

Esame di Tecnologie Web

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

Data Appello: 10/02/2022 - turno 1

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 EOL. 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 un ristorante. Il ristorante è organizzato in *sale*, ognuna individuata da un nome (stringa) e con un numero massimo di coperti (valore intero); ogni sala può contenere diversi *tavoli* ognuno identificato da un numero (valore intero) e con un numero di coperti (anche questo valore intero).

Scrivere un file in formato JSON o YAML.

L'API permette di:

- ottenere l'elenco di tutte le sale che possono contenere almeno X coperti
- aggiungere un tavolo in una sala; l'API prevede anche una risposta per il caso in cui non è possibile aggiungere il tavolo se non c'è posto
- rimuovere un insieme di tavoli da una sala; il parametro in input è quindi un vettore di interi, identificativi dei tavoli

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

Scrivere il codice HTML e CSS per ottenere la visualizzazione mostrata in Figura 1 (per semplicità non considerare le differenze tra dispositivi).

Oltre alle caratteristiche tipografiche già evidenti in figura, si tenga presente che:

- l'area principale occupa l'intera larghezza del viewport ed è divisa in due parti che occupano rispettivamente i 2/3 e 1/3 dello spazio in larghezza
- cliccando sul titolo di una notizia di apre la pagina corrispondente; creare un link arbitrario, NON è richiesto creare anche le pagine relative alle notizie
- cliccando sulle aree a destra (Eventi, Pubblicazioni) si accede alle pagine corrispondenti; le destinazioni dei link non sono rilevanti
- la barra in alto con sfondo scuro è in posizione fissa anche dopo lo scrolling dei contenuti
- colori e dimensioni esatte di margini e padding non rilevanti, purchè appropriate

Vincoli:

- non è ammesso usare attributi @id e @class nel sorgente HTML
- NON è ammesso usare tabelle per organizzare il layout
- non è possibile usare Javascript (eventuali comportamenti dinamici vanno nell'esercizio successivo).

Risorse:

Immagini disponibili su: <http://diiorio.nws.cs.unibo.it/twe/resources/1002/>

Figura 1 - Resa della pagina in un browser:

Un sito ufficiale dell'Unione europea

Notizie, eventi e pubblicazioni

COMUNICATO STAMPA | 8 Febbraio 2022
Sovranità digitale: la Commissione propone una legge per far fronte alle carenze di semiconduttori e rafforzare la leadership tecnologica dell'Europa

COMUNICATO STAMPA | 4 Febbraio 2022
Sicurezza alimentare: riparte il commercio UE-USA di molluschi bivalvi

COMUNICATO STAMPA | 2 Febbraio 2022
Tassonomia dell'UE: la Commissione presenta un atto delegato complementare sul clima per accelerare la decarbonizzazione

[Tutte le notizie >](#)

Eventi

Prossimi eventi organizzati dalla Commissione europea.

Pubblicazioni

Pubblicazioni dell'UE accessibili da EU Bookshop e altre fonti.

Esercizio 3

Data una pagina HTML, con la stessa struttura della precedente ma in cui l'area centrale con le notizie e l'area a destra sono entrambi vuoti, si realizzino i comportamenti dinamici Javascript descritti sotto.

Vincoli:

- è ammesso l'uso di jQuery
- è ammesso usare attributi @id e @class
- è ammesso (e consigliato!) 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 della pagina si accede asincronicamente in GET ai servizi web <http://diiorio.nws.cs.unibo.it/twe/api/eu/news.php> e <http://diiorio.nws.cs.unibo.it/twe/api/eu/topics.php>, ottenendo i dati JSON con le notizie e i contenuti dell'area a destra. La pagina viene quindi popolata riproducendo la visualizzazione dell'esercizio precedente.
2. Cliccando su una delle aree a destra (Eventi, Pubblicazioni) lo sfondo cambia colore e diventa bianco; cliccando di nuovo si torna al colore originale e così via; se si clicca sul titolo dell'area invece si apre la pagina corrispondente

Altre note:


- generare dinamicamente le parti di pagina relative a tutti i dati contenuti nei file JSON
- l'etichetta "COMUNICATO STAMPA" è in maiuscolo
- le notizie sono visualizzate in ordine di apparizione del file JSON
- il link delle pagine relative ad ogni notizia sono calcolati dall'ID della notizia, nella forma `/news/<ID>/` ;
- le destinazioni dei link nelle voci di menù a destra (Eventi, Pubblicazioni) sono letti dal file JSON

Esercizio 4

Scrivere il codice HTML, CSS e Javascript per ottenere la seguente visualizzazione e il comportamento dinamico descritto sotto:

Weekend o lavorativo?

Weekend Lavorativo	
0	0

Ogni volta che l'utente seleziona una nuova data dal calendario, viene verificato se questa è in un weekend o è un giorno lavorativo e viene incrementato il contatore corrispondente.

Vincoli:

- E' ammesso l'uso di jQuery o altri framework

Nota:

- Suggerimento: usare l'oggetto Date() e il relativo costruttore, e il metodo getDay() che restituisce il giorno della settimana di una data
- la formattazione esatta (margini, padding, ecc.) non è rilevante ed è ammesso usare tabelle per visualizzare i dati