

# Esame di Tecnologie Web - parte II

## CdL in Informatica per il Management

**Tempo a disposizione: 2h**

### INFORMAZIONI PRELIMINARI

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

VIRTUALE: <https://virtuale.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:

- Librerie jQuery, Bootstrap: <http://diiorio.nws.cs.unibo.it/twe/lib/>

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.

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

- Nella versione desktop, barra di navigazione e layout occupano l'intero viewport in larghezza (predisporre anche una versione per dispositivi mobili, dettagli sotto).
- Il layout è composto da:
  - Una **barra di navigazione** con le label rappresentate nella schermata di riferimento. **NON** è necessario riprodurre l'icona della lente per la ricerca.
  - Una **sezione dedicata al prodotto** a sua volta composta da un breadcrumb (in alto a sinistra con le pagine/sezioni visitate per arrivare a quella corrente), una grande immagine del prodotto centrale e una colonna verticale a sinistra con immagini di anteprima. Sulla destra una colonna contiene i dettagli del prodotto: nome del prodotto e prezzo, dropdown con opzioni per il colore, vari box per le taglie disponibili e pulsante "Add to Bag". Quando viene selezionata una taglia il box selezionato viene segnalato con un bordo nero inferiore di 3px. **NON** è necessario riprodurre gli altri elementi oltre a quelli elencati qui sopra.
  - **Cliccando sull'immagine di anteprima l'immagine centrale cambia di conseguenza.** Infine, al passaggio del mouse il pulsante "Add to Bag" cambia colore di sfondo in un grigio chiaro.
  - Cliccando su testo o descrizione di un elemento cliccabile si apre la pagina corrispondente; creare un link arbitrario; **NON** è richiesto creare anche le pagine rispettive.
  - Colori e dimensioni esatte di margini e padding non rilevanti, purché appropriate.

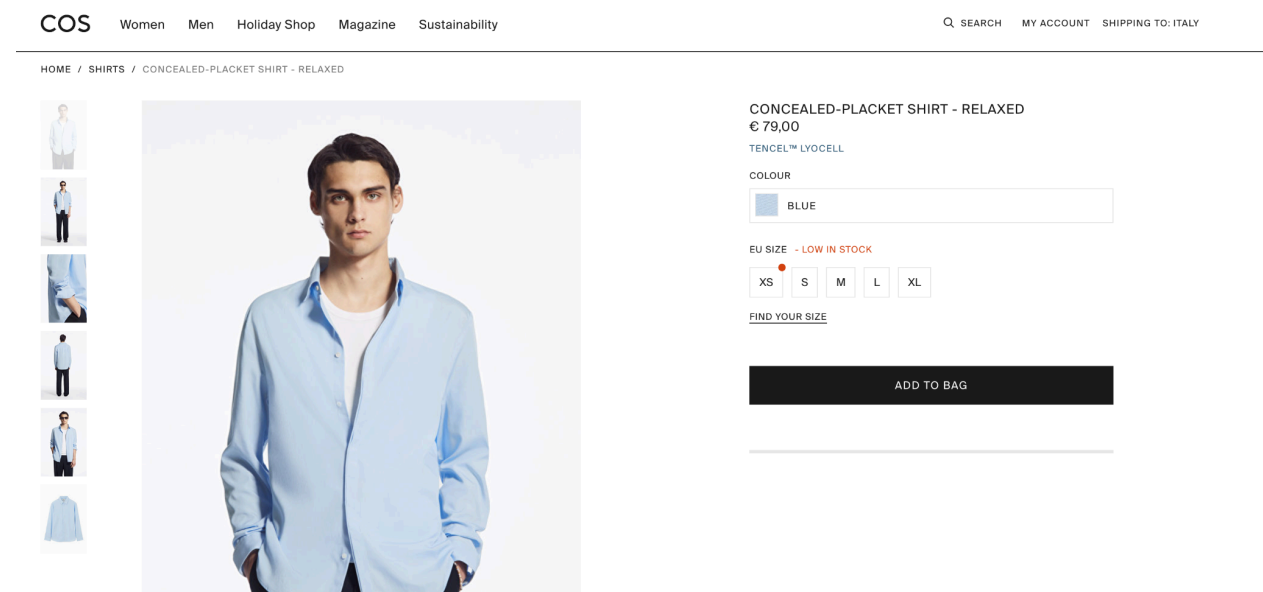
#### Vincoli:

- è ammesso usare attributi @id e @class nel sorgente HTML:
- non è ammesso usare tabelle per organizzare il layout

**Risorse:** Immagini disponibili su EOL

**Per gli studenti DSA: non è richiesto includere la barra di navigazione in alto e il breadcrumb**

Figura 1 - Resa della pagina in un browser



## Esercizio 2

Scrivere il codice HTML, CSS e Javascript per realizzare un timer e cronometro, mostrati nella stessa interfaccia.

L'applicazione permette agli utenti di avviare, fermare e resettare il cronometro, e impostare il timer con un tempo specifico.

### Note:

- L'interfaccia deve avere:
  - Un display per mostrare il tempo del cronometro e del timer.
  - Bottoni per avviare, fermare e resettare il cronometro, e un bottone per impostare il timer.
  - Un campo di input per impostare il tempo del timer (in secondi).
- Il cronometro deve incrementare il tempo ogni secondo quando è avviato.
- Il timer deve decrementare il tempo ogni secondo quando avviato, e fermarsi quando raggiunge zero, mostrando un messaggio di "Tempo scaduto!".
- Implementare la logica per avviare, fermare e resettare il cronometro e il timer, aggiornando il display di conseguenza.
- I dettagli di presentazione (posizione elementi, margini, padding, etc.) non sono rilevanti.

### Vincoli:

- È ammesso l'uso di attributi `@id` e `@class`.
- È ammesso l'uso di jQuery.

**Per gli studenti DSA: è richiesto implementare il timer o il cronometro (con tutte le funzionalità richieste).**

### Esercizio 3

Progettare un API REST (parziale) per gestire un insieme di laboratori informatici.

Ogni laboratorio è identificato da un nome (stringa) ed ospita diversi computer. Ogni computer è identificato da un codice (per semplicità stringa). Per ogni computer si memorizza il processore (stringa per semplicità) e la RAM in Gb (intero). Inoltre, su ogni computer è installato un sistema operativo tra *Windows*, *Linux* o *MacOS*. Per semplicità si può installare un solo sistema operativo su un computer.

Scrivere un file in formato JSON o YAML.

L'API permette di:

1. cancellare tutti i computer di un dato laboratorio X su cui è installato il sistema operativo Y
2. aggiungere un laboratorio, inizialmente vuoto
3. aggiornare la RAM di un dato computer

Scrivere un file in formato JSON o YAML.

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

**Per gli studenti DSA: non è richiesto svolgere il punto 3**