

MHW2: quiz di personalità

In questo homework, dovrete creare un quiz di personalità. Ovvero:

- Scegliete un argomento.
- Scegliete tre domande relative all'argomento.
- Per ogni domanda, scegliete nove possibili risposte sotto forma di immagini.
- Ogni risposta corrisponde a una tra nove possibili "personalità"; alla fine del quiz, mostrate la personalità più frequente (oppure, se le scelte sono tutte diverse, mostrate la personalità corrispondente alla prima scelta).
- Le risposte sulla personalità sono già fornite. Non importa se non sono inerenti alle risposte scelte (ma siete comunque liberi di modificarle a piacimento).

Il codice HTML e CSS fornito contiene già buona parte della soluzione. Dovrete modificare i seguenti file per completare l'homework:

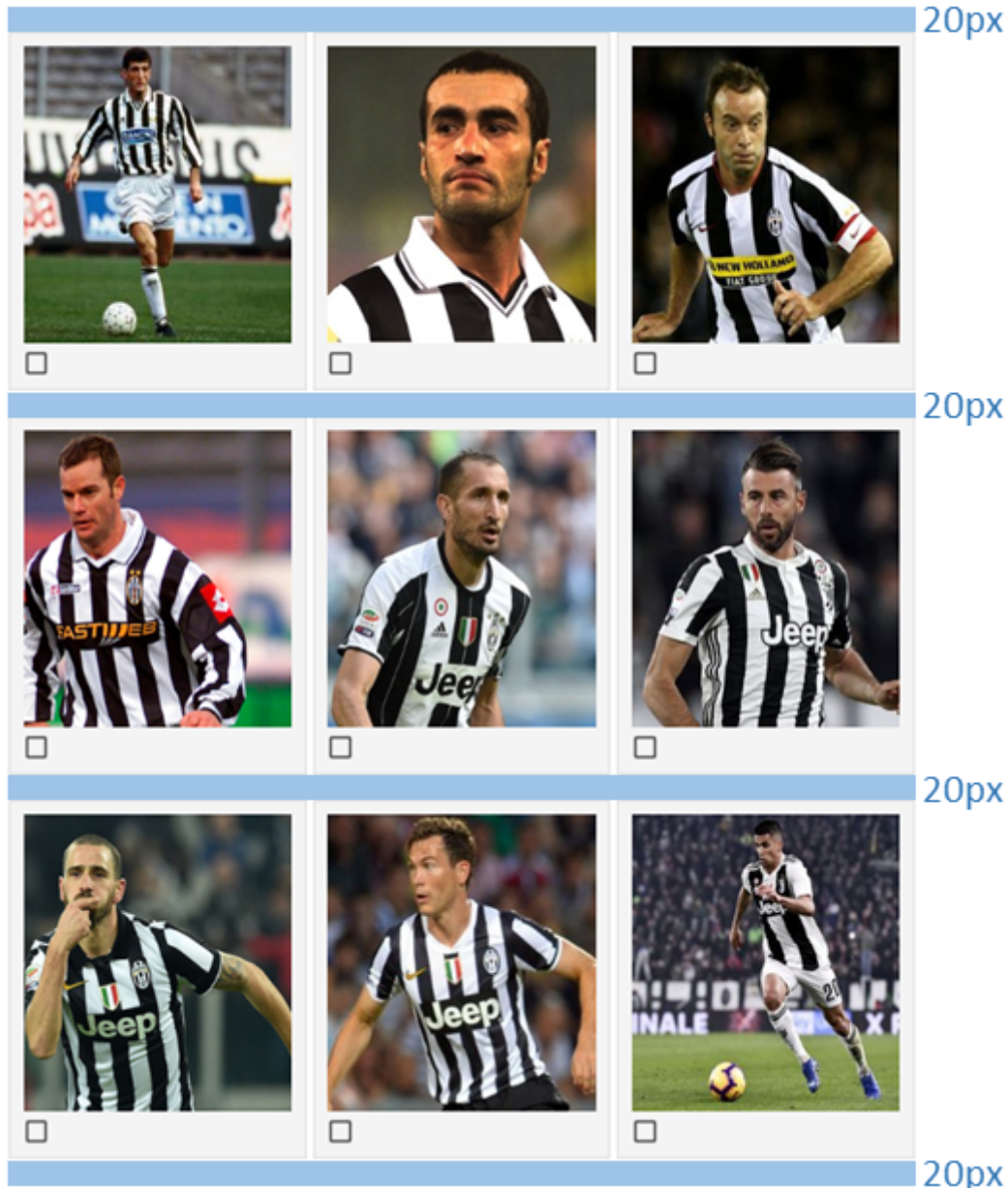
- `index.html` : Buona parte dell'HTML è già scritto, ma dovrete effettuare alcune modifiche per il layout mobile e JavaScript.
- `style.css` : Anche qui, buona parte del CSS è già scritto in `provided-style.css`, ma dovrete applicare alcune modifiche. **NON** modificate `provided-style.css`, ma aggiungete le vostre regole in `style.css`.
- `script.js` : Inserite il codice JavaScript qui.

Aspetto e comportamento generale

L'utente deve poter cliccare su ciascun immagine e selezionarla come propria risposta. Cliccare su una risposta ad una domanda per cui è già stata selezionata la risposta deve comportare il deselezionamento della risposta precedente e la selezione della nuova risposta. Quando tutte e tre le risposte sono selezionate, non deve poter essere più possibile modificarle. A quel punto, viene visualizzata la personalità dell'utente.

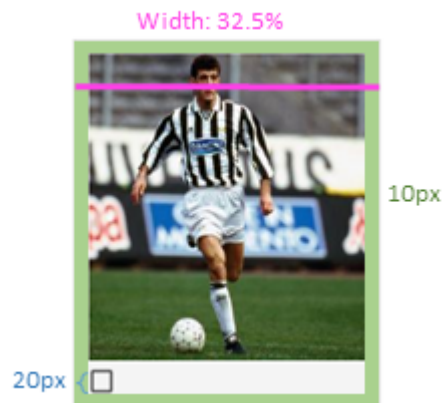
1. Griglia di scelta

Struttura della griglia



- Per ottenere questo layout potete usare una delle tecniche viste nell'esempio sul tic-tac-toe (ad esempio, impostando `flex-wrap: wrap;` sul contenitore).
- Tra ogni riga c'è uno spazio di 20px .
 - **Nota:** in Flexbox non viene applicato il margin collapsing.
- Ogni risposta (flex item) ha una larghezza pari a 32.5% , ed è distribuita su tutto lo spazio a disposizione.
 - **Nota:** la proprietà `width` definisce l'ampiezza del *contenuto*, senza considerare padding, bordo e margine. Potete usare `calc()` per impostare correttamente la larghezza; in alternativa, potete anche vedere come funziona la proprietà [box-sizing](#) .

Risposta non selezionata



- Contenitore:
 - Il colore di sfondo è #f4f4f4 .
 - Il bordo ha spessore 1px e colore #dcdcdc .
 - La larghezza dell'elemento è 32.5% .
 - Lo spazio tra il bordo e il contenuto dell'elemento è 10px .
- Immagine di risposta:
 - L'immagine visualizzata deve avere dimensione quadrata.
 - **Suggerimento:** è possibile utilizzare immagini le cui dimensioni originali non siano uguali, ma è più semplice se sono già quadrate in partenza.
- Checkbox:
 - L'immagine per il checkbox non selezionato è images/unchecked.png .
 - La larghezza e l'altezza dell'immagine è sono pari a 20px .
 - **Nota:** Non usate un `<input type="checkbox">` . Utilizzate l'immagine.

2. Layout mobile

Avrete bisogno di modificare il CSS e l'HTML per implementare il supporto alla vista da mobile

Note: non è necessario caricare la vostra pagina web sul telefono per testare il layout. Utilizzate il *device mode* di Chrome.

- Se la pagina è visualizzata da mobile:
 - Il livello di zoom della viewport deve essere del 100%, e la larghezza deve essere pari alla larghezza del dispositivo.
- Se la larghezza dello schermo è minore di 700px :
 - La larghezza del contenuto della pagina dovrà essere pari a 95% invece di 700px .
 - I cerchi gialli nell'header non devono apparire.
- Se la larghezza dello schermo è minore di 500px :
 - Ogni risposta deve avere larghezza pari al 49% invece di 32.5% .

3. Funzionamento del quiz

Suggerimento: alcuni aspetti del quiz sono simili al comportamento dell'esempio sul tic-tac-toe.

Attributi data

Dovrete utilizzare gli attributi `data` associati alle possibili risposte (cioè, ai corrispondenti elementi HTML in `index.html`):

- `data-choice-id`: Identifica il risultato del quiz per il quale questa risposta "conta" (i possibili risultati sono definiti in `constants.js`).
- `data-question-id`: Identifica il numero della domanda: `one`, `two` e `three`.

In Javascript, questi attributi saranno accessibili tramite:

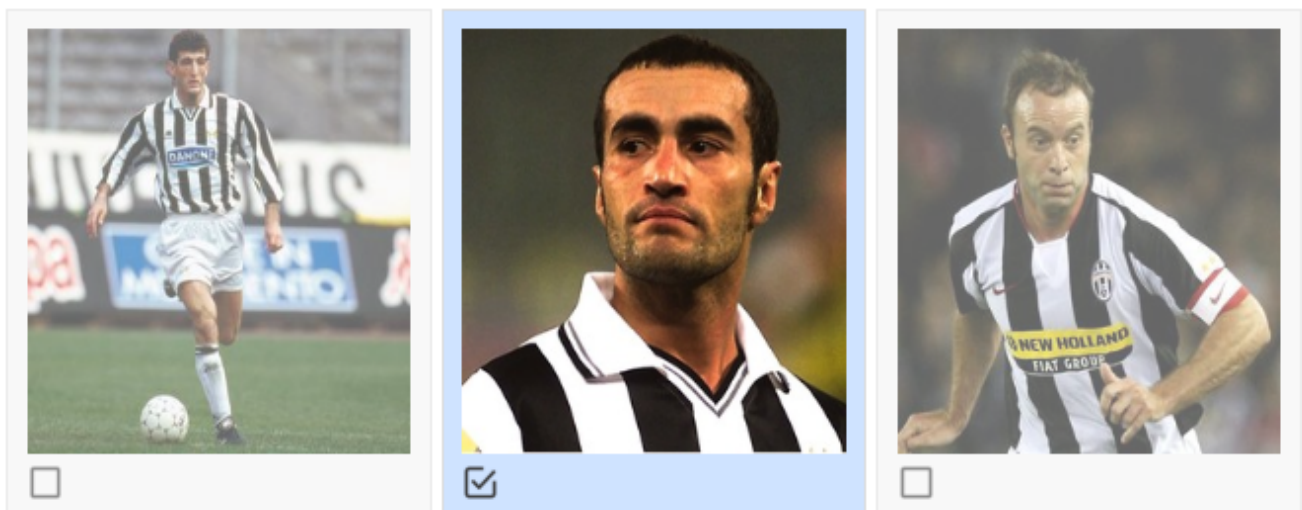
- `element.dataset.choiceId`
- `element.dataset.questionId`

Potete anche selezionare gli elementi HTML corrispondenti a certi valori degli attributi, come in questi esempi:

- `[data-choice-id='blep']`
- `[data-question-id='two']`

Selezionare una risposta

Quando l'utente clicca su una risposta, questa dovrebbe essere visualizzata nel seguente modo:



- Per l'elemento selezionato:
 - L'immagine del checkbox deve diventare `images/checked.png`.
 - Il colore di sfondo è `#cfe3ff`.
- Per gli elementi non selezionati (di quella domanda):
 - Devono essere resi semi-trasparenti (opacità `0.6`).
 - **Suggerimento:** potete utilizzare un overlay per implementare la semitrasparenza; tuttavia, dato che tutto il contenuto del `div` che contiene la risposta deve essere resto semi-trasparente, potete anche utilizzare la proprietà CSS [opacity](#).

Modificare una risposta

Se l'utente non ha ancora completato il quiz (cioè, se almeno una risposta non è stata selezionata), deve poter modificare la propria risposta cliccando su un'altra immagine.

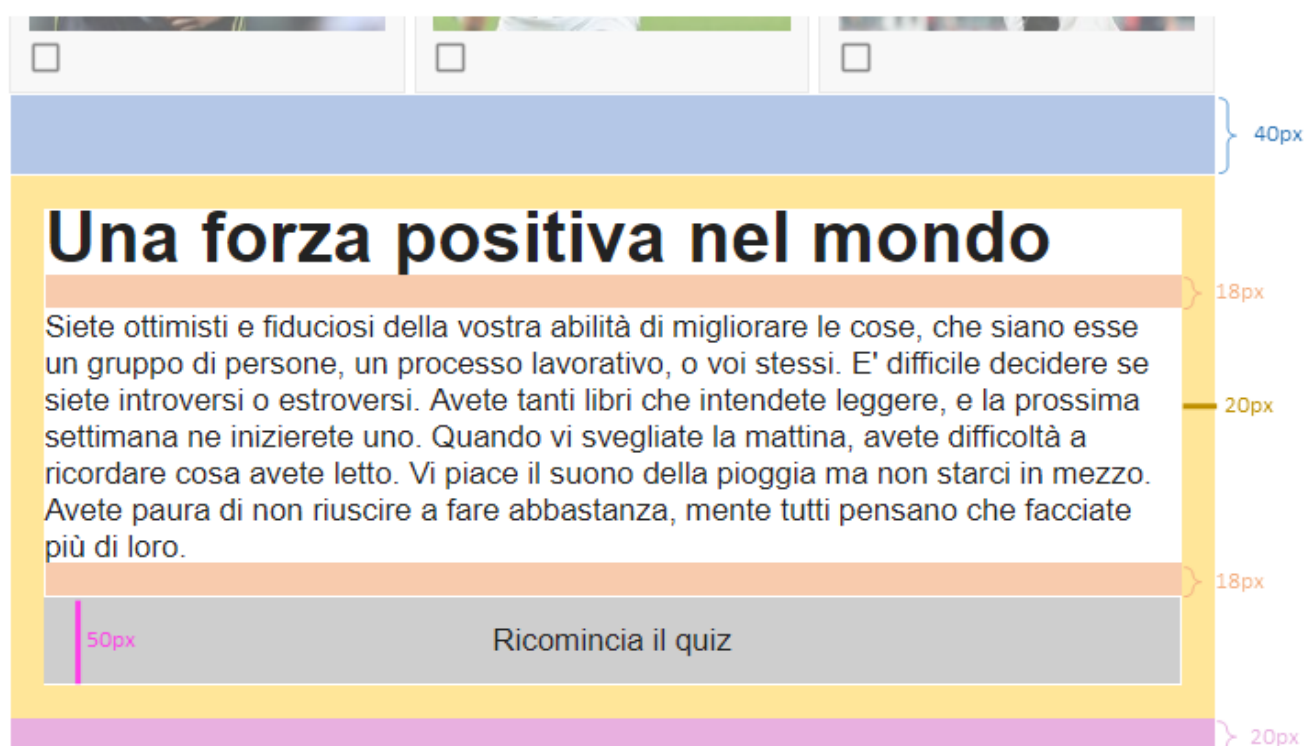
Dopo che l'utente ha risposto a tutte le domande, le risposte non devono poter essere più modificate, finché l'utente non clicchi "Ricomincia il quiz" o aggiorni la pagina.

Completamento del quiz

Dopo che l'utente ha risposto a tutte le domande, il quiz è completo.

I risultati sulla personalità devono apparire in fondo alla pagina. Le informazioni da mostrare sono definite all'interno di `constants.js`.

Di seguito, le specifiche di come il risultato deve essere visualizzato:



- Bottone:
 - Il colore di sfondo è `#cecece`.
 - Quando l'utente muove il mouse su di esso (`hover`), il colore diventa `#e0e0e0`.

Risultato del quiz

Il valore di `data-choice-id` di ogni risposta è mappato ad uno dei possibili risultati definiti da `RESULTS_MAP`, all'interno di `constants.js`. Potete accedere a `RESULTS_MAP` da dentro `script.js` perchè `constants.js` è incluso prima di `script.js` in `index.html`.

Quando il quiz è completo, analizzate i valori di `data-choice-id` di ogni risposta. Ad esempio, se i valori sono `blep`, `sleepy` e `blep`, dovrete visualizzare il titolo e i contenuti della personalità descritta in `RESULTS_MAP['blep']`.

In caso di pareggio, cioè se l'utente sceglie valori unici per `data-choice-id`, come risultato del quiz viene utilizzata la risposta alla prima domanda. Ad esempio, se i valori selezionati sono, in ordine, `burger`, `nerd` e `shy`, dovrete visualizzare i titoli e i contenuti della personalità descritta in `RESULTS_MAP['burger']`.

Ricominciare il quiz

Se l'utente clicca sul pulsante "Ricomincia il quiz", la pagina deve tornare allo stato iniziale:

- Le risposte scelte devono tornare al loro aspetto originario.
- Il risultato sulla personalità deve scomparire.
- Le risposte devono poter essere nuovamente selezionabili.
- La pagina deve sembrare e comportarsi come se aveste ricaricato la pagina (senza aver effettivamente ricaricato la pagina).