**Lezione 11/04/2024**

**web developer**

**CSS (Cascading Style Sheets)**

**Scopo:**

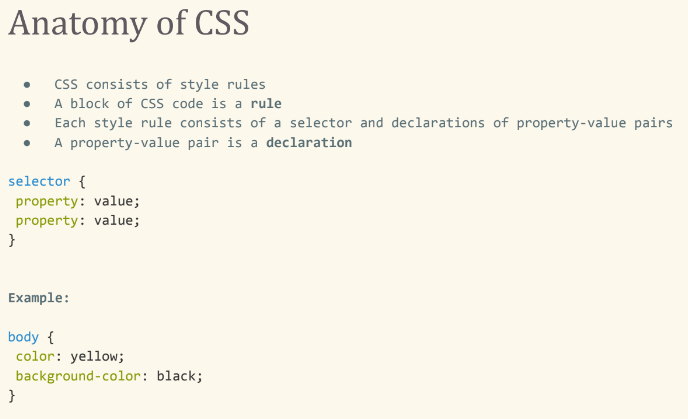
presentazione, grafica, personalizzazione, disposizione elementi(layout) della pagina html (img, adattamento a più device: responsive)

inoltre tra le funzioni che offre abbiamo le animazioni e le transizioni

nell’html avevamo elementi inline e block, main article header … e la cosa che potevamo modificare “visivamente” è l’ordine nella progettazione di un layout strutturato

in **html** il linguaggio è **nesting** (annidato, a tag)

in **css** il linguaggio è **cascading** (un elemento genitore contiene proprietà figlie)



ci sono il **selettore** che rappresenta elemento html alle quali vengono affidate delle **dichiarazioni** composte da **proprietà** il suo valore (attributo invece per html occhio)

**Metodi di applicazione css**

**Inline**



**Embedded**

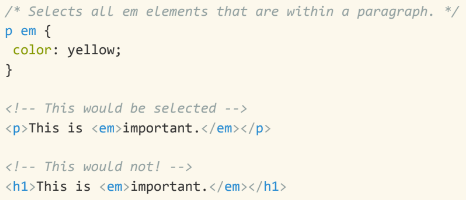
****questi non convengono per questione di funzionalità(html è strutturale) e prestazione(accumulo cache)

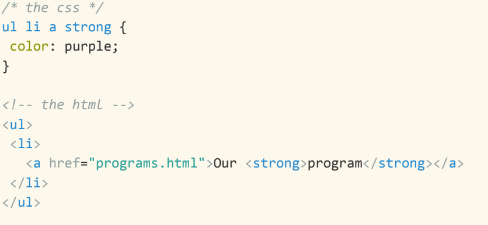
il metodo migliore è **l’esterno**  e richiamarlo nel html

****

**nb i selettori** che sono collegati a un tag html **coinvolgerà in modo attivo tutti gli elementi html con quel tag**

Per evitarlo servono selettori più intelligenti come:

This is *important*

our **program**

(più elementi inserisco per specificare, più il selettore sarà più preciso)

Ancora più specifici sono i selettori

**classe** (è attribuibile a più elementi)**: .nomeClasse**

**id** (è attribuibile solo ad un elemento o nessuno)**: #nomeId** (nb: label-input = for-id)

**Proprietà**

**color**(del testo) / **background-color:** applica colore

*nome(red), hexadecimal (#FF0000 /#FF0000FF), rgb(255, 0, 0)/rgba(255, 0, 0, 1)  
 hsl, hsla( questi due si intendono i valori hue, saturation, light e poi alpha)*

**width/height:** applica dimensioni di larghezza e altezza ad un elemento di tipologia block (non inline perché è variabile)(check)

*px: pixel unità minima che lo schermo riesce a visualizzare (20x)*

*em: dimensione carattere dell’elemento n em = tot pixel genitore (cioè n X totpx)*

*%: usato per dipendere da un elemento genitore*

*vw:viewport, misura di una porzione di spazio che se varia rimane proporzionato*

*basato allo spazio del padre(?)*

unità di misura

<https://css-tricks.com/the-lengths-of-css/>

<https://www.w3schools.com/cssref/css_units.php>

**font-family:** definisce il carattere di un elemento, devono essere disponibili nel dispositivo dell’utente, altrimenti vengono elencati font in serie uno dopo l'altro per prevenire problemi

casi font google o personali = performance lenta nel sito

Quelli esistenti li prende dal sistema operativo gli altri o si scaricano o da google font o simili

nb font-stack

serif e sans-serif sono gruppi di font non dei tipo

se non c’è un font allora verrà visualizzato quello del user agent style (font default del browser)

**font-size:** dimensione del carattere, se ci sono valori e si usa em, ci saranno dei condizionamenti da parte delle dimensioni di quegli altri elementi

**font-weight:** lvl di grassetto

e altro ancora

NB



interrompe il principio del cascading (se usato troppo o a prescindere ) di elementi con priorità inferiore

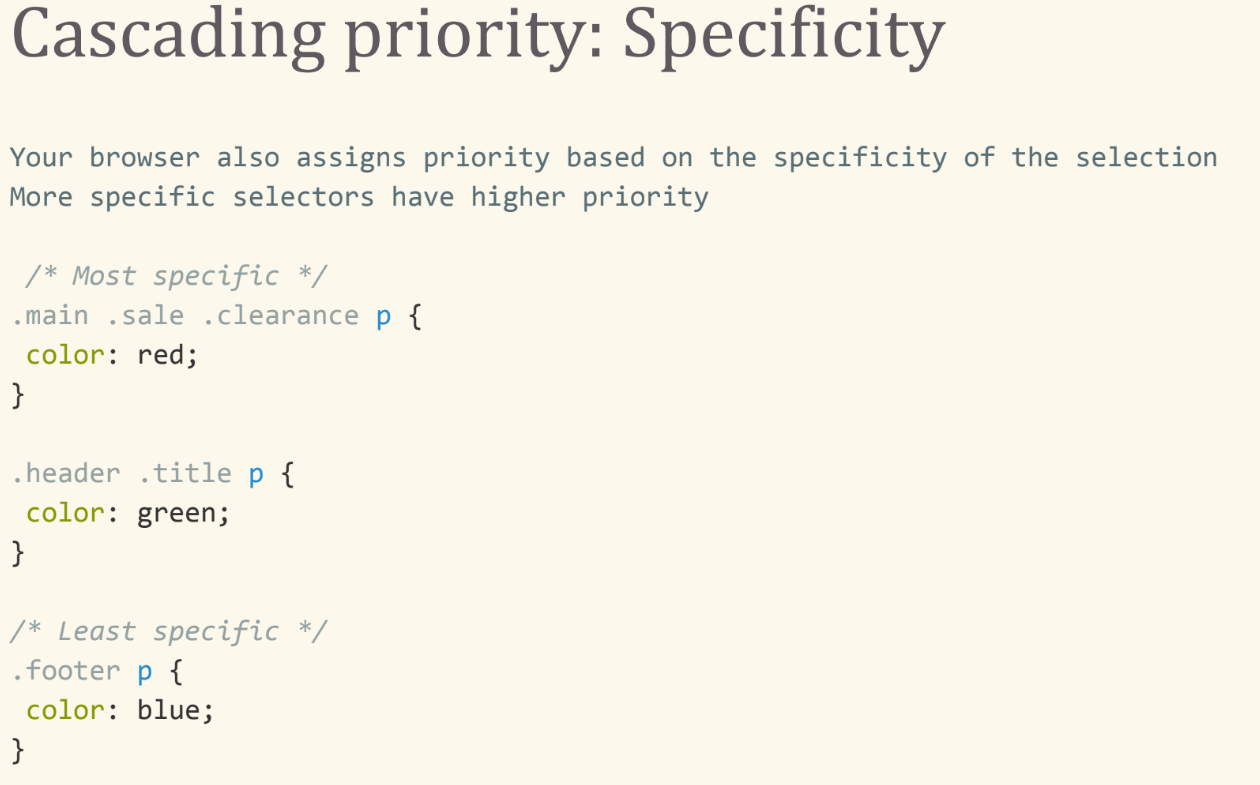
la regola nel css avrà Priorità assoluta sovrastando altre dichiarazioni, impatta sulla singola dichiarazione

nb



nb specificità: più specifico più considerato

!important va sopra inline al primo posto



!important: rompe le regole da usare in caso di sovrascrittura, USO SCONSIGLIATOOOOO







