TECNOLOGIE WEB

Esame del 13/09/2019 – Compito A  
Tempo a disposizione: 2 ore

**REGOLE.**

Tramite browser, potete accedere solo a:

- localhost

- <http://achecker.ca/checker/index.php> (per validare l’accessibilità del codice HTML)

- <https://validator.w3.org/> (per validare il documento HTML)

I link sopra sono accedibili facendo Ctrl-C Ctrl-V sul link ed aprendolo con Google Chrome.

Nella cartella del compito trovate una copia del W3Schools a cui potete accedere offline usando il browser.  
Potete usare Atom per scrivere il codice.

**IMPORTANTE**:

**lavorate all’interno della cartella della consegna, e consegnate TUTTI i file delle soluzioni (non solo il doc del compito).**

|  |
| --- |
| ESERCIZIO N. 1 **(7 punti)** |

Scrivere un documento HTML valido con codice HTML5 accessibile e semanticamente corretto per realizzare una form che invia i seguenti dati con metodo POST:

* Gruppo di campi riguardanti un personaggio di Star Wars, tra cui: Nome, Altezza (numerico), Peso (numerico) e corredati dal titolo “Campi Personaggio”.

Deve essere presente un bottone con scritto “Aggiungi” e la form è preceduta da un titolo: Personaggi Star Wars.

|  |
| --- |
| ESERCIZIO N. 2 **(6 punti)** |

**CSS**

Immagine che contiene screenshot

Descrizione generata automaticamente

Dato il file html dell’esercizio 1, realizzare il file .css (esterno) così da ottenere il layout e lo stile riportati nello screenshot qui sopra (browser: Chrome), tenendo in considerazione quanto segue:

* Tutti i font devono avere lo stesso font-family, che deve essere Arial. La dimensione deve essere del 100%.
* Il testo del titolo è grassetto, di colore green, allineato al centro, e il suo box ha un bordo dello stesso colore, spessore 2px, con un radius di 15px; il occupa una larghezza pari al 90% della pagina. Margini e padding devono essere simili a quelli riportati nello screenshot.
* I testi dei legend devono essere in colore green e allineati al centro.
* I testi delle label sono di colore green, allineati a sinistra.
* Gli elementi di tipo input hanno grandezza pari al 100% con bordo inferiore di 2px green. Margini e padding dei bottini devono essere simili a quelli riportati nello screenshot.
* Il bottone deve avere larghezza pari al 100%, colore del testo bianco e colore dello sfondo green. Margini e padding dei bottoni devono essere simili a quelli riportati nello screenshot.

L’uso delle media-query non è richiesto.

|  |
| --- |
| ESERCIZIO N. 3 **(5 punti)** |

Descrivere brevemente le principali differenze e analogie tra i solution stack LAMP e MEAN:

Scrivere qui la risposta:

|  |
| --- |
| ESERCIZIO N. 4 **(7 punti)** |

**PHP**

Scrivere il codice PHP valido (ovvero che esegua correttamente su server web Apache) che, dato l’esercizio precedente, venga inserito un nuovo personaggio di Star Wars e ritornato l’insieme dei personaggi salvati su database (in formato JSON), in base al tipo di richiesta.

In questa pagina occorrerà quindi:

Controllare anzitutto se la richiesta sia di tipo POST o GET.

* Nel caso la richiesta fosse di tipo POST:
  + controllare che le variabili inviate (name, mass, height) siano settate e non nulle
  + inserire il nuovo personaggio all’interno del database
* Nel caso la richiesta fosse di tipo GET:
  + controllare che nel database sia presente almeno un personaggio
  + selezionare tutti i personaggi e generare un file in formato JSON da ritornare al client simile al file star\_wars.json.
* **Dovete supporre che il db esista (nome database: settembre; nome tabella: starwars; username: root, pw: ) e che la tabella "starwars" sia strutturata e riempita secondo le istruzioni che trovate nel file "README\_DB.txt".**

|  |
| --- |
| ESERCIZIO N. 5 **(7 punti)** |

**JQuery**

* **N.B. E’ richiesto l’esercizio 1 e 4 per svolgere questo esercizio.**

Dato il file html dell’esercizio 1 'index.html', includere la libreria Jquery che trovate nella cartella Jquery e creare il file ‘soluzione.js’ JavaScript o jQuery in modo tale che:

* Al caricamento della pagina dovrà essere effettuata una richiesta GET alla pagina index.php dell’esercizio precedente, leggere i dati JSON e visualizzarli in una tabella (da aggiungere come ultimo elemento del body).
* Al click sul bottone “Aggiungi” si dovrà:
  + Controllare che il contenuto degli input non siano vuoti.
  + Se non vuoti, mandarli in post alla pagina index.php dell’esercizio precedente.
  + Eseguire nuovamente una richiesta GET per ottenere i dati aggiornati.
* **NB: NON SONO AMMESSE MODIFICHE AL FILE HTML**