

JavaScript (6)



Marina Ribaudó, marina.ribaudó@unige.it

Cookies

2

*“Cookies are used for persistence of user data, all cookies valid for a page are **sent from the browser to the server** for every request **to the same domain** - this includes the original page request, any subsequent Ajax requests, all images, style-sheets, scripts and fonts.”*

Però i cookies hanno delle limitazioni...

Web Storage

3

- Permette di memorizzare dati sul browser dell'utente in modo
 - Permanente: **localStorage**
("simile" ai cookies persistenti)
 - Temporaneo: **sessionStorage**
("simile" ai cookies di sessione)



*"Web storage is intended for **storing information client-side** (and **never transmitted to the server**), while cookies are intended for communication with server and automatically added to headers of all requests, if applicable."*

https://en.wikipedia.org/wiki/Web_storage

Web Storage

4

- Web storage fornisce **maggiori capacità di memoria** rispetto ai 4 KB dei cookies
 - 5 MB per ogni origine in Firefox e Opera
 - 10 MB per ogni origine in Google Chrome
 - 25 MB per BlackBerry

Web Storage

5

- Web storage fornisce **maggiori capacità di memoria** rispetto ai 4 KB dei cookies

The screenshot displays a web application interface on the left and its developer tools on the right. The web application, titled "SVILUPPO DI APPLICAZIONI WEB - 65704", features a navigation bar with links for Course, Settings, Participants, Grades, Reports, and More. A welcome message for the "SAW, edizione 2023/24" course is shown, along with a list of course details: "Orario delle lezioni: LUN 14:00-16:00, MER 14:00-14:00, aula 505, Via Dodecaneso 35", "Date degli esami: da definire", and a link to the UniGe program. Below this, there are buttons for "Modalità d'esame" and "Annunci".

The developer tools on the right are open to the "Application" tab, showing the "Storage" section. The "Local storage" for the URL "https://2023.aulaweb.unige.it" is expanded, revealing a list of key-value pairs. The "Value" column shows HTML snippets, including a list item and a modal backdrop. The "Background services" section is also visible, listing various services like Back/forward cache, Background fetch, and Background sync.

Key	Value
-2035694709/core_te...	{{^unmappedIcon}} <i class="icon fa {{key}} fa...
-2035694709/core_str...	Restore to view
-2035694709/core_str...	Done
-2035694709/core_str...	Page {\$a}
-2035694709/core_ic...	{^core/docs^-:fa-info-circle",core/book^-:fa-bo...
-2035694709/core_te...	<div class="card dashboard-card" role="listite...
-2035694709/core_te...	{{#showSuggestions}} <div class="d-md-inlin...
-2035694709/core_str...	Course is starred
-2035694709/core_str...	Show {\$a} items per page
-2035694709/core_te...	<div class="modal-backdrop in hide" aria-hidd...
-2035694709/core_str...	Starred course
-2035694709/core_str...	Course progress:
-2035694709/core_te...	<a class="dropdown-item {{#active}}active({/...
-2035694709/core_str...	Collapse
-2035694709/core_str...	Expand
-2035694709/core_str...	Previous page
-2035694709/core_str...	Last page
-2035694709/core_str...	Save changes

Web Storage: localStorage

6

- Memorizza dati in forma di coppie “**key**” = “**value**”

localStorage.setItem()

Es. `localStorage.setItem("client1", "Rossi");`
oppure `localStorage.client1 = "Rossi"`

localStorage.getItem()

Es. `var c = localStorage.getItem("client1");`
oppure `var c = localStorage.client1;`

localStorage.removeItem()

Es. `localStorage.removeItem("client1");`

localStorage.clear()

Web Storage: sessionStorage

7

- L'oggetto **sessionStorage** è uguale all'oggetto **localStorage**, ma memorizza i dati solo per la durata della sessione
- I dati vengono quindi persi quando si chiude browser/tab

Geolocation API

8

- Oggetto: **navigator.geolocation**
- Permette di ottenere latitudine e longitudine del dispositivo dell'utente (GPS o Wi-fi)
- Per accedere a questa informazione viene richiesta l'autorizzazione
- Funziona solo in HTTPS

Geolocation API

9

- Possiamo verificare se l'oggetto esiste

```
if (navigator.geolocation) {  
    /* geolocation is available */  
}  
else {  
    /* geolocation IS NOT available */  
}
```

Geolocation API

10

- **Geolocation.getCurrentPosition(success, error, options)**

*“[...]initiates an **asynchronous request to detect the user's position**, and queries the positioning hardware to get up-to-date information. When the position is determined, the **defined callback** function is executed. ...”*

- Ha tre parametri, solo il primo obbligatorio
 - **success** definisce la callback che viene chiamata in caso di successo
 - **error** definisce la callback che viene chiamata in caso di errore
 - **options**

<https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/getCurrentPosition>

Geolocation API

11

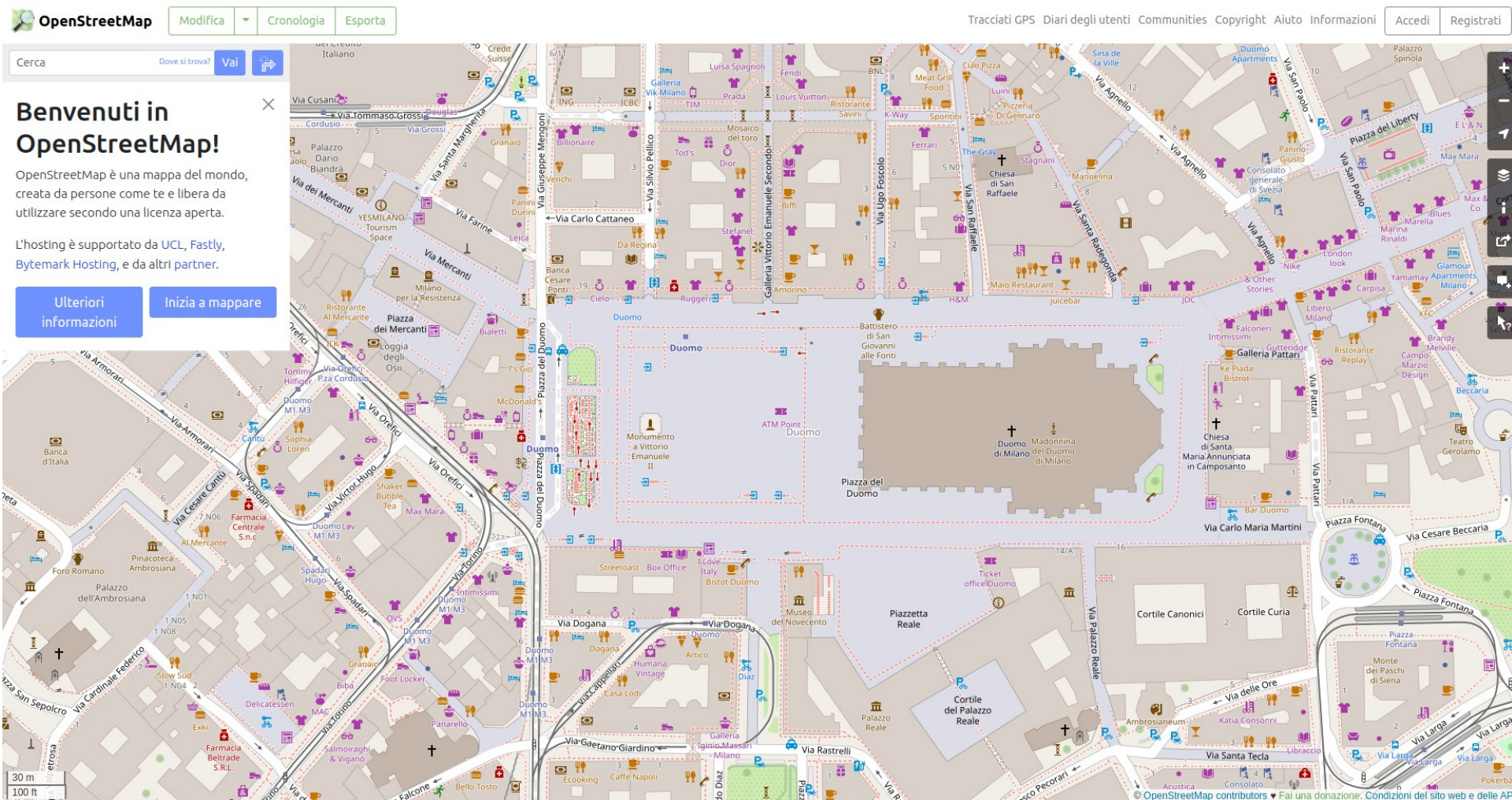
- **Geolocation.watchPosition(success, error, options)**

*“[...]the watchPosition() function, which has the **same input parameters** as getCurrentPosition(). **The callback function is called multiple times**, allowing the browser to either update your location as you move, or provide a more accurate location as different techniques are used to geolocate you. The error callback function, which is optional just as it is for getCurrentPosition(), can be called repeatedly.”*

Geolocation API

12

Non proprio precisa... come mai?



Web Storage e Geolocation

13

