

Esame di Tecnologie Web

CdL in Informatica per il Management - a.a. 2020/2021

Data Appello: 24/06/2021 - turno I

Tempo a disposizione: 2h

INFORMAZIONI PRELIMINARI

L'esame consiste nella scrittura di codice HTML, CSS e Javascript ed esercizi su API ed altri argomenti del corso. Scrivere il codice usando un editor di testo (jEdit o Gedit). Per testare il risultato usare il browser Firefox o Chromium.

La scelta dei nomi dei file, se non espressamente indicati, non è rilevante. I file possono essere salvati in directory diverse. E' possibile comprimere file/directory e decomprimere.

Consegnare i file su IOL. Non consegnare file temporanei e nominare i file in modo chiaro (i numeri degli esercizi devono corrispondere!).

NON è consentito l'uso di testi e appunti cartacei.


E' consentito il solo accesso a Virtuale. NON è consentito l'accesso ad altre risorse on-line.

NOTE TECNICHE

Risorse utili allo svolgimento della prova:

- Librerie jQuery, Bootstrap: <http://diiorio.nws.cs.unibo.it/twe/lib/>
- Risorse e immagini: <http://diiorio.nws.cs.unibo.it/twe/resources/>

Combinazioni di tasti e note sulle applicazioni:

- Tasto "Windows" : per aprire menù con le applicazioni disponibili
 - *Nautilus*: esplora file
 - *Firefox/Chromium*: browser
 - *jEdit, GEdit*: editor di testo
- ALT+TAB: per passare da un'applicazione all'altra
- CTRL+C, CTRL+V: copia&incolla
- CTRL+R: ricarica pagina nel browser
- CTRL+S: salva file
- Per cambiare le impostazioni del testo in jEdit: menù "Preferences"->Global Options->jEdit->Textarea

NOTA LOGISTICA

Agli studenti affetti da disturbi quali dislessia, è richiesto di completare tutti gli esercizi in un tempo maggiorato del 30%.

Nota per lo studente: Per sicurezza, indicare "studente con DSA" al momento della consegna.

TESTO DEL COMPITO

Esercizio 1

Progettare un API REST (parziale) per la gestione di una spiaggia. La spiaggia è organizzata in stabilimenti balneari, ognuno individuato da un numero intero (progressivo), un nome e una descrizione. Ogni stabilimento è formato da ombrelloni numerati. Ogni ombrellone quindi è individuato da un numero intero e può essere *'libero'*, *'occupato giornaliero'* o *'occupato stagionale'*.

Scrivere un file in formato JSON o YAML.

L'API permette di:

- aggiornare la descrizione di uno stabilimento
- ottenere l'elenco di tutti gli ombrelloni liberi in un dato stabilimento
- aggiungere un ombrellone ad uno stabilimento; l'ombrellone appena aggiunto è in stato *'libero'*

Specificare: URL di accesso, metodi HTTP, parametri e risposte con esempi.

L'API restituisce un errore, con codice 400, se i parametri in input non sono corretti.

Note:

- Non è richiesto includere le sezioni *host*, *schemes*, *servers*, *tags*
- Non è richiesto gestire autenticazione

Esercizio 2

1. Scrivere il codice HTML e CSS, in un file esterno, per ottenere la visualizzazione mostrata in Figura 1 quando il documento viene visualizzato in un browser.
Oltre alle caratteristiche tipografiche già evidenti in figura, si tenga presente che:
 - l'area principale occupa la metà della larghezza del viewport ed ha bordi arrotondati (misure esatte non rilevanti)
 - le righe della tabella hanno sfondo verde se la postazione è libera, rosso se non disponibile; la riga di intestazione ha sfondo nero con testo in bianco
 - il menù mostra l'elenco delle sole postazioni libere (nell'esempio non contiene la voce 'Primissima' ma solo 'Standard' e 'Love dogs');
 - si può affittare per un numero massimo di 7 giorni
 - il prezzo è visualizzato in basso a destra, in un'area inclinata e con bordo doppio e di colore rosso; font, distanze e inclinazione esatta non rilevanti ma è sufficiente approssimarle
 - le misure esatte di margini, bordi e padding non sono rilevanti, così come i colori
2. Includere l'elemento HEAD con le informazioni appropriate

Vincoli:

- non è ammesso usare attributi @id e @class nel sorgente HTML;
- non è ammesso usare tabelle per organizzare il layout del form, ma solo per i dati
- non è possibile usare Javascript (eventuali comportamenti dinamici vanno nell'esercizio successivo).

Risorse:

Immagini disponibili su: <http://diiorio.nws.cs.unibo.it/twe/resources/2406a/>

Figura 1 - Resa della pagina in un browser:

Nome	Descrizione	Prezzo (giornaliero)	Disponibilità
Standard	Ombrellone con due lettini	20 €	✓
Love dogs	Standard, due lettini e spazio dedicato al tuo cane	40 €	✓
Primissima	Prima fila, formato capanna, tre posti	50 €	✗

Calcola il tuo preventivo e prenota!

Postazione

Giorni

Cliente

160 €

Esercizio 3

Data la pagina HTML dell'esercizio precedente, in cui l'area con le informazioni relative agli ombrelloni è inizialmente vuota, si realizzino alcuni comportamenti dinamici in Javascript.

Vincoli:

- è ammesso l'uso di jQuery
- è ammesso usare CSS e immagini dell'esercizio precedente

Nota:

- Scrivere la soluzione in un file `esercizio3.html` diverso dal precedente

In particolare si realizzi:

1. Al caricamento, la pagina accede asincronamente in GET al servizio web <http://diiorio.nws.cs.unibo.it/twe/api/bagni/bagni.php>, ottenendo un JSON con i dati delle postazione, e li visualizza in modo appropriato in forma tabellare:

```
[
  {
    "id":"p1",
    "nome":"Standard",
    "descrizione":"Ombrellone con due lettini",
    "prezzo":20,
    "libero":true
  },
  {
    "id":"p2",
    "nome":"Love dogs",
    "descrizione":"Standard, due lettini e spazio ...",
    "prezzo":40,
    "libero":true
  },
  {
    "id":"p3",
    "nome":"Primissima",
    "descrizione":"Prima fila, formato capanna...",
    "prezzo":50,
    "libero":false
  }
]
```

Note sulla visualizzazione:

- Riprodurre la visualizzazione dell'esercizio precedente e costruire dinamicamente la tabella e il menù con le opzioni
 - Il menù con le postazioni da affittare include solo le postazioni libere (proprietà "libero" nei dati JSON)
 - i dati JSON di esempio includono tre tipi di postazioni ma il codice deve funzionare anche per un numero diverso.
2. Alla selezione di una postazione o alla modifica del numero di giorni richiesti viene aggiornato il prezzo (con il prodotto: *prezzo giornaliero x giorni affitto*)

Esercizio 4

Scrivere una funzione Javascript che prende in input un vettore di parole (stringhe) e restituisce una stringa contenente le iniziali di queste parole, in ordine alfabetico, senza duplicati, in minuscolo e separate da una virgola.

Ad esempio:

- Input: ["italia", "germania", "Portogallo", "SPAGNA", "SVEZIA", "Grecia"]
- Output: "g,i,p,s"

- Input: ["estonia", "albania", "Grecia", "SPAGNA", "SVEZIA", "Galles", "svizzera"]
- Output: "a,e,g,s"

Scrivere una semplice pagina HTML, la cui formattazione non è rilevante, che al caricamento mostra esempi di invocazione della funzione e stampa del risultato.

I dati possono essere stampati in console, o con un alert o aggiunti alla pagina.