

# Dev & Test

## 2- #RUN

Il tuo team sta sviluppando un programma per elencare i top runners di una gara, e come sviluppatore front-end, il tuo ruolo e' quello di implementare l'applicazione web. L'app dovrebbe funzionare secondo le seguenti specifiche:

### # RUN

Filter leaderboard by category All categories Filter

When the user click, the list is updated to display only the runners of the selected category

Here we display the top 3 runners

	<b>Carlo pro</b>	Time 30 mins	Selfies 10	#1
	<b>Amanda pro</b>	Time 38 mins	Selfies 7	#2
	<b>Beatrice amateur</b>	Time 40 mins	Selfies 5	#3

Here we display the rest

Fabrizio (pro)	Time 30 mins	Selfies 10	#4
Jayden (pro)	Time 30 mins	Selfies 10	#5

I runners devono essere caricati dinamicamente dall'array **data**, che possiede la struttura:

```
var data = [
{
  name : "Carlo",
  pic : "img/runner1.jpg",
  time : 30,
  selfies : 10,
  category : "pro"
},
{ .. }, ...
]
```

1. All'apertura della pagina vengono mostrati i runners di tutte le categorie. Questi sono caricati dall'array **data** che ha la struttura descritta sopra.
2. L'utente puo' scegliere una categoria, e cliccare su "Filter" per visualizzare i top runners della categoria scelta.
3. Se l'utente sceglie "All categories" si cancellano i filtri e vengono mostrati tutti i runners.

## Guidelines

- Scaricare il codice di base da <https://github.com/mbaezpy/trentose-run.git>
  - ❑ index.html -> il documento principale
  - ❑ styles.css -> dove dovete mettere il css
  - ❑ app.js -> qui dovete scrivere la logica della vostra app
  - ❑ data.js -> qui si trovano i dati di prova. Non modificare questo file.
  - ❑ mockup -> screenshot del mockup

## Dev & Test

### 3- #COACH

Questa libreria implementa servizi che permettono di stabilire un obiettivo d'allenamento, tracciare le attività sviluppate e monitorare il proprio progresso.

#### **Coach.start(program) ;**

Permette di inizializzare un programma d'allenamento che stabilise il numero di calorie da bruciare (goal). Riceve in input il codice del programma (tabella sotto), e restituisce *false* se il codice non e' valido, *true* nel caso contrario.

#### **Coach.track(activity, distance) ;**

Riceve in input il codice di un'attività (tabella sotto) e la distanza (in km), calcola le calorie bruciate per quell'attività, e accumula il totale per il piano attuale d'allenamento.

Restituisce il numero di calorie bruciate nell'attività attuale, null se l'attività non e' valida.

#### **Coach.checkProgress() ;**

Restituisce il progresso (%) nel piano attuale d'allenamento (tot bruciato / goal calorie del programma). Il formato e' 0-100 senza decimali (arrotondato).

#### **Example of Use, and test data**

```
Coach.start("easy")      // set 500 as goal
> true
Coach.track("walk", 1)   // 50 * 1 = 50   // tot = 0 + 50
> 50
Coach.track("bike", 2)   // 100 * 2 = 200 // tot = 50 + 200
> 200
Coach.checkProgress()    // 250 / 500 -> x 100 and rounded.
> 50
```

Activity	calories-km
walk	50
bike	100
run	150

Program	calories
easy	500
normal	1000
crazy	2000

#### **Vi chiediamo di:**

Scrivere casi di test in javascript (con il framework jasmine) per le funzioni precedenti. Il codice di base lo puoi scaricare da <https://github.com/mbaezpy/trentose-run.git>