browser (usabilità).

Progressive Web App

Un'applicazione Web progressiva (PWA) è distribuita attraverso il Web e creata utilizzando tecnologie web.

Sono app web che simulano il comportamento delle app mobili native

Utilizzano funzionalità della piattaforma Web in combinazione con il miglioramento progressivo per offrire agli utenti un'esperienza pari a quella delle app native.

Per progressively enhanced si intende una strategia nel web design che pone l'accento sul contenuto, consentendo a tutti di accedere al contenuto e alle funzionalità di base, mentre gli utenti con funzionalità del browser aggiuntive o un accesso a Internet più rapido ricevono la versione avanzata.

Questa strategia prevede la separazione del contenuto dalla presentazione.

Una PWA dovrebbe essere:

- ***Rilevabile:*** i contenuti possono essere trovati attraverso i motori di ricerca
- **Installabile**: può essere disponibile sulla schermata iniziale del dispositivo o sul programma di avvio delle app
- ***Collegabile***:puoi condividerlo inviando un URL
- **Indipendente** dalla rete: funziona **offline** o con una connessione di rete scadente.
- **Progressively enhanced**: è ancora utilizzabile a livello base sui browser più vecchi, ma perfettamente funzionante su quelli più recenti
- **Re-engageable**: è in grado di **inviare notifiche** per nuovi contenuti.
- **Responsive**: è utilizzabile su qualsiasi dispositivo dotato di schermo e browser: cellulari, tablet, laptop, TV, frigoriferi, ecc.
- **Sicuro**: le connessioni tra l'utente, l'app e il tuo server sono protette da eventuali terze parti che tentano di accedere a dati sensibili

Ibrido

Le app mobili ibride sono applicazioni installate su un dispositivo, proprio come qualsiasi altra app, ciò che le differenzia è il fatto che combinano elementi da app native con elementi da app Web.

Distribuite in un contenitore nativo che usa un oggetto WebView mobile. Quando si utilizza l'app, questo oggetto visualizza i contenuti web grazie all'utilizzo di tecnologie web.

Come sviluppare app

Ecco alcune tecnologie:

- Flutter: creato da Google, utilizzato per sviluppare applicazioni multipiattaforma, usa Dart come linguaggio;
- React Native: create da Facebook, utilizzato per sviluppare applicazioni Android e iOS, usa JavaScript e tecnologie web;
- Ionic Framework: è un toolkit dell'interfaccia utente open source per la creazione di app mobili e desktop performanti e di alta qualità utilizzando tecnologie Web, consente di creare app per tutti i principali app store da un'unica base di codice, nativo e ottimizzato per il web;
- Cordova: framework di sviluppo mobile open source di Apache che utilizza le tecnologie web;
- Capacitor: progetto open source che esegue app Web moderne in modo nativo su iOS, Android e Web.

Data Visualization

Quando si usano i dati per mostrare o spiegare qualcosa in maniera visiva, esempio:

Si usa perchè: l'idea della visualizzazione ci aiuta a vedere ciò che non abbiamo notato prima. Particolarmente vero quando si cerca di identificare le relazioni e di trovare un significato in enormi quantità di dati raccolti (cioè i Big Data).

Visualization: qualsiasi tipo di rappresentazione visiva di informazioni progettata per consentire la comunicazione, l'analisi, la scoperta, l'esplorazione, la trasmissione di un messaggio, ecc.

Information Visualization: studio delle rappresentazioni visive di dati astratti (

sia dati numerici che non numerici) per rafforzare la cognizione umana.

Infografica: rappresentazione visiva di informazioni in più sezioni con lo scopo di comunicare uno o più messaggi specifici. Con lo scopo di rendere i dati facilmente comprensibili a colpo d'occhio o con una breve attenzione.

Data visualization: rappresentazione grafica di informazioni e dati.

Qualità delle buone visualizzazioni:

1. Truthful: essere onesti facendo le giuste scelte di design.
2. Functional: scegliere le forme grafiche in base ai compiti che si desidera permettere di fare.
3. Beautiful: gli oggetti devono essere o vissuti come belli dal maggior numero possibile di persone.
4. Insightful: le buone visualizzazioni aprono la strada a scoperte preziose che sarebbero inaccessibili se le informazioni fossero presentate in un modo diverso.
5. Enlightening: qualità composta dalle quattro precedenti.

Processo di visualizzazione dei dati

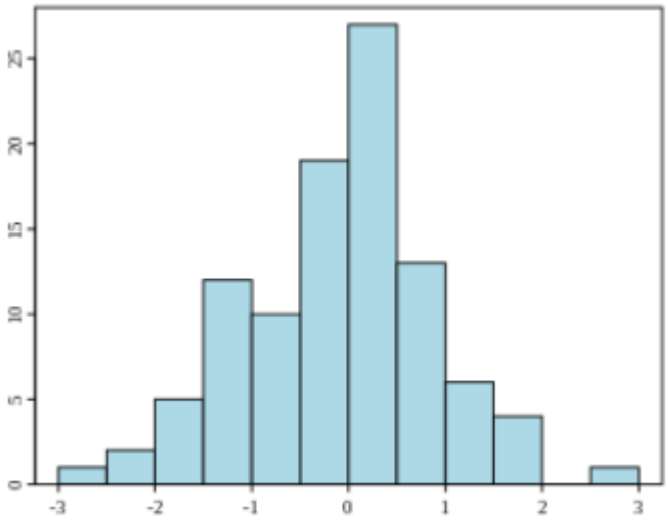
I dati possono essere:

- Categorici: descrivono categorie o gruppi
- Numerici: rappresentano numeri (quantità, range ecc..)

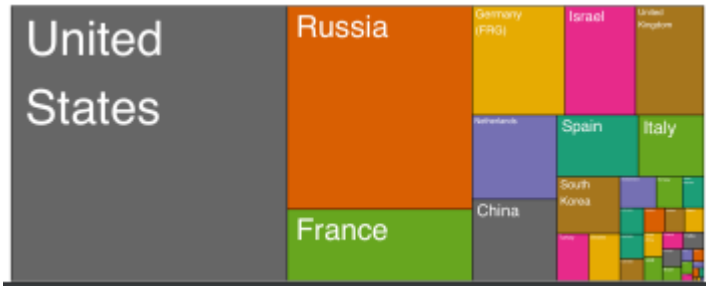
Per la visualizzazione esistono diversi grafici:

- Bar chart: per comparare più valori, se modifichiamo troppo la scala manipoliamo le informazioni.

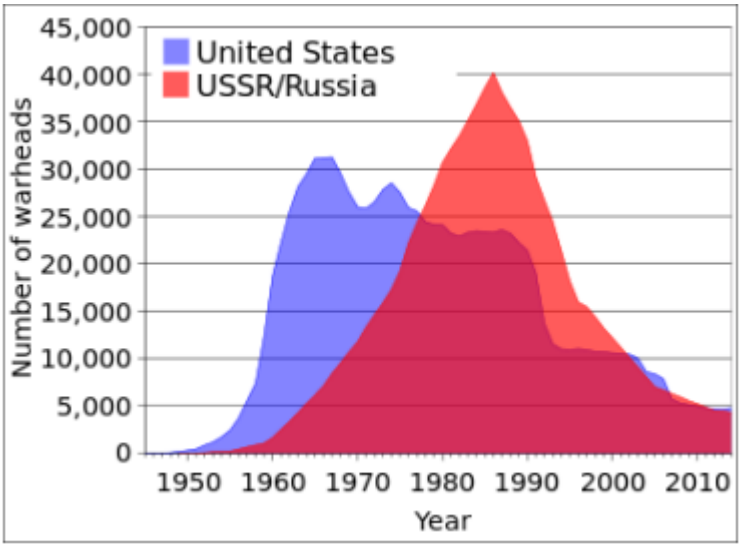
- Istogramma: rappresentazione grafica di una distribuzione di una variabile numerica. Asse X: intervalli invece Asse Y: frequenza.



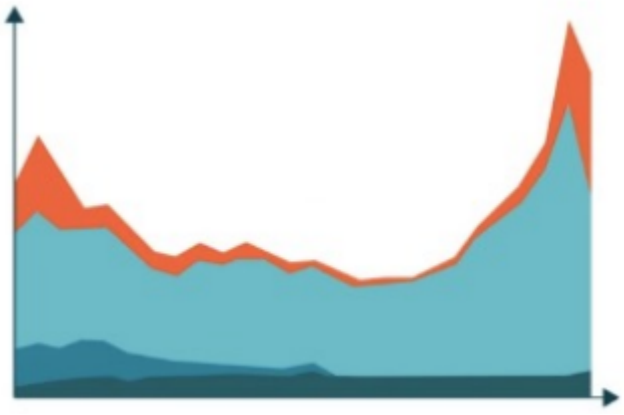
- Grafico a torta: usato per rappresentazioni di variabili quantitative misurate su classi di categorie, circolare per evitare un ordine, ha il difetto che la dimensione delle slice non è un facilmente leggibile. NON è la scelta migliore.
- Treemap: visualizza i dati gerarchici come un insieme di rettangoli annidati.



- Area chart: descrivono i cambiamenti delle variabili nel tempo

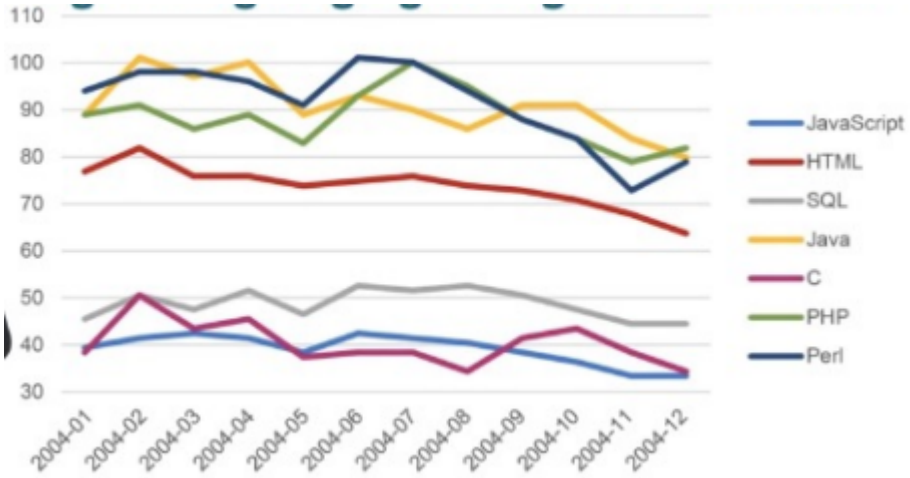


- Stacked area chart:



- Line chart: utilizzati per rappresentare i dati delle serie temporali. È

adatto a mostrare dati che cambiano in modo continuo.



- Scatter plot: due variabili di un dataset sono riportate su uno spazio cartesiano, dati visualizzati tramite una collezione di punti.
- Bubble chart: tipo di grafico in cui ogni entità rappresentata è definita in termini di tre parametri numerici distinti.
- Choropleth map: usano i colori per rappresentare quantità e qualità di una certa area geografica i cui confini sono prestabiliti.
- Dot maps: usano una stessa forma per rappresentare ogni singola unità di dati. Il risultato visivo è che le zone con maggiore densità risultano ricoperte da una quantità maggiore di punti.

D3.js

D3.js (Data-Driven Documents) è una libreria Javascript per produrre visualizzazioni di dati dinamiche e interattive nei browser, estremamente veloce e supporta comportamenti dinamici.

Gamification

"Gamification" is the use of game design elements in non-game contexts.