

Esame di Tecnologie Web

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

Data Appello: 15/07/2021 - turno II

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 gestire un circuito di palestre e descriverla in Swagger/OpenAPI. Il circuito è formato da un insieme di palestre, ognuna caratterizzata da un nome identificativo (di tipo *stringa*) e una capienza (di tipo intero). Ogni palestra inoltre organizza un certo numero di attività (ad esempio fitness, pilates, ecc.), identificate a loro volta da un nome identificativo e una descrizione, entrambe stringhe.

Scrivere un file in formato JSON o YAML.

L'API permette di:

- elencare tutte le palestre che hanno una capienza superiore a X posti
- aggiungere un'attività in una palestra
- cancellare un'attività in una palestra

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, interno o esterno, per ottenere la visualizzazione mostrata in Figura 1 quando il documento è caricato in un browser.

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

- L'area a sinistra per la ricerca occupa il 30% della larghezza del viewport, il restante 70% è occupato dall'area centrale
- Il campo del form 'Playing position' può assumere i valori 'goalkeeper', 'defender', 'midfield', 'forward'
- L'area centrale include un testo e tre immagini disposte come in figura
- NON è richiesto copiare il testo descrittivo dei calciatori, usare un testo arbitrario
- Colori e dimensioni esatte di margini e padding non rilevanti

Vincoli:

- non è ammesso usare attributi @id e @class nel sorgente HTML;
- non è ammesso tabelle di layout
- non è ammesso usare Javascript (eventuali comportamenti dinamici vanno nell'esercizio successivo).

Risorse:

Immagini disponibili su: <http://diiorio.nws.cs.unibo.it/twe/resources/1507a/>

Figura 1 - Resa della pagina in un browser:



Esercizio 3

Data la pagina HTML dell'esercizio precedente, in cui l'area con l'elenco dei calciatori e la scheda di ognuno è inizialmente vuota, si realizzino alcuni comportamenti dinamici in Javascript.

Vincoli:

- è ammesso l'uso di jQuery
- è ammesso (e consigliato!) usare CSS e immagini dell'esercizio precedente

Nota:

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

Il particolare realizzare:

1. Al caricamento della pagina e al cambiamento di ruolo (Playing position), viene invocato asincronicamente in GET il servizio web <http://diiorio.nws.cs.unibo.it/twe/api/euro2/ruoli.php>, si ottiene quindi un JSON con l'elenco dei calciatori e si popola/aggiorna il menù a sinistra; l'elenco contiene l'ID dei calciatori usato per la richiesta successiva (punto 2); l'area centrale è vuota finché non si seleziona un calciatore

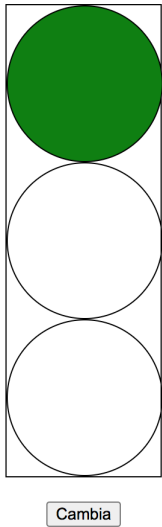
Nota: Il servizio (non implementato come API RESTful) fornisce solo alcuni dati utili per gli scopi dell'esercizio; il codice deve funzionare anche per un numero diverso di calciatori

2. cliccando sull'area relativa ad un calciatore a sinistra, identificato dal proprio ID, viene invocato asincronicamente in GET il servizio web <http://diiorio.nws.cs.unibo.it/twe/api/euro2/calciatori.php?id=XXX> e con i dati ottenuti si popola l'area centrale che descrive il calciatore;
 - le immagini possono essere caricate da:
<http://diiorio.nws.cs.unibo.it/twe/resources/1507a/>

Nota: Il servizio (non implementato come API RESTful) fornisce solo alcuni dati utili per gli scopi dell'esercizio con `id=donnarumma` e `id=neuer` o un messaggio di errore

Esercizio 4

Scrivere il codice HTML, CSS e Javascript per realizzare un semplice semaforo visualizzato in un browser come mostrato di seguito:



Vincoli:

- È ammesso l'uso di jQuery

Note:

- Il semaforo inizialmente è "verde"; al click sul bottone "Cambia" si passa al giallo (cerchio centrale in giallo e il verde torna in bianco); al successivo click si passa al rosso e così via
- l'immagine mostra il semaforo verde a titolo esplicativo
- le misure esatte delle varie aree non sono rilevanti purchè consistenti e proporzionate