

## **Esame di Tecnologie Web - parte II**

### **CdL in Informatica per il Management**

**Tempo a disposizione: 2.5 h (incluse domande di teoria)**

**Per gli studenti DSA: vedere varianti di ogni esercizio**

#### **INFORMAZIONI PRELIMINARI**

EOL: <https://eol.unibo.it/>

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 disponibili su EOL.

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

## TESTO DEL COMPITO

### Esercizio 1

Scrivere il codice HTML e CSS per ottenere la visualizzazione mostrata in Figura 1 (pagina successiva) quando il documento è caricato in un browser.

La pagina deve contenere:

#### 1. Header:

- Logo "+Babbel".
- Quattro voci di menu allineate a destra:
  - **Home. Ripasso. Live. Esplora.**
  - NOTA: non è richiesto includere l'icona di fianco a Ripasso

#### 2. Titolo

- Un titolo grande centrato orizzontalmente: "**Si comincia, Utente**"

#### 3. Sezione principale (larghezza 60%):

- Un'area principale che occupa il 60% della larghezza della pagina, divisa in due sezioni:

##### ■ Primo blocco superiore:

- Un titolo "Descrivere la tua città e dire perché ti piace".
- Un carosello, i cui item hanno le seguenti caratteristiche:
  - Uno sfondo di un colore a scelta.
  - Un titolo che descrive il contenuto.
  - Un'immagine in sovrapposizione allo sfondo.
  - Un bottone in basso a destra
  - Larghezza pari al 66% della larghezza del carosello
- NOTA: per semplicità includere nel carosello un solo item (es. "E' il momento di ripassare")
- NON è necessario gestire lo scorrimento, né aggiungere gli indicatori di stato del carosello.

##### ■ Secondo blocco inferiore:

- Un titolo "Corsi" allineato a sinistra.
- Una label "Vedi tutto" allineata a destra.
- Un box rettangolare con sfondo uniforme.
- Tre card cliccabili con un'immagine (puoi usare sempre la stessa, i.e. *person.png*) e una breve descrizione.

#### 4. Interattività:

- Al passaggio del mouse sulle voci e sui riquadri della sezione principale, deve essere applicato un effetto di evidenziazione (ad esempio, cambio di colore o ombreggiatura).
- I link devono essere cliccabili, anche se non è necessario creare pagine di destinazione.

### 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

**Risorse:** Immagini disponibili su EOL

**+Babbel** Home Ripasso • Live Esplora

## Si comincia, Utente

Primi Passi II (A1.2) - Unità 2  
Descrivere la tua città e dire perché ti piace

Esercizi con i vocaboli

È il momento di ripassare



Ripassa ora

Lezione 3

Sur la Place du Ter



Vedi tutto

Corsi

Grammatica - Primi passi (A1)

Desrivere la tua città e dire perché ti piace

Desrivere persone e oggetti e dire cosa ti piace

## Esercizio 2

Realizzare una **una versione semplificata del gioco UNO in solitario**. L'obiettivo è scartare tutte le carte rispettando alcune semplici regole e verificare quanti turni sono stati necessari per farlo.

### Specifiche:

1. La partita inizia al caricamento della pagina. Il giocatore inizia con 3 carte.
2. Ogni carta è definita da un **numero** (da 1 a 9) e un **colore** (rosso, verde, blu, giallo).
3. All'inizio viene estratta casualmente una carta da porre sulla **pila degli scarti**.
4. Ad ogni turno:
  - o Il giocatore verifica se ha una carta compatibile (stesso numero o stesso colore).
  - o Se sì, scarta una di queste carte.
  - o Altrimenti, pesca una nuova carta dal mazzo (anch'essa generata casualmente).
  - o Il mazzo può essere infinito.
5. Il gioco termina quando il giocatore finisce le carte. Viene stampato il numero di turni eseguiti (quante carte sono state pescate).
6. NON c'è bisogno di rappresentare graficamente le carte. E' sufficiente mostrare: un numero da 1 a 9 in combinazione con una lettera (R, V, B, G) per il rispettivo colore

### L'interfaccia deve avere:

1. **Pila degli scarti:**
    - o Mostrare l'ultima carta messa sulla pila degli scarti.
  2. **Sezione del giocatore:**
    - o Mostrare le carte in mano del giocatore.
  3. **Sezione turno e fine partita**
    - o Mostrare un messaggio con il numero di turni
  4. **Controlli:**
    - o Un bottone:
      - **"Mazzo"**: Cliccando sul Mazzo il giocatore pesca una carta e la aggiunge alle sue.
      - o **Sezione del giocatore e Carte**: Cliccando su una delle sue carte in mano, se compatibile con quella in cima alla pila degli scarti, il giocatore gioca la sua carta.
- NOTA: ricordarsi di aggiornare la carta in cima alla pila degli scarti.

### Requisiti:

- È ammesso l'uso di attributi id e class.
- È consentito utilizzare librerie JavaScript come **jQuery**.

La formattazione è libera e non influisce sulla valutazione.

### Esercizio 3

Progettare un API REST (parziale) per la gestione di una chat. La chat è organizzata in *stanze*, ognuna individuata da un *ID* e una *data di apertura*; ogni *stanza* include un insieme di *messaggi*, ognuno caratterizzato da un *ID*, una *data di pubblicazione* ed *autore* e *testo*, entrambi di tipo stringa.

Scrivere un file in formato JSON o YAML.

L'API permette di:

- cancellare tutti i messaggi scritti da uno stesso autore in una data stanza
- aggiungere un nuovo messaggio ad una stanza
- modificare la data di pubblicazione di un messaggio

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