

# Insegnamento di Tecnologie Web

## CdS In Informatica

(A.A. 2021-22)

Esame scritto del del 23/06/2022

*Prova in presenza*

**Nome:**

**Cognome:**

**Matricola:**

**Corso di Studi**

*Solo se diverso da Informatica Triennale*

**Anno di frequenza**

*Come specificato nel piano di studi: o "2021-22" oppure "precedente".*

**Attenzione:**

- Questi computer sono limitati ad accedere solo ad alcuni siti: eol.unibo.it, virtuale.unibo.it, developer.mozilla.org e getbootstrap.com. Non funzionano Google, stack overflow, W3 Schools, 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 (4 punti)

### a) HTTP

“Un metodo HTTP è sicuro (*safe*) se lo si usa all’interno di una connessione HTTPS” Questa affermazione è vera o falsa? Perché?

### b) HTML

Cosa sono gli elementi generici in HTML?

### c) CSS

Dato il frammento `<p class="saluto">Hello world</p>` e le regole CSS:

```
[saluto] {color:red;}  
#saluto {color:blue;}  
saluto {color:green;}
```

, In che colore verranno scritte le parole del paragrafo?

### d) Codifica caratteri

Si faccia un esempio di una parola dell’italiano comune di almeno cinque lettere che abbia una lunghezza diversa in byte a seconda che venga codificata in ISO Latin 1, in UCS-2 o in UTF-8. Si specifichino anche le rispettive lunghezze.

## **Domanda #2 - HTML + CSS (10 punti)**

Scrivere il codice HTML e CSS (bootstrap è ammesso solo se importato correttamente nella pagina) della seguente pagina web. Le immagini sono fornite in uno zip scaricabile da EOL. Il codice deve funzionare su Firefox. Può essere usato come base il documento `base.html`. Non è importante essere totalmente precisi con colori e misure, ma essere ragionevolmente attenti alle differenze tra elemento ed elemento. Se il file non si apre correttamente nel browser, il punteggio è 0.

**Codice HTML** (*ed eventualmente CSS interno*)

**Codice CSS** (*solo se esterno*)



ON-DEMAND: Maximize Site Efficiency with Spot's Newest Capabilities

[WATCH NOW](#)

## Spot

Automate data capture and inspection

[SEE SPOT](#)



## Stretch

Mobile, automated case handling for more efficient warehouse operations.

[LEARN MORE](#)



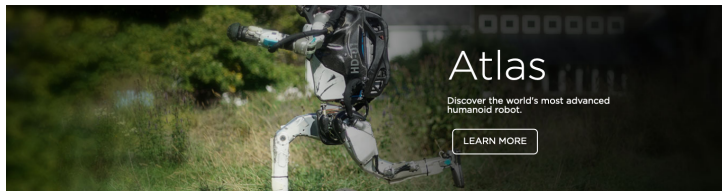
### Spot Inspection Solution

[LEARN MORE >](#)



### Stretch Warehouse Solution

[LEARN MORE >](#)



## Atlas

Discover the world's most advanced humanoid robot.

[LEARN MORE](#)

Figure 1: domanda HTML e CSS

### Domanda #3 - Javascript (10 punti)

Si consideri un sito di un organizzazione di biglietteria per eventi (concerti, spettacoli, eventi sportivi, sagre, ecc.). Alcuni eventi sono individuali, altri appartengono ad una sequenza di eventi individuali equivalenti ma in date e/o luoghi diversi (ad esempio, una tournée dello stesso spettacolo in città diverse, o una sagra che si svolge su più giorni. In questo caso l'utente prima sceglie l'evento, e poi, sulla base della disponibilità, la singola data/luogo. Alcuni eventi hanno un biglietto generico, altri sono collegati ad uno specifico posto a sedere in una griglia predefinita. Per semplicità assumiamo che ogni location sia descritta da una griglia rettangolare di  $n \times m$  posizioni. Ogni posizione può essere associata ad uno e uno solo dei seguenti valori:

- palco (non vendibile)
- non-posto (corridoi, colonne, spazi liberi di ogni tipo)
- posto riservato
- posto già venduto
- posto libero riservato a carrozzine e disabili e in vendita a prezzo 1
- posto libero in vendita a prezzo 2
- posto libero in vendita a prezzo 3
- posto libero in vendita a prezzo 4

Usate il documento allegato, base.html, per fare le vostre prove. Il file contiene, già configurati, un accesso a jquery e a bootstrap. Potete usare i servizi <http://site202100.tw.cs.unibo.it/info> oppure <http://www.fabioitali.it/TW/test/2021/doResponse.php> per effettuare prove e debugging. Non dovete descrivere l'API, solo ipotizzarla con chiarezza e usarla in maniera corretta. Basandosi, dove si ritiene, su uno o più framework Javascript a piacere tra quelli illustrati a lezione:

Il sistema espone quattro servizi:

- <https://www.site.com/listing?from=YYYY-MM-DD&to=YYYY-MM-DD> fornisce un JSON con tutti gli eventi tra le due date specificate. Ogni evento ha un id associato, uno o più campi descrittivi (nome artista, nome spettacolo, cast, etc.). Non specifica né location né date (a parte nella sezione descrittiva).
- <https://www.site.com/show?id=XXX> fornisce l'elenco di date, location dello show XXX (dove XXX è l'id definito precedentemente). Ogni combinazione di show, data e location è associato ad un eventId diverso.
- <https://www.site.com/event?id=XXX> fornisce la mappa dei posti disponibili per l'evento XXX (dove XXX è l'eventId definito precedentemente). Ogni mappa riporta tutti i biglietti disponibili e quelli già venduti per il singolo evento, e i posti liberi (definiti con un placeID) vengono visualizzati sotto forma di elemento cliccabile.
- <https://www.site.com/buy?place=XXX> permette all'utente di acquistare il biglietto relativo al posto in questione.

**Parte I (Solo HTML oppure HTML + Javascript)** Si scriva il codice HTML che contiene la struttura della pagina della applicazione, con la possibilità di scegliere progressivamente show, evento, e posto (si usino in maniera corretta ed appropriata immagini, descrizione, prezzo e giorni disponibili),

**Parte II (Riempire con solo Javascript solo se la parte I contiene solo HTML)** Si scrivano uno o più script Javascript che popolino progressivamente la pagina delle informazioni scelte dall'utente, prima selezionando gli eventi disponibili in un certo intervallo di date, poi scegliendo uno show, poi scegliendo una data e una location, e infine scegliendo il posto.

**Parte III (HTML + Javascript)** Si modifichino l'HTML e gli script in modo che sia possibile acquistare più biglietti in un'unica azione, senza necessariamente pagarli separatamente. I biglietti, tuttavia, sono sempre dello stesso evento nella stessa location e nella stessa data, ma non sono necessariamente contigui di posizione. Il sistema calcola il totale del costo del biglietto prima di eseguire l'acquisto. Si documenti il cambiamento dell'uri dell'API specificata sopra in caso di acquisti multipli. In questa sezione si presentino una copia di HTML e script presentati precedentemente e si applichino le modifiche necessarie per questa aggiunta.

#### **Domanda #4 - Semantic Web (4 punti)**

Scrivere in Turtle il grafo RDF della seguente frase, poi specificare quante triple contiene: «R. Madhavan, pseudonimo di Madhavan Balaji Ranganathan (Jamshedpur, 1° giugno 1970), è un attore e produttore cinematografico indiano.».



### **Domanda #5 - Domanda di accessibilità (5 punti)**

Il form contatti del sito <https://includeretutti.it> contiene diversi campi, tra cui:

- due campi di testo per inserire nome e cognome;
- un campo per scegliere il genere fra le tre opzioni “maschile”, “femminile” e “altro”, da implementare mediante pulsanti radio;
- una textarea per inserire il messaggio;
- una checkbox per indicare di aver letto ed accettare l’informativa sulla privacy;
- il pulsante per inviare il form.

Scrivere il codice HTML della pagina del form contatti rispettando tutti i criteri di accessibilità. Inserire anche i landmark ed i livelli di intestazione necessari.