TECNOLOGIE WEB

Esame del 31/01/2018 – Turno 1 B  
Tempo a disposizione: 2 ore

|  |  |
| --- | --- |
| ES 1 |  |
| ES 2 |  |
| ES 3 |  |
| ES 4 |  |
| ES 5 |  |

|  |
| --- |
| Cognome  Da compilare |
| Nome  Da compilare |
| Numero Matricola  Da compilare |

**REGOLE.**

Tramite browser, potete accedere solo a:

<http://www.w3schools.com/> (manuali e tutorial)

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-click sul link ed aprendolo con IE.  
Potete usare Atom per scrivere il codice.

**IMPORTANTE**:

1. compilare i campi relativi alle vostre informazioni (nome, cognome, matricola)
2. lavorate all’interno della cartella della consegna, e consegnate TUTTI i file delle soluzioni (non solo il doc del compito o il file della soluzione).

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

**HTML**

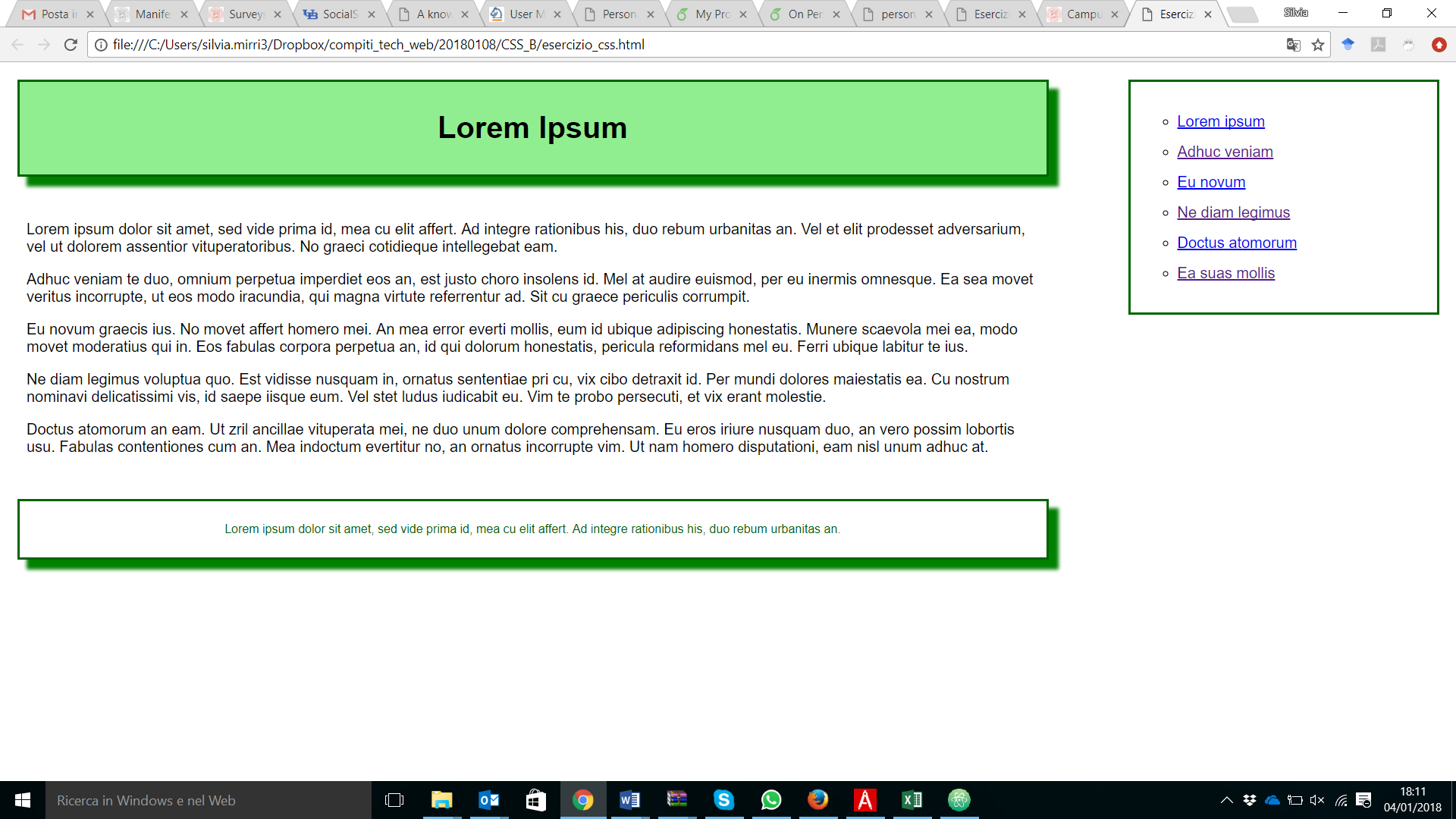
Scrivere il codice HTML5 valido, ben formato, accessibile e semanticamente corretto per realizzare un documento che contenga:

* una sezione dal titolo “ordine cover”. La sezione a sua volta contiene
* una form in cui inserire i seguenti input:
  + modello della cover a scelta (menù a tendina) tra iPhone, Samsung, Huawei.
  + La data di consegna
  + Submit e cancella della form

Il documento deve essere HTML5 accessibile secondo le WCAG2.0 a livello A (la validazione con tool automatici dell’accessibilità non è di per sé sufficiente).

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

**CSS**



Dato il file esercizio\_css.html (non modificabile), realizzare il file .css (esterno) così da ottenere il layout e lo stile riportati nello screenshot qui sopra, tenendo in considerazione quanto segue:

* L’elemento <nav> è posizionato a destra ed occupa circa il 20% della larghezza della finestra del browser. Il bordo è di colore darkgreen, spessore di 2px.
* Gli item della lista discendente di <nav> sono identificati da un cerchietto; l’interlinea è pari a 2 em.
* I link discendenti di <nav> hanno la caratteristica di sfondo colore darkgreen e testo in colore bianco in caso di mousehover.
* L’elemento <header> occupa circa il 70% della larghezza della finestra del browser; ha sfondo lightgreen, bordo darkgreen, di spessore 2 px. Il testo ha colore nero. Ha una ombreggiatura di colore green, con un effetto di sfocatura di 5px.
* L’elemento <section> occupa circa il 70% della larghezza della finestra del browser. Il testo è di colore nero, allineato a sinistra.
* L’elemento <footer> occupa circa il 70% della larghezza della finestra del browser, in modo simile a <header> e <section>. Il bordo deve essere di colore darkgreen, spessore 2px. Il testo deve essere centrato e di colore darkgreen. La dimensione del font deve essere inferiore a quella del testo riportato in <section>. Ha una ombreggiatura di colore green, con un effetto di sfocatura di 5px (come l’elemento <header>).
* Le dimensioni del testo e di larghezza dei blocchi sono espresse con dimensioni relative e mai assolute.
* Il layout deve essere liquido (responsive design).

Il posizionamento degli elementi di blocco può essere basato su una soluzione float oppure flexbox.

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

**Domanda**

Descrivere brevemente le metodologie di codifica dei colori nelle pagine Web e nei fogli di stile CSS:

Scrivere qui la risposta:

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

**JQuery**

Dato il file html **'esercizio\_javascript.html'** in allegato, creare il codice JavaScript o jQuery in modo tale che:

* Al caricamento della pagina devono essere nascoste gli span con classe “vinto” o “perso” e la parola codificata deve essere inserita nello span con id “parola”.
* Al click sul buttone, si dovrà leggere il contenuto del campo di input. Potete assumere che l’input sia sempre valido, ovvero che sia un singolo carattere.
  + Nel caso in cui la lettera sia presente nella parola, dovrete renderla visibile all’interno della parola codificata e aggiornare la parola codificata all’interno dello span con id “parola”. Infine, bisogna controllare se l’utente ha vinto, ovvero se sono mostrate tutte le lettere presenti nella parola. In caso l’utente abbia vinto, va mostrato lo span con classe “vinto”.
  + Nel caso in cui la lettera non sia presente, va aggiornata l’immagine (incrementando il suo valore). Nel caso siano finite le immagini visualizzabili, va segnalato all’utente che ha perso, mostrando lo span con classe “perso”.
* **NB:** Potete usare la funzione replaceAt che vi abbiamo messo a disposizione.

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

**PHP**

Scrivere il codice PHP valido (ovvero che esegua correttamente su server web Apache) per:

* Controllare che la variabile “numero” sia stata inviata tramite il metodo Get.
* Interrogare il database richiedendo tutte le righe della tabella “set” salvandosi la stringa “numeri”.
* Dovete supporre che il db esista (nome database: set; nome tabella: set; username: “root”, pw: “”) e che la tabella "set" sia strutturata e riempita secondo le istruzioni che trovate nel file "README\_DB.txt".
* Convertire la stringa “numeri” in un array.
* Per ogni valore nell’array controllare se questo corrisponde alla variabile “numero” passato tramite metodo Get, nel caso:
  + Eliminare dall’array il numero che troviamo nella variabile “numero”;
  + Convertire l’array in stringa (ogni numero dovrà essere intervallato dal simbolo “-”, come quello già presente nel database);
  + Inserire nel database la stringa risultante.