Lezione 16 gennaio 2024

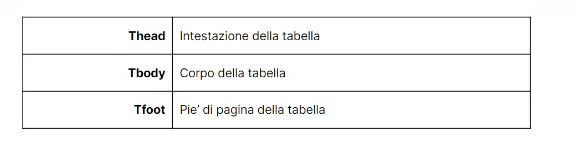
Tabella ed elementi multimediali

Le tabelle non scalano benissimo in tutte le soluzioni.

Tamplate dell’html:  
-html:5  
-!

Se scriviamo questo all’inizio ci apparirà il tamplate intero della pagina html.

**Tabella**, composta da:  
- Tag <Table> delimita l’area di una tabella  
- Per ogni riga servirà un elemento <tr> (table row)  
-Per delimitare l’area dell’intestazione usare <thead>  
- Intestazione (grassetto) è table heading <th>  
- Per ogni elemento th scritto avremo tot colonne  
-Per delimitare l’area delle colonne usare <tbody>  
-thead e tbody funzionano come head e body del tameplate  
- Per una nuova cella di dati usare <td> per riempire di dati le colonne  
-Colspan (css) serve ad aumentare il numero di colonne occupate per un particolare td  
-Rowspan (css) per mettere un solo elemento che occupa tot colonne (decise da me)

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

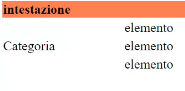
Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

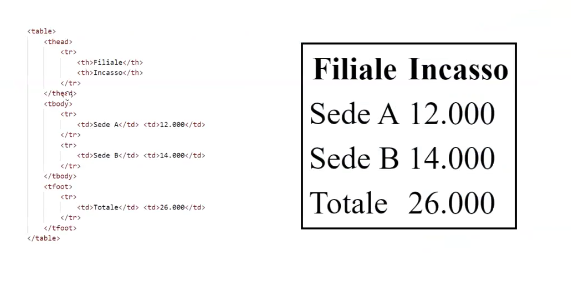
-Rowspan



Border ormai è deprecabile perciò non usarlo per fare i bordi alla tabella.

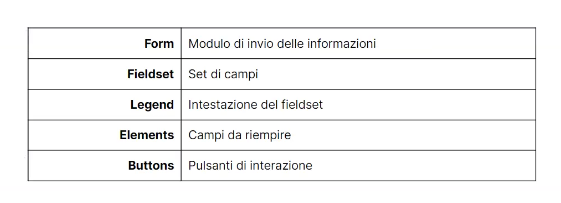


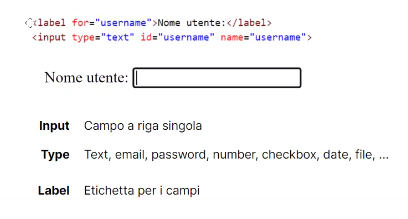
Sono obsoleti.



**FORM**

-I form servono per ricevere i dati dall’utente.





-Form è collegato a action per collegare un file php, ma ormai se si usa javascript action non ci serve più.

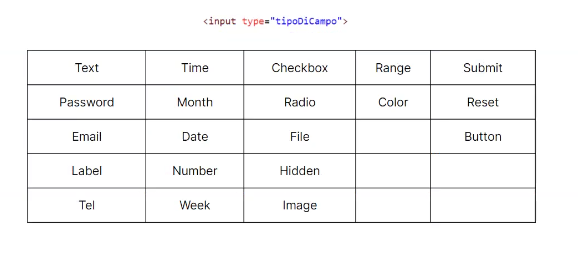


Per fare un form ci serve una <label> e un <imput>.



Input button è obsoleto, adesso c’è il tag <button> che è più moderno.

Label si usa sempre con un input, da scegliere tra i vari input esistenti.   
-Andare sul sito mdn web docs per avere la lista di tutti gli input possibili



Gli input hanno molti attributi, leggerli sempre nella lista del sito e non imparare a memoria. Sapere solo le possibilità che esistono per capire quale attributo usare.

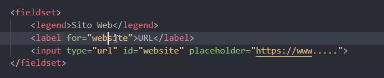
L’input deve essere legato alla sua label tramite id, quindi quello che si scrive in id verrà scritto anche per for per legarli.



Label è un elemento inline, mentre gli elementi input nascono come lineblock. Gli elementi lineblock possono avere una propria grandezza che può essere modificata, gli elementi inline invece no perché sono costretti all’interno della dimensione del testo.

-**Fieldset**

Raggruppa il label, input e **legend**.





Crea una sorta di spazio chiuso tra i vari input di inserimento, crea una sorta di struttura intorno.  


-Radio input

Nell’input del radio il nome è fondamentale. Più button radio devono avere il nome identico per legarsi tra di loro e permettere la selezione singola del radio (che si differenzia dal checkbox). Senza name uguale si comporta come checkbox.

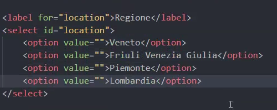
Per il **checkbox** il name non ci serve ma l’id si per collegarlo calla label. La checkbox funziona anche cliccando solo la sua label.

Quando un utente deve scrivere qualcosa di lungo, possiamo fornire l’input **textarea** (modificabile dall’utente stesso). E’ un tag che deve essere chiuso, può avere il placeholder. E’ possibile dare un numero di colonne massimo che possono essere aggiunte, oppure un massimo di caratteri da scrivere (vedere attributi del tag).



Cols e rows servono a ingrandire lo spazio visibile di scrittura senza che l’utente deve modificarlo da solo. Meglio gestire la larghezza che l’altezza (che può essere un valore qualsiasi).

**Select** è composta da un elemento select e le sue optioni interne.



Value serve a trasformare una parola molto lunga in un valore più piccolo da inviare al database che dovrà elaborarlo. Se non serve eliminare l’attributo value, il valore di ritorno sarà quello compreso in option.

Usare **optgroup** che serve a suddividere a gruppi le opzioni inserite in un select.

Il bottone <button> che invia il form non funziona da solo, il bottone submit invia i dati.  
Il bottone inerte serve per altri scopi, come per esempio il reset dei campi appena scritti.





Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

Con un attributo **required**, facciamo si che i campi devono essere per forza compilati correttamente per poter inviare il form.

**-Elementi multimediali incorporati ed esterni**

Il tag audio serve a integrare un suono in un documento e può contenere più di una fonte audio.   
l’attributo controls ci fornisce i controlli dell’audio.



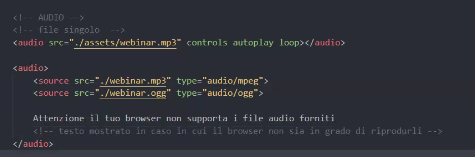
Il tag audio senza controlli è invisibile. Il tag audio ha bisogno di una fonte o più fonti. Con più fonti insieme il browser sceglierà quale suonare.

Immagine che contiene testo, schermata, Carattere

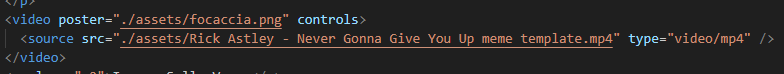
Descrizione generata automaticamente

**-Attributi degli elementi multimediali**

-autoplay: suona automaticamente appena può (cercare di evitare per la user experience)  
-controls: attiva i controlli dell’audio  
-loop: va in lopp  
-muted: può essere mutato



Esiste anche il tag **video** per i video e si comporta come il tag audio.



Tra gli attributi si può mettere anche un **poster**, ovvero un’immagine che si vedrà quando il video si sta caricando.



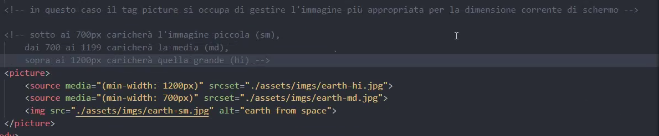
**Iframe** è un tag che serve per incorporare un sito esterno (quando si cerca di condividere un video e si incorpora). Usare sempre il codice fornito da incorpora in quanto il link diretto del video spesso non funziona.

Immagine che contiene testo, Viso umano, Software multimediale, software

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Sito Web, Pagina Web

Descrizione generata automaticamente

Il tag **picture** da un margine di default.



-**GIT**

Git è differente da Github.   
Con Git puoi salvare i tuoi dati come se fossero checkpoint. Man mano che andiamo avanti sceglieremo noi quando fare checkpoint, se qualcosa va storto possiamo ricaricare l’ultimo salvataggio.  
Checkpoint 🡪 **Commit**

Git è quindi il sistema che ci permette di tenere traccia del nostro progetto.

**DIV** 🡪 Raccoglitore di elementi, è inblock. Può essere usato molte volte quante volte si vuole. Serve per raggruppare gli elementi in uno e gli conferisce un bordo preciso.

Immagine che contiene testo, schermata, Carattere, biglietto da visita

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Header, rispetto al div, è più specifico.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Spesso il footer si mette a fine pagina ed è un piccolo riassuntino della pagina con informazioni importanti riguardanti la pagina.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, ricevuta, Carattere

Descrizione generata automaticamente

Si può mettere il **Main** dall’header al footer, deve essercene uno per pagina. **Nav** posso essere diversi. **Section** serve per delimitare sezioni particolari del testo, semanticamente legate (es. sezione correlati). **Aside** di solito si trova alla destra in verticale della pagina.

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente