

JS Basics

Part 2

Lecture 6

Esercizio 1.0 - Cambiare testo e stile con evento

In questo esercizio, scrivere il codice JS necessario per modificare alcuni elementi del DOM.

La pagina contiene tre paragrafi, che condividono la stessa classe, e un bottone. Al clic sul bottone, il primo paragrafo deve **cambiare testo**, diventare **rosso** e acquisire la classe **.bold** (definita nel CSS), mentre il terzo paragrafo deve diventare **blu** e anch'esso acquisire la classe **.bold**.

Dovrete ottenere, dopo il click, una schermata come quella sotto (ad esclusione del bordo del bottone).

Ricordatevi che è necessario attendere che il DOM sia completamente caricato prima di selezionare e modificare gli elementi.

Esercizio 1.0 - Cambiare testo e stile

Il testo è stato cambiato!

Questo testo dovrà rimanere così.

Questo testo dovrà cambiare solo nello stile

Cambia testo

Esercizio 2.0 - Genera Mostri

In questo esercizio, scrivere il codice JS necessario per generare e gestire dei piccoli “mostri” all’interno della pagina. Il file HTML fornisce un bottone con ID **genera** e un contenitore vuoto con ID **areaMostri**. Ogni volta che il bottone viene premuto, deve essere creato un nuovo mostriattolo tramite JavaScript.

Un mostriattolo deve essere rappresentato da un nuovo elemento **<div>** con classe **mostro**. All’interno del contenitore vanno inseriti dinamicamente:

- un **paragrafo** contenente un nome casuale, scelto da un elenco predefinito di nomi;
- un bottone “**Nutri**”, che quando viene cliccato applica al mostriattolo la classe CSS **felice**, modificandone aspetto e stile;
- un bottone “**Libera**”, che quando viene cliccato rimuove quel mostriattolo dalla pagina.

In questo esercizio, scrivere il codice JS che:

1. attende il caricamento completo del DOM;
2. definisce una funzione che genera un nome casuale scegliendo un elemento da un array **nomi**;
3. crea un nuovo **<div>** per ogni mostriattolo, assegnandogli la classe CSS prevista;
4. genera il nome, crea gli elementi interni e li inserisce nel DOM tramite **appendChild()**;
5. aggiunge i due bottoni con i relativi **event listener** (“Nutri” e “Libera”);
6. inserisce il mostriattolo completo dentro **#areaMostri**.

Adotta un mostro

Premi il pulsante qui sotto per generare un nuovo mostro. Ogni creatura avrà un nome e due pulsanti: uno per nutrirlo e uno per liberarlo.

Genera mostro

I tuoi mostri:

Zippy

Nutri Libera

Blobbo

Nutri Libera

Gloop

Nutri Libera

Munch

Nutri Libera

Narf

Nutri Libera

Gloop

Nutri Libera

Blobbo

Nutri Libera

Poff

Nutri Libera

Esercizio 1 - Lista di lettura

Scrivere il codice HTML, CSS e JavaScript per creare una piccola applicazione che ci permette di aggiungere libri ad una lista di lettura, mostrata nelle figure presentate:

Figura 1 (stato iniziale):

The screenshot shows a web page with a header "Libri" and a list of five books: "Crime and Punishment", "Pinocchio", "Gli Argonauti", "Tecnologie Web", and "La Rosa". Each book has a green "Aggiungi alla Lista di Lettura" button below it. Below the header is a section titled "Lista di Lettura" which is currently empty.

Figura 2 (dopo aver aggiunto libri alla lista):

The screenshot shows the same web page after some books have been added. The "Lista di Lettura" section now contains the following items:

- Crime and Punishment: 3
- Gli Argonauti: 8
- La Rosa: 1
- Pinocchio: 1
- Tecnologie Web: 10

Each item in the list has a green "Aggiungi alla Lista di Lettura" button next to it.

Note:

- A. La pagina contiene diverse aree corrispondenti ai libri che è possibile aggiungere alla lista; la lista dei libri è predefinita e include 5 libri
- B. All'inizio, prima che venga cliccato alcun pulsante, la lista di lettura è vuota (Fig. 1)
- C. Ogni libro ha un pulsante corrispondente "Aggiungi alla lista di lettura"
- D. Al clic sul pulsante, il libro viene aggiunto alla lista di lettura (in fondo alla pagina), che viene quindi aggiornata e visualizzata.
- E. La lista di lettura mostra anche il numero di libri che sono stati aggiunti (se l'utente clicca tre volte sul pulsante per aggiungere il libro Crime and Punishment, la lista di lettura mostrerà "Crime and Punishment: 3").

- F. L'applicazione è interamente lato client e non salviamo alcun dato, quindi se l'utente ricarica la pagina, la lista di lettura verrà svuotata.
- G. Se si clicca 10 volte su un libro, il bottone "Aggiungi" per quel libro viene disabilitato
- H. La lista di lettura deve anche essere ordinata alfabeticamente. Ogni volta che viene aggiunto un libro alla lista, l'ordine alfabetico viene verificato.

Vincoli:

- è ammesso l'uso di jQuery

Figura 1 (stato iniziale):

Libri

Crime and Punishment

[Aggiungi alla Lista di Lettura](#)

Pinocchio

[Aggiungi alla Lista di Lettura](#)

Gli Argonauti

[Aggiungi alla Lista di Lettura](#)

Tecnologie Web

[Aggiungi alla Lista di Lettura](#)

La Rosa

[Aggiungi alla Lista di Lettura](#)

Lista di Lettura

Figura 2 (dopo aver aggiunto libri alla lista):

Libri

Crime and Punishment

[Aggiungi alla Lista di Lettura](#)

Pinocchio

[Aggiungi alla Lista di Lettura](#)

Gli Argonauti

[Aggiungi alla Lista di Lettura](#)

Tecnologie Web

[Aggiungi alla Lista di Lettura](#)

La Rosa

[Aggiungi alla Lista di Lettura](#)

Lista di Lettura

- Crime and Punishment: 3
- Gli Argonauti: 8
- La Rosa: 1
- Pinocchio: 1
- Tecnologie Web: 10

Esercizio 1bis - Lista di lettura

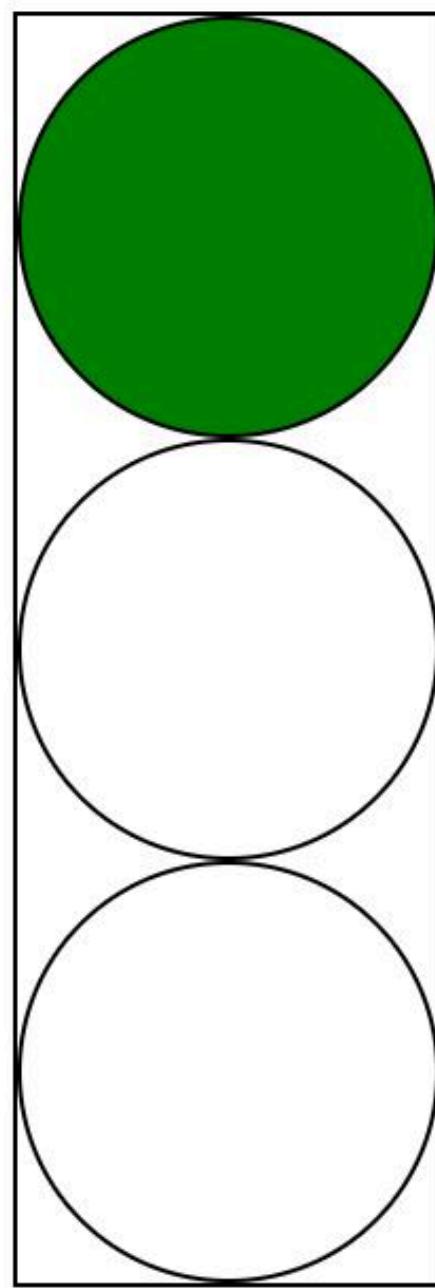
Come avrete notato, il funzionamento si basa sull'accoppiamento preciso di **#btnX e dei titoli dei libri nella pagina HTML e nell'Array del JS** (#btn0 - Crime and punishment; #btn1 - Pinocchio; ecc...).

Questo non è molto funzionale, un minimo cambiamento nei titoli o in altro risulterebbe abbastanza scomodo da fare seguendo questa logica, e potrebbe portare a vari errori.

Per casa, cercate di trovare una soluzione più flessibile.

Per qualunque dubbio o correzione, scrivetemi via mail.

Esercizio 2 - Semaforo



Scrivere il codice HTML e CSS per mostrare in un browser un semplice semaforo, con la visualizzazione mostrata di lato.

Aggiungere inoltre il codice Javascript per gestire il funzionamento del semaforo:

- Al caricamento della pagina il semaforo 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.

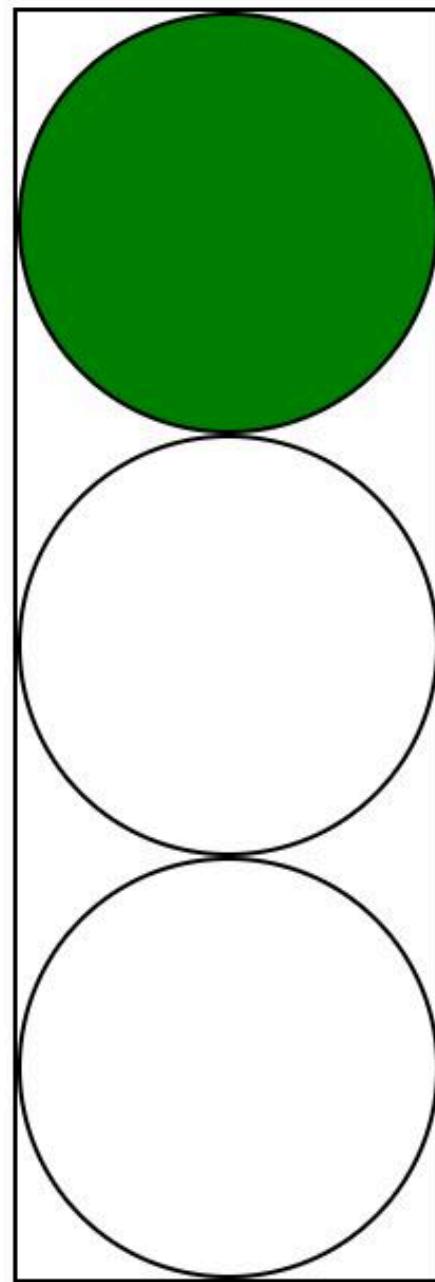
Vincolo aggiuntivo: Aggiungere l'**onClick** direttamente sul bottone. Successivamente proveremo a fare lo stesso esercizio ma con un **Listener** e l’aggiunta dinamica dell'**onClick**.

Vincoli:

- È ammesso l’uso di jQuery

Note:

- l’immagine mostra il semaforo verde a titolo esplicativo
- le misure esatte delle varie aree non sono rilevanti purché consistenti e proporzionate



Come abbiamo appena visto, HTML, CSS, e JS di questo esercizio sono molto semplici. Come da richiesta, provate ora a rimuovere l'`onClick` dal bottone e ad aggiungerlo direttamente in JavaScript tramite un `listener`.

Inoltre, guardando alla soluzione fornita, si possono notare alcuni elementi “migliorabili”. Provate a capire quali e a trovare soluzioni alternative rispetto a quelle appena viste.

Esercizio 3 (Per Casa) - Lista dei desideri

Scrivere il codice HTML, CSS e JavaScript per creare una piccola applicazione che ci permette di aggiungere dei prodotti ad una lista dei desideri, mostrata nella figura sotto (**specifiche nella prossima slide**):

Liste dei desideri

Prodotti

Prodotto 1 Aggiungi alla lista

Prodotto 2 Aggiungi alla lista

Prodotto 3 Aggiungi alla lista

La tua lista dei desideri :

Prodotto 1 : 7
Prodotto 2 : 4
Prodotto 3 : 7

Esercizio 3 - Lista dei desideri - Specs

Specifiche:

- L'elenco prodotto è predefinito e include 3 prodotti. Ogni prodotto ha 10 pezzi in magazzino.
- Ogni prodotto ha un bottone corrispondente: "Aggiungi alla lista"
- Al click su questo bottone, il prodotto verrà aggiunto alla lista dei desideri (nella parte inferiore della pagina), che verrà quindi aggiornata e visualizzata.
- Alla lista vengono aggiunti non solo i nomi dei prodotti, ma anche il numero di pezzi che sono stati aggiunti (click cinque volte sul pulsante per aggiungere Prodotto 1 -> nella lista dei desideri verrà visualizzato "Prodotto 1: 5").
- Se un prodotto viene cliccato 10 volte, il pulsante "Aggiungi alla lista" per quel prodotto viene disattivato.
- L'applicazione è completamente lato client e non memorizziamo alcun dato, quindi se l'utente aggiorna la pagina la wishlist verrà svuotata.
- La lista dei desideri deve inoltre essere ordinata in ordine alfabetico. Ogni volta che un prodotto viene aggiunto all'elenco, viene verificato l'ordine alfabetico
- I dettagli di presentazione (dimensioni, font, margini, ecc.) non sono rilevanti
- I nomi dei prodotti possono essere arbitrari
- I bottoni devono diventare più scuri nel momento in cui il mouse va in hover
- Il puntatore del mouse, quando i bottoni sono in hover, dovrà diventare di tipo pointer