

Esame di Tecnologie Web

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

Data Appello: 15/07/2021 - 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 gestire una catena di alberghi e descriverla in Swagger/OpenAPI. La catena include alberghi, ognuno caratterizzato da un nome identificativo (di tipo *stringa*) e un certo numero di stanze (intero). Ogni albergo offre una serie di servizi extra (ad esempio piscina, sauna, escursioni, ecc.), caratterizzate a loro volta da un nome e una descrizione, entrambe stringhe.

Scrivere un file in formato JSON o YAML.

L'API permette di:

- elencare tutti gli alberghi che hanno almeno X stanze
- modificare la descrizione di un servizio offerto da un albergo
- aggiungere un nuovo servizio tra quelli offerti da un dato albergo

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
 - le proporzioni esatte tra immagini e testo non solo rilevanti, purché appropriate
- NON è richiesto copiare il testo descrittivo dei calciatori, usare un testo arbitrario
- Colori e dimensioni esatte di margini e padding non rilevanti

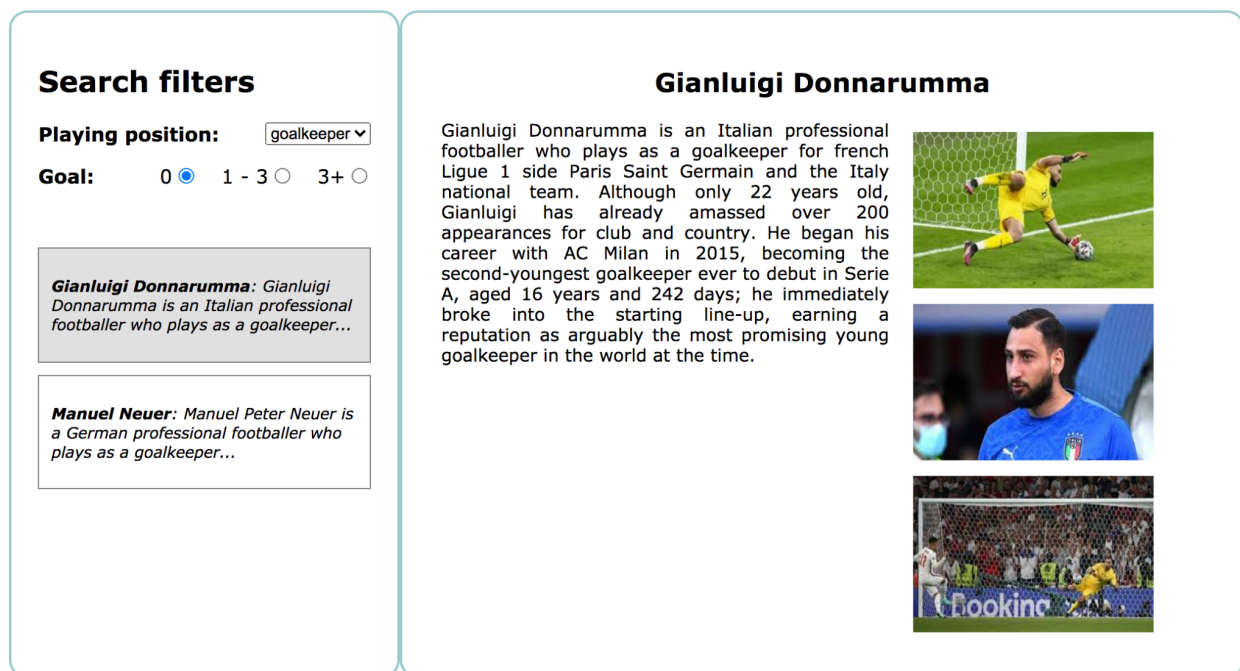
Vincoli:

- 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 qualunque campo del form, viene invocato asincronicamente in GET il servizio web <http://diiorio.nws.cs.unibo.it/twe/api/euro/euro.php?role=XXX&goals=YYY>, si ottiene quindi un JSON con i dati dei calciatori e si popola/aggiorna il menù a sinistra; l'area centrale è vuota finché non si seleziona un calciatore (punto 2)
 - La voce relativa ad un calciatore contiene i primi 85 caratteri della biografia contenuta nei dati JSON

Nota: Il servizio (non implementato come API RESTful) fornisce solo alcuni dati utili per gli scopi dell'esercizio con *role=goalkeeper*, oppure un messaggio di errore. Passare entrambi i parametri in input anche se *goals* è ignorato dall'implementazione del servizio

Nota: i dati JSON di esempio includono i dati di due soli calciatori ma il codice deve funzionare anche per un numero diverso.

2. cliccando sull'area relativa ad un calciatore a sinistra, viene popolata l'area centrale con i dati (testo e immagini) del calciatore
 - le immagini possono essere caricate da:
<http://diiorio.nws.cs.unibo.it/twe/resources/1507a/>

Esercizio 4

Dato il seguente frammento HTML aggiungere il codice Javascript per ottenere il comportamento dinamico descritto sotto:

```
<div>
  <p>Scegli data:
  <input name="data" type="date" value="2021-07-15"></p>
  <p>Sabato: <span id="sabato">0</span></p>
</div>
```

Ogni volta che l'utente seleziona una nuova data viene verificato se questa è un "sabato" e viene aggiornato il contatore mostrato nel paragrafo successivo.

Vincoli:

- NON è ammesso l'uso di jQuery o altri framework

Nota:

- Suggerimento: usare l'oggetto Date() e il relativo costruttore
- La formattazione della pagina non è rilevante