

Insegnamento di Tecnologie Web

CdS In Informatica

(A.A. 2022-23)

Esame scritto del 24/07/2023

Nome:

Cognome:

Matricola:

Corso di Studi

Anno di frequenza

Attenzione:

- Questi computer sono limitati ad accedere solo ad alcuni siti: `eol.unibo.it`, `virtuale.unibo.it`, `developer.mozilla.org`, `getbootstrap.com` e `site212248.tw.cs.unibo.it`. Non funzionano Google, stack overflow, etc.
- *Rispondete solo negli spazi delimitati dai blocchi ```` qui la risposta ````, senza modificarli o eliminarli.*
- *Consegnate solo questo file. Copiate ed incollate dentro agli appositi spazi la vostra risposta per intero.*
- *Potete decidere se inserire il CSS inline nel file HTML o metterlo in un file esterno. Nel secondo caso inserite l'elemento nella posizione corretta e mettete il CSS in un blocco separato.*
- *You can use either English or Italian for your answers.*
- *Per favore, per favore, per favore: nessun errore di ortografia. Questa è un'università e non la scuola elementare.*

Domanda #1 - Domande di base (6 punti totali)

a) Accessibilità

Scrivere un form in HTML che rispetti i criteri di accessibilità per ordinare una maglietta colorata. Deve contenere i seguenti input:

- nome e cognome
- indirizzo e-mail
- indirizzo di spedizione
- colore della maglietta (input di tipo color)
- taglia della maglietta (XS, S, M, L, XL...)
- un input testuale per eventualmente stampare testo sulla maglietta.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Magliette accessibili</title>
</head>
<body>
  <form>

  </form>
</body>

</html>
```

b) Codifica caratteri

Quanti byte sono richiesti in UTF-8 per rappresentare le seguenti parole? NB: Le parole in Tamil e in Armeno contengono (senza contare gli spazi) 11 caratteri.

- Napoletano: bònà jurnàta
- Svedese: ha en bra dag
- Tamil: கால வணக்கம்
- Basco: Egun on
- Armeno: Բարի առավոտ
- Bielorusso: Добрай раницы

c) CSS

Scrivere un selettore CSS che seleziona solo gli elementi dispari del secondo elemento all'interno del body.

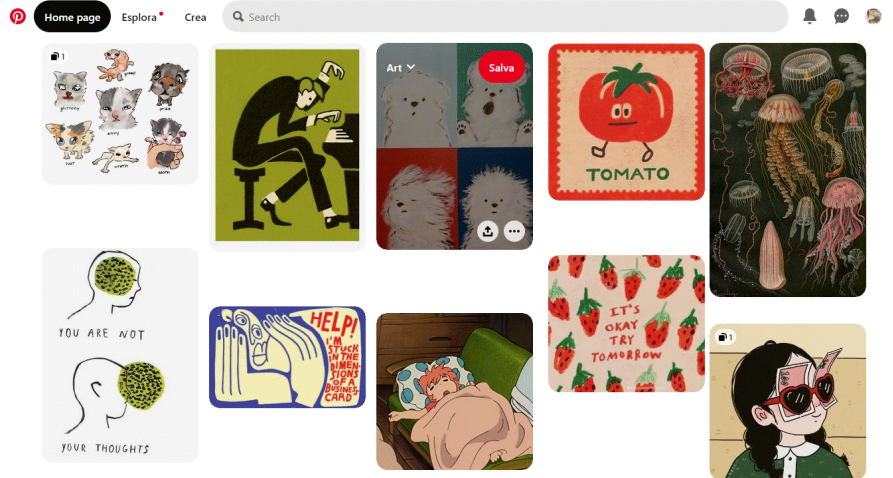
```
<body>
  <header>
    <h1>Magliette Accessibili</h1>
    <nav>
      <h2>Navbar</h2>
    </nav>
  </header>
  <main>
    <p>Maglietta blu</p>
    <p>Maglietta gialla</p>
    <p>Maglietta nera con scritto "Pippo"</p>
    <p>Maglietta bianca multicolore (finita per sbaglio coi rossi)</p>
    <p>ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn</p>
    <p>Maglietta rossa</p>
  </main>
  <footer>
  </footer>
</body>
```

d) Javascript

Spiegare il concetto di prototipo in JavaScript e come funzionano all'interno degli oggetti. Fornire un esempio.

Domanda #2 - HTML + CSS (10 punti)

Scrivere il codice HTML e CSS cercando di riprodurre la seguente pagina web:



Le immagini sono disponibili in SVG (pinterest_logo.svg); quasi tutte le immagini della bacheca si trovano nello zip scaricabile da EOL (non è necessario averle tutte, se ne può usare anche solo una). E' possibile utilizzare Bootstrap. Per le icone presenti sulla navbar a destra possono essere utilizzati emoji o testo.

- Notification: 🔔
- Message: ✉

Codice HTML (ed eventualmente CSS interno)

Codice CSS (solo se esterno)

Domanda #3 - Javascript (12 punti)

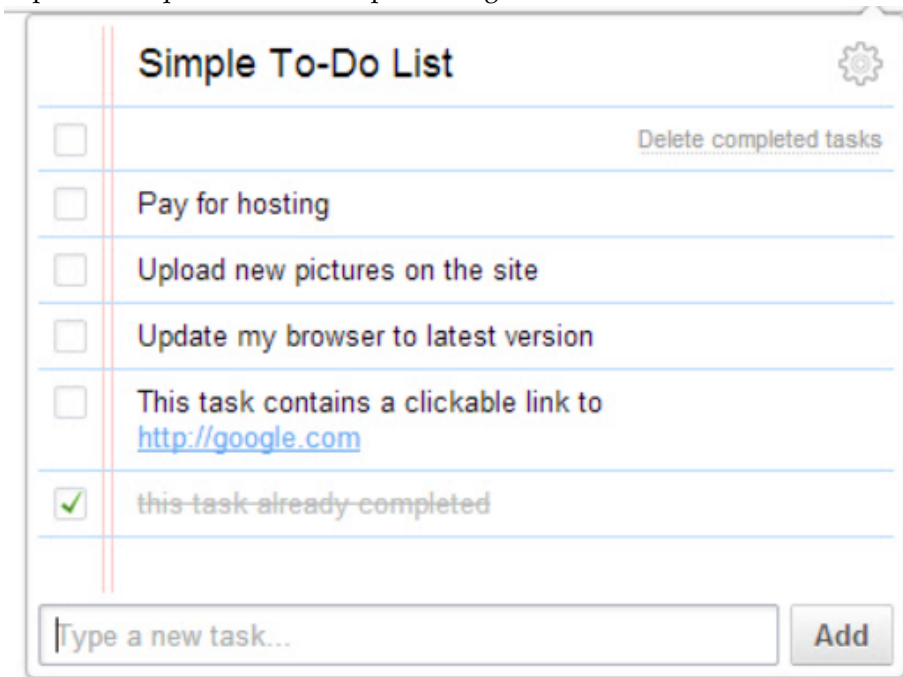
Scrivere il codice JS per fare suonare una batteria. Il template e i materiali di partenza sono presenti nella cartella `DrumKit.zip` su EOL.

- I tasti delle immagini devono essere mappati ai bottoni, ovvero si deve poter interagire con la batteria sia tramite click che tramite tastiera.
- Quando uno dei tasti viene premuto oppure cliccato, bisogna assegnare all'immagine la classe 'pressed' per generare una animazione.
- Si può testare in `index.js` il codice JS, ma tutto il codice deve essere incollato qui sotto.
- Il codice verrà testato nella valutazione.
- Tip: per utilizzare un file audio, usare il costruttore `Audio()` in JS.

Domanda #4 - Framework (6 punti)

Creare una semplice lista di attività (*To-Do List*) utilizzando un framework web a vostra scelta. La lista deve consentire all'utente di aggiungere nuove attività, contrassegnarle come completate e infine rimuoverle. Bisogna interagire con la lista senza ricaricare la pagina.

- Si parta da un framework a vostra scelta, e.g. React, Angular, Vue, Svelte, etc.
- E' possibile suddividere la soluzione in più componenti, ma devono essere tutti richiamati nel componente padre.
- Il componente riceve un array di oggetti immagine di partenza; ogni oggetto è composto dall'URL dell'immagine, titolo e descrizione.
- La galleria mostra un'immagine alla volta. Oltre all'immagine, titolo e descrizione, ci deve essere un modo per navigare tra un'immagine e l'altra.
- Si può usare questo screenshot per immaginare il risultato:



Punti chiave:

- Si assuma di poter partire dalla configurazione standard di un progetto
- Si implementi la lista partendo da una lista vuota; deve essere possibile aggiungere nuove attività, e subito vederle come disponibili sulla lista.
- L'utente deve poter creare un nuovo item e metterlo in lista. Deve esserci dunque sia l'input per un nuovo list item sia un pulsante per aggiungerlo.

- Gli item nella lista devono una checkbox per contrassegnare l'attività come completata.
- Un pulsante deve permettere la rimozione di tutti gli item completati.
- *EXTRA* mostrare la data e l'orario in cui è stato aggiunto un list element
- *EXTRA*: suddividere gli item della lista per data di aggiunta
- *EXTRA*: quando un item viene creato, dare la possibilità di poter aggiungere la data di scadenza dell'attività.