





# Traduttore

"Duolingo questo lo sa fare?"

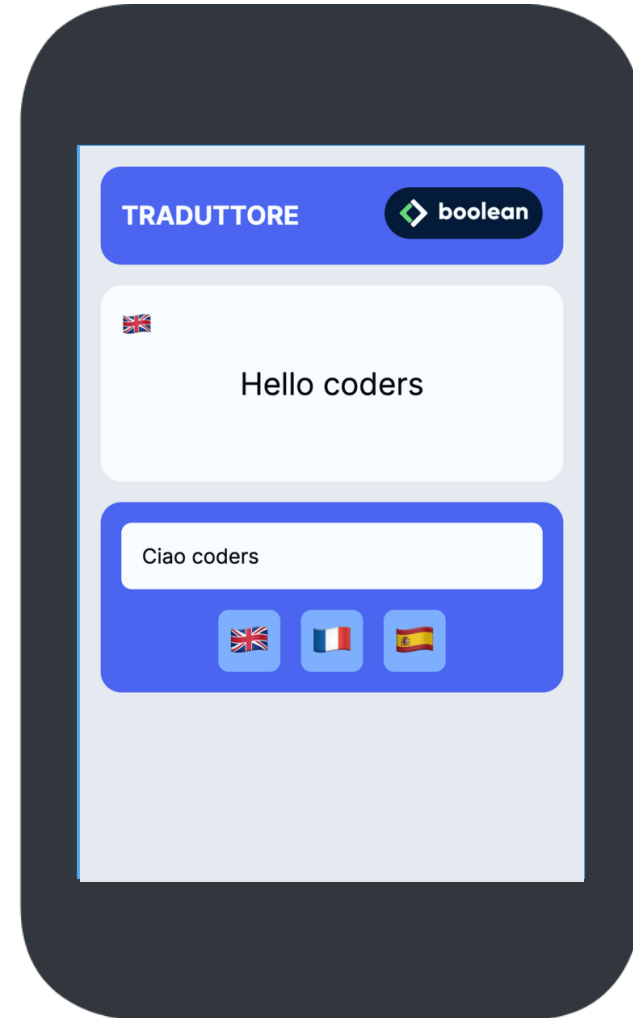


# Traduttore

Sviluppiamo una semplice app che sfrutta un servizio esterno per implementare la funzionalità di traduttore.

## Contenuti:

- CSS positions
- JS objects
- Web API
- Fetch method
- JSON





# CSS Positions



# Position

**Position** imposta la modalità con cui un elemento deve essere posizionato nella pagina, si applica a tutti gli elementi e non viene ereditata.

Può assumere uno dei seguenti valori:

- **static** - il valore di default
- **relative**
- **absolute**
- **fixed**





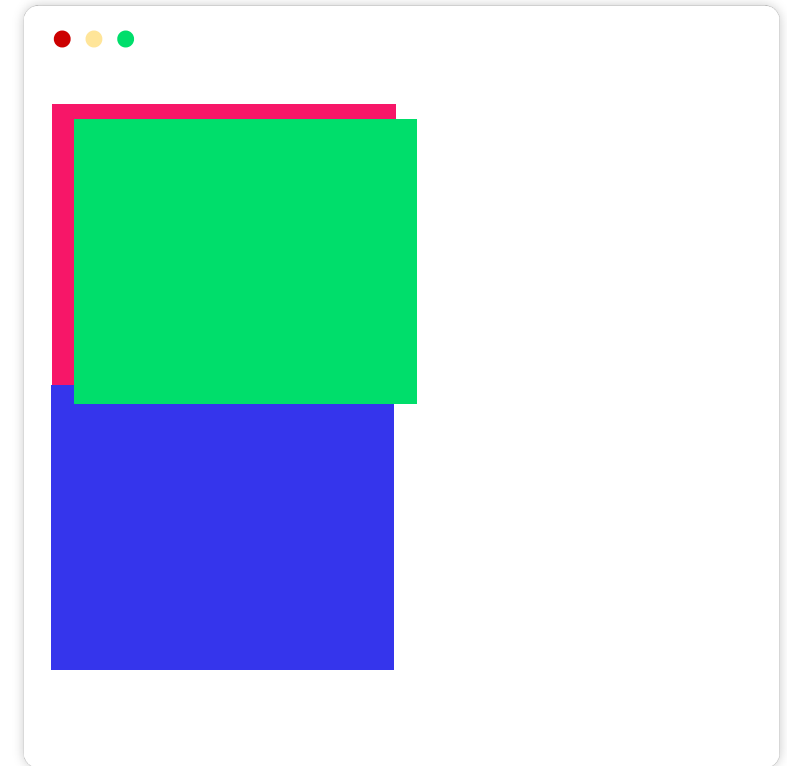
## Position: absolute

Il posizionamento assoluto avviene sempre rispetto al "contenitore" dell'elemento rappresentato dal primo elemento antenato (parent) che abbia un posizionamento diverso da *static* (default).

**L'elemento viene rimosso dal flusso** del documento in tutto, ed è posizionato in base ai valori forniti con le proprietà `top`, `right`, `bottom` o `left`.

```
1 <div class="white">
2   <div class="pink"></div>
3   <div class="green"></div>
4   <div class="blue"></div>
5 </div>
```

```
1 .white{
2   position: relative;
3 }
4
5 .green{
6   position: absolute;
7   top: 10px;
8   left: 10px;
9 }
```



Se non esiste un ancestor con position diverso da *static*, il posizionamento assoluto avviene rispetto all'elemento radice `<html>`.

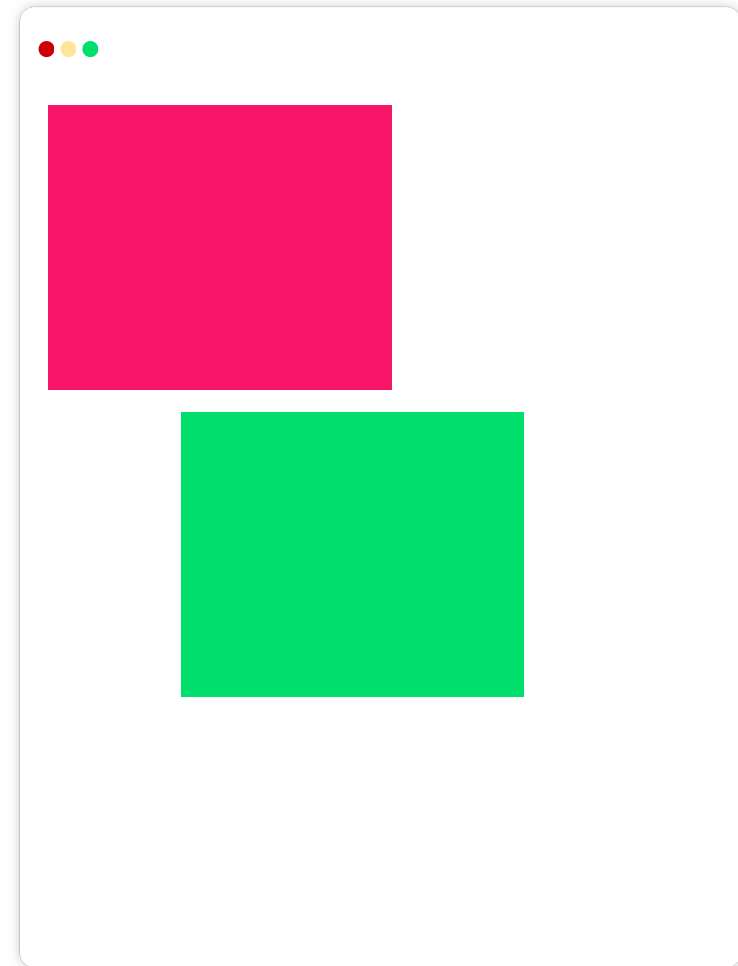


## Position: relative

L'elemento viene posizionato **relativamente alla posizione naturale** che l'elemento stesso avrebbe occupato nel normale flusso del documento. *Top, right, bottom e left* (in px, %, etc...) consentono lo spostamento dell'elemento *rispetto alla sua posizione naturale*.

**L'elemento non viene rimosso dal flusso del documento** e quindi gli altri elementi vicini, influenzati dal suo ingombro, continuano a considerare il suo ingombro originale.

```
1 .green {  
2   position: relative;  
3   top: 0;  
4   left: 25px;  
5 }
```





## Position: fixed

Un elemento con posizionamento fisso non scorre con il resto del documento: rimane sempre fisso al suo posto.

La differenza rispetto ad absolute sta nel fatto che:  
**per fixed il "contenitore" di riferimento è sempre il Viewport** (cioè la parte visibile della pagina HTML).





# JavaScript

Gli oggetti



# JS Objects

## Un "raggruppatore" di informazioni

Un **Object** *oggetto letterale*:

è una struttura di dati che può contenere più valori ai quali si può accedere per mezzo di un **indice di nomi**.

E' un'entità che raggruppa dati e comportamenti correlati: ad esempio, un oggetto "**persona**" potrebbe avere proprietà come "nome", "età" e "sesso", che contengono i rispettivi valori e metodi come "cammina", "mangia", "dormi" per le rispettive funzioni.





# Objects

## La sintassi

Come si scrive un oggetto letterale?

1. Creiamo una variabile
2. Apriamo le parentesi graffe
3. Inseriamo una serie di coppie  
nome: valore

```
1 const car = {  
2   color: "Red",  
3   firm: "FIAT",  
4   model: "Doblò"  
5 }
```



Una coppia (detta anche *chiave/valore*) viene detta **proprietà**.



# Objects

## La sintassi

Come si recupera un valore da un object?

### dot notation

```
1 const car = {  
2   color: "red",  
3   firm: "FIAT",  
4   model: "Doblò"  
5 }  
6 console.log(car.model); // 'Doblò'  
7  
8
```



# Strutture dati

## Time to Recap



### **Variabili con tipi di dato semplice:**

servono per immagazzinare un singolo valore, che può essere modificato nel tempo.



### **Array:**

servono per salvare una lista ordinata di dati. L'indice numerico viene utilizzato per accedere al dato e può servire per identificare l'ordine della lista.



### **Oggetti:**

servono per salvare, nello stesso "contenitore", più informazioni riguardanti una unica entità. La chiave della proprietà viene usata per accedere al valore corrispondente.

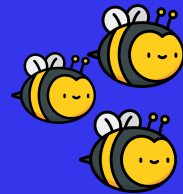
### **E combinazioni di questi tipi:**

- Array di oggetti
- Oggetti che contengono array tra le proprietà
- ...



# API

Application Programming Interface





# API

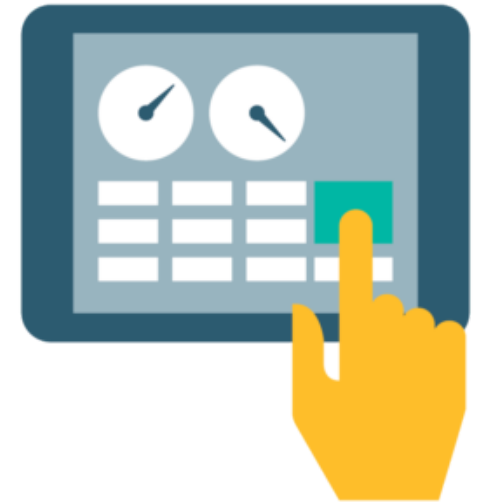
## Concetto di Interfaccia

Un' **interfaccia** è un insieme di regole che permettono di "comunicare" con qualcosa. Ne siamo circondati nella vita di tutti i giorni:

- Interruttore della luce
- Telecomando del televisore
- Elettrodomestici vari

In tutti questi casi ci sono delle regole che vanno rispettate, e delle modalità precise per ottenere il risultato desiderato!

Ad esempio non posso cambiare il canale della TV premendo il tasto off, né spegnerla premendo il tasto 5.





# API

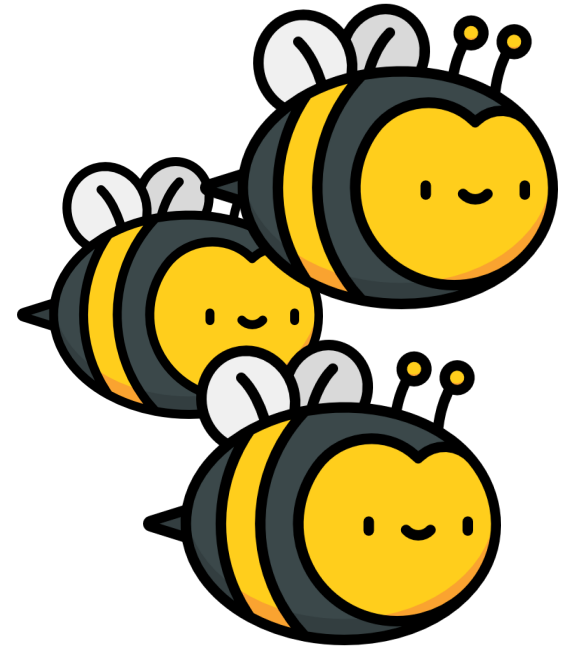
## Application Programming Interface

Quando parliamo di **API** parliamo di un'interfaccia per la comunicazione tra sistemi (programma/software/servizio ecc.)

**Con API si intende un insieme di funzionalità rese disponibili (ad un programmatore) per svolgere un insieme coerente di operazioni.**

**È un termine generico** che indica delle operazioni rese disponibili nascondendo parte della complessità sottostante.

Ciascuna API è diversa: se ne studia la documentazione e si apprendono le regole per ottenere il risultato desiderato, secondo le indicazioni fornite.







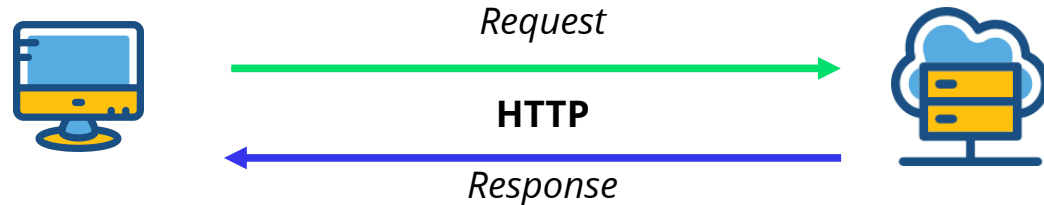
# MyMemory Web API

## Servizio traduzione di terze parti

Database di traduzioni gratuito basato su traduzioni generate dalla sua community.

Ha una interfaccia web, ma mette a disposizione anche una API con la sua documentazione.

<https://mymemory.translated.net/doc/>



[Home](#) | [About MyMemory](#) | [Contact Us](#)

[About MyMemory](#)  
[MyMemory in detail](#)  
[CAT integration](#)  
**[API specifications](#)**  
[Usage Limits](#)  
[Terms of Service](#)

### How-To

[Look up terms](#)  
[Get a TM from a TXT](#)  
[Get a TM from a TMX](#)  
[Contribute with a TM](#)

## MyMemory: API technical specifications

### Quick start with our REST API!

- [Search and contribute to the archive](#)
- [Import whole TMX memories to the archive](#)

Our REST API has been designed to be as easy as possible to implement and lets you **search and contribute to MyMemory archives**.

Please, before you start, have a look at the [Usage Limits](#) and to the [Terms of Service](#).

## Get

Searches MyMemory for matches against a segment.

Call example:

<https://api.mymemory.translated.net/get?q=Hello World!&langpair=en|it>



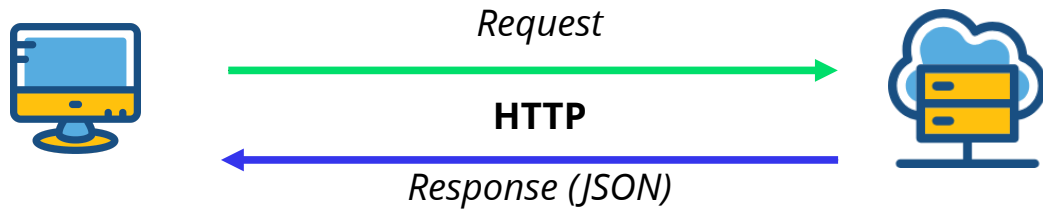
# JSON

## JavaScript Object Notation

Come comunicano client e server?

Esistono alcuni standard, ma il più comune resta il JSON.

**JSON (JavaScript Object Notation)** utilizza la notazione che conosciamo per definire gli oggetti in JavaScript



```
1 {  
2   "users": [  
3     {  
4       "name": "Ema",  
5       "cognome": "Princi",  
6       "indirizzo": "Torino",  
7       "anni": 20  
8     },  
9     {  
10      "name": "Max",  
11      "cognome": "Rossi",  
12      "indirizzo": "Roma",  
13      "anni": 54  
14    }  
15  ]  
16 }  
17
```



# JSON Viewer

## Chrome Extension

Estensione di Chrome che formatta le risposte JSON e ne semplifica la visualizzazione da browser.

<https://chromewebstore.google.com/detail/json-viewer/gbmdgpbipfallnflgajpaliibnhdgobh>



chrome web store

Scopri

Estensioni

Temi

Boolean Careers

## { } JSON Viewer

In primo piano 4,5 ★ (1.060 valutazioni)

Estensione

Strumenti per sviluppatori

1.000.000 utenti

```
1 // 20150625170715
2 // https://api.github.com/repos/tulios/json-viewer
3
4 {
5   "id": 12635853,
6   "name": "json-viewer",
7   "full_name": "tulios/json-viewer",
8   "owner": {
9     "login": "tulios",
10    "id": 33231,
11    "avatar_url": "https://avatars.githubusercontent.com/u/33231?v=3",
12    "gravatar_id": "",
13    "url": "https://api.github.com/users/tulios",
14    "html_url": "https://github.com/tulios",
15    "followers_url": "https://api.github.com/users/tulios/followers",
16    "following_url": "https://api.github.com/users/tulios/following{/other_user}",
17    "gists_url": "https://api.github.com/users/tulios/gists{/gist_id}",
18    "starred_url": "https://api.github.com/users/tulios/starred{/owner{/repo}",
19    "subscriptions_url": "https://api.github.com/users/tulios/subscriptions",
20    "organizations_url": "https://api.github.com/users/tulios/orgs",
21    "repos_url": "https://api.github.com/users/tulios/repos",
22    "events_url": "https://api.github.com/users/tulios/events{/privacy}",
23    "received_events_url": "https://api.github.com/users/tulios/received_events",
24    "type": "User",
25    "site_admin": false
26  },
27  "private": false,
28  "html_url": "https://github.com/tulios/json-viewer",
29  "description": "It is a Chrome extension for printing JSON and JSONP."
```



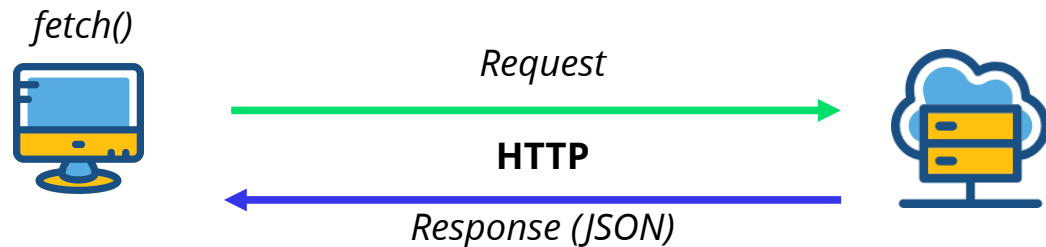
# Fetch API

## Eseguire chiamate asincrone a un server

La **Fetch API** in JavaScript è una funzionalità che consente di effettuare **richieste HTTP asincrone** verso server web e di gestire le risposte ottenute.

[https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)

```
1 async function translate() {  
2   const url = /* ... */;  
3   const response = await fetch(url);  
4   const jsonData = await response.json();  
5   console.log(jsonData);  
6 }
```





# Querystring

## Passare al server dati

Un modo per passare dati al server è attraverso l'URI "appendendo" al termine una **querystring**

querystring

---

www.boolean.careers?email=fabio@boolean.careers&title=CEO



# Querystring

Passare al server dati

www.boolean.careers?email=fabio@boolean.careers&title=CEO

- 1 ? indica che inizia la querystring
- 2 chiave=valore
- 3 ogni chiave-valore è separata da un &



# HTML Data Attribute

## Aggiungere dati custom nell'HTML

Forniscono la possibilità di memorizzare dati personalizzati direttamente negli elementi HTML.

Iniziano con ***data-*** e possono essere utilizzati per qualsiasi scopo, come memorizzare informazioni aggiuntive per scopi di scripting o stile.

L'attributo ***dataset*** di un elemento del DOM restituisce un oggetto che contiene tutti i data attributes dell'elemento stesso.

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Howto/Use\\_data\\_attributes](https://developer.mozilla.org/en-US/docs/Learn/HTML/Howto/Use_data_attributes)



```
1 <div id="product" data-id="123">Product #1</div>
```



```
1 const element = document.querySelector('#product');  
2 const productId = element.dataset.id;  
3 console.log(productId); // Stampa '123'
```



# Challenge





# Boolingo

## 🇮🇹 Temi per ogni lingua

Ogni lingua cambia l'aspetto dell'app ad es. riprendendo i colori della nazione.

## 🧰 Pulsanti lingua dinamici

Genera i pulsanti delle lingue tramite JS partendo da un array di oggetti.

## 🔄 Traduzione random

Impariamo nuove parole inserendo alcuni elementi di casualità. Potresti inserire un tasto per generare una traduzione a caso o persino tradurre una parola a caso con una nuova API.

<https://random-word-api.herokuapp.com/home>

## ★ Salva le traduzioni preferite

Sfrutta la LocalStorage API per salvare le tue traduzioni preferite e tenerle a portata di mano (inventa tu la UI!).





# Dev Hints



# Dev Hints

## Qualche spunto per le challenge!

### JS - Math.random()

Metodo che ritorna un numero casuale compreso tra 0 (incluso) e 1. Opportunamente utilizzato può restituire un *numero random da 0 a n*.

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Math/random](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random)

### CSS - background-image

Proprietà utilizzata per specificare un'immagine di sfondo per un elemento HTML. Accetta come valore il percorso o url dell'immagine da utilizzare come sfondo.

<https://developer.mozilla.org/en-US/docs/Web/CSS/background-image>

```
1 .element {  
2     background-image: url('path/to/image.jpg');  
3 }
```



```
1 const array = ['item 1', 'item 2', 'item 3'];  
2 const randIndex = Math.floor(Math.random() * array.length);  
3 const randItem = array[randIndex];  
4 console.log(randItem);
```

### JS - Array di oggetti

Combine le struttura dati conosciute per crearne una più complessa che contiene più oggetti all'interno di un singolo array.



```
1 const languages = [  
2     { lang: "es", flag: "🇪🇸" },  
3     { lang: "fr", flag: "🇫🇷" }  
4 ];
```



# Deploy

1. Registrati Netlify, in questo modo il tuo progetto sarà pubblicato per un tempo illimitato e non richiederà una password
2. Trascina la cartella del tuo progetto dentro Netlify drop
3. Ottieni il link generato e condivisibile

E' fatta!

<https://app.netlify.com/drop/>

