

---

# METODOLOGIE DI PROGRAMMAZIONE PER IL WEB

## JAVASCRIPT NEL BROWSER

In questo terzo laboratorio, aggiungerai dei comportamenti “dinamici” all’interfaccia web realizzata nel laboratorio precedente. Per farlo, utilizzerai JavaScript (a oggetti), la manipolazione del DOM e i form.

### ESERCIZIO 0 - ORGANIZZAZIONE

Prima di partire, pensa a come vorresti organizzare le classi JavaScript da utilizzare nel tuo progetto. Ti suggerisco di seguire la seguente struttura di file/decomposizione di classi:

- `main.js` - lo script che fa partire l’applicazione JavaScript. Crea solo un oggetto *App*.
- `app.js` - l’oggetto *App* incapsula l’intero stato dell’applicazione. Crea la lista di task (**Esercizio 1**) e l’oggetto seguente per rappresentare la funzionalità di filtri (**Esercizio 2**).
- `filters.js` - genera e gestisce le varie viste mostrate quando un utente clicca su un filtro specifico.

Dovresti anche creare un oggetto *Task*, che rappresenti le proprietà di un singolo task. La lista di task, con i suoi metodi, può sia essere in una classe dedicata (per esempio, *TaskManager*) oppure in *App*.

Infine, usa la libreria `moment.js` (<https://momentjs.com>) per gestire le date e il tempo: scarica la libreria dal sito e includila nel file HTML, tra gli altri script.

Riempi le classi necessarie solo quando ne hai bisogno, seguendo le indicazioni dei prossimi esercizi. Non scrivere codice JavaScript direttamente nel markup HTML, ma importa i file `.js` nella pagina web.

*Suggerimenti:*

1. Se preferisci, usa la soluzione disponibile per il Lab 2 come punto di partenza: <https://github.com/luigidr/2020-metweb-lab2-html-css>
2. Se necessario, continua a utilizzare le classi e le funzionalità di Bootstrap 4.4 per stilizzare la pagina web: <https://getbootstrap.com>
3. Per un esempio, puoi riferirti all’esercizio svolto in settimana: <https://github.com/luigidr/2020-metweb-es-js-browser>

### ESERCIZIO 1 - RIEMPIRE LA LISTA DI TASK

Partendo dallo svolgimento del laboratorio precedente, rimpiazza la lista “statica” di tutti i task presenti nel “contenuto principale” (area destra della pagina) con una lista generata da un array JavaScript (nella classe *TaskManager* o in *App*).

L’array JavaScript conterrà diversi oggetti di tipo *Task*, con tutte le loro proprietà (descrizione, importante, scadenza, nome del progetto, privato o condiviso). Per ora, considera la proprietà “privato/condiviso” come un *flag*. In aggiunta ogni task deve avere associato un *identificatore univoco* (cioè, una proprietà “id”).

La lista nella pagina web dovrà mantenere lo stesso aspetto di prima: i nomi dei progetti saranno in un'etichetta colorata, i task importanti avranno dei punti esclamativi davanti, ecc. Per l'obiettivo di questo lab, tutte le etichette dei progetti possono avere lo stesso colore.

## ESERCIZIO 2 - FILTRI!

Fai funzionare i filtri, estendendo l'esercizio precedente. In particolare, devi abilitare le azioni dei seguenti filtri, presenti nella sidebar sinistra della pagina HTML:

- **Tutti**, per mostrare *tutti* i task (come nell'Esercizio 1).
- **Importanti**, per mostrare solo i task etichettati come *importanti*.
- **Oggi**, per visualizzare i task la cui scadenza è *oggi* (per esempio, 8 Maggio 2020).
- **Prossimi 7 giorni**, per mostrare i task la cui scadenza è tra *domani e i prossimi 7 giorni* (per esempio, da sabato 9 Maggio a venerdì 15 Maggio, estremi inclusi).
- **Privati**, per visualizzare i soli task etichettati come *privati*.
- **Condivisi con...**, per mostrare i task *condivisi* con qualcuno.

Tutte queste azioni aggiorneranno la lista di task visualizzata nello stesso "contenuto principale" popolato nel primo esercizio, cioè senza creare una nuova pagina o una nuova area. In altre parole, questo creerà una vista diversa della stessa lista di task.

Aggiungi alcuni task adatti a testare i filtri, all'array JavaScript creato nell'esercizio precedente.

## ESERCIZIO 3 - AGGIUNGERE UN TASK

Aggiorna l'applicazione web sviluppata finora per permettere a un utente di inserire un nuovo task in maniera *dinamica e interattiva*.

Per far ciò, usa il componente Modal di Bootstrap: quando l'utente clicca sul bottone '+', apri un modale e chiedi le informazioni necessarie a creare un nuovo task, riempiendo un form. Sottomettendo il form, aggiungi il task appena inserito nell'array JavaScript.

**Attenzione:** i campi dei form vanno validati, per rinforzare i campi obbligatori e per evitare di avere valori non ammessi.

*Suggerimenti:*

1. Puoi trovare la documentazione del componente Modal all'indirizzo:  
<https://getbootstrap.com/docs/4.4/components/modal/>

## ESERCIZIO OPZIONALE - PROGETTI

Fai funzionare il menu dei progetti. In particolare, dovresti generare la lista dei progetti (nella sidebar di sinistra) leggendo la lista di task in JavaScript. Cliccando sul nome di un progetto, la lista nel "contenuto principale" si aggiornerà, proprio come capita per i filtri.