Notes IU

> *you’re more intelligent than what you think you are*

https://pomofocus.io/

Web development and communication strategy

Teoria

I linguaggi

- HTML (Hypertext Transfer Protocol)- il linguaggio standard di *markup* per realizzare pagine web: è un linguaggio di marcatura= modo in cui comunichiamo con il mezzo (come le cose devono apparire). I linguaggi di markup non sono linguaggi di programmazione come (Javascript), permettono di annotare un testo. Consente ad una macchina (il browser, e.g. mozilla firefox, Google Chrome, Opera, Safari) di visualizzare i contenuti di una pagina web in strutture logiche (titoli, sezioni, paragrafi, immagini).
- CSS - linguaggio per la grafica web: CSS non ci fa dire quali sono i blocchi logici ma quello che permette di manipolare la grafica. I linguaggi per la grafica web

permettono di
applicare stili agli elementi delle pagine web.

- Javascript, Python, Java, PHP, etc. - linguaggio di programmazione utilizzato per gestire l'interazione utente sul web, aggiunge interattività. I linguaggi di programmazione forniscono istruzioni alle macchine per compiere azioni (non solo web) e.g., calcoli, interattività dei contenuti.

Tool
-Sublime Text 2 editor di testo (come word) gratis. È un rich text editori che permette di scrivere codice.
-Github è un repository che permette di caricare contenuti html (un sito) in modo gratuito.

HTML

Inventato nel 1990 da Sir Tim Berners-Lee, HTML è la *lingua franca* per la creazione di pagine web.
Consiste di una serie di tag per annotare del testo, normalmente redatto usando un text editor, e dare indicazioni al browser su come visualizzare la struttura logica della pagina (e.g. intestazione, sezioni, paragrafi), le caratteristiche del testo (e.g. titoli, italico, grassetto, lista puntata) e i multimedia (e.g. immagini, video). Attualmente, il linguaggio è alla versione 5
(HTML5).

-html mostra i link, i titoli con grandezze diverse, grassetto perché i browser hanno una minima grafica associata agli elementi di html quindi il browser sa vagamente come applicare queste info grafiche.

-'*lingua franca*' vuol dire che tutti i siti del mondo usano questo linguaggio per mettere online contenuti

-dovremo spiegare al nostro browser che una certa stringa è un titolo, che lì ci vuole un'immagine ecc. Tutto questo lo facciamo annotando il testo. Non essendo un linguaggio proprietario, non ha bisogno di tool proprietari e significa che usando un qualsiasi editor posso scrivere in html. Un editor funzionante funziona ovunque.

-le annotazioni in html ci consente di scrivere alcune cose: si occupa della struttura logica della pagina (paragrafi, sezioni, titoli ecc) ma non della presentazione grafica; specifica alcune caratteristiche grafiche e permette di inserire place holders per oggetti multimediali.

-un sito web è composta da uno o più file di testo (uno per ogni pagina web) + uno o più pagine di grafica.

Visibilità di un sito nel web
Un sito ben organizzato (navigazione, linking), con le giuste parole chiave (labelling), con contenuti studiati ad hoc e una struttura della pagina conforme agli standard (HTML) è sufficiente per essere posizionati in alto nei motori di ricerca.

Head e body sono la base di tutto:

- head= non lo vediamo nella pagina
- body= tutto quello che vediamo nella pagina (ne sono parte menu, impostazioni ecc..)

La codifica del testo per il web
La codifica del testo consente di creare un linguaggio condiviso tra l'uomo e la macchina affinché quest'ultima possa interpretare alcune semplici *istruzioni* sulla struttura logica e l'aspetto dei contenuti di un testo.

HTML permette di codificare/annotare/marcare elementi della struttura logica di un testo (divisione in blocchi, titoli, immagini) e aspetti tipografici (corsivo, grassetto) mediante (1) l'utilizzo di un vocabolario controllato di termini e (2) una sintassi prestabilita per l'ordine dei termini (una sorta di grammatica).

html è un **linguaggio di markup**= permette di annotare del testo, di non fare altro. Permette di dire che tale cosa è un'immagine, quella cosa è un titolo ecc. Si spiega alla macchina che significato ha quell'elemento. Html fa da intermediario tra noi e la macchina fornendo un vocabolario, ovvero una serie di parole chiave e una sintassi (regole per scrivere e spiegare alla macchina)

Una **pagina HTML** è un documento di testo - che può essere redatto con qualsiasi editor di testo - che utilizza il vocabolario e la sintassi HTML per marcare i contenuti ed è salvato nel formato .html .

HTML fornisce un vocabolario prestabilito per descrivere gli elementi principali di una pagina web.

Tag= qualsiasi cosa stia tra parentesi uncinate; sono annotazioni, puntatori che consentono di specificare dove inizia l'elemento logico e dove finisce; possiamo inserire elementi che permettono di manipolare alcuni aspetti grafici. Si mettono dentro a *parentesi uncinate*, di solito il tag di chiusura ha uno **slash**.

Elemento= keyword o nodo, formato da tag iniziale + *valore* + tag finale → es. *ciao*

Valore= ciò che sta dentro il tag di apertura e il tag di chiusura → perciò può essere:

-solo testo come nel caso di es autore: Andrea Bocelli

-altri elementi come nel caso di <html> </html> es <section> ... </section>

-testo + altri elementi es titolo: *Romanza*

- ** , ** = paragraph, sotto-divisione del testo (elemento di organizzazione logica della pagina dove si va a capo quando finisce lo spazio)

- **,** = italic, corsivo

Sono un paragrafo con un testo in *corsivo*. → ** ** Sono un paragrafo con un testo in corsivo.** **

- struttura base di una pagina HTML:

1. **<!doctype html>** = dichiarazione del linguaggio usato, sta spiegando al browser il tipo di documento, ovvero si aspetta di trovare un documento html
2. **<html>** = nodo radice, elemento radice, sia all'inizio che alla fine. L'elemento distintivo di una pagina web, racchiude tutte le informazioni della pagina

Le altre righe sono tutte **indentate** -> tra la keyword iniziale e finale abbiamo altre cose e la stessa cosa vale per body (allo stesso livello)

3. **<head>, </head>** = intestazione della pagina, sta allo stesso liv di body e tutto quello che sta dentro non è visibile nella pagina ma includono informazioni fondamentali per la macchina (browser)→ dentro a head trovo <title>. Head può contenere altre informazioni, come i link a risorse esterne, e.g. i fogli di stile .css, o

metadati (meta), ovvero descrizioni dei contenuti della pagina per l'indicizzazione dei motori di ricerca.

4. **<title>,</title>** = viene visualizzato nel tab del browser, non nella pagina! Title è un nodo discendente di head, che è nodo parente di body
5. **<body>,</body>**= corpo del testo della pagina, contiene tutto quello che può essere visualizzato in una pagina web. Sections, paragraphs possono stare solo qui, non in head.
6. **<section>, </section>** = creo sezioni, come dei blocchi dentro al quale posso mettere quello che voglio (di solito img o p).
7. nelle sezioni ho diversi tipi di titolo: **heading + n' (livello) → h1, h2, h3, h4**…→ posso averne quanti voglio tranne h1 che è solo uno ed è il titolo principale.

Le spaziature non sono obbligatorie ma le aggiungiamo per dare più leggibilità al codice html: vedo che al primo liv di indentazione ci sono head e body, aggiungo una spaziatura e ho un ulteriore liv di innestamento (come matrioske). Noi in html non abbiamo il potere di dire alla macchina di andare a capo se non attraverso apertura e chiusura della keyword.

NB: la struttura base è

-html

—head

……title ,/title

—/head

—body

……section

…….h1,

…….p, /p

…../section

—/body

struttura gerarchica dei contenuti: body > section > p > i → la gerarchia è evidenziata dall'utilizzo (consistente) di spazi all'inizio della linea, chiamato indentazione. La gerarchia è rappresentata mediante l'apertura e la chiusura di tag, e.g. e

rispetta delle *regole di semantica dei termini*, e.g. una sezione può contenere paragrafi (e.g. section > p) ma non viceversa (p > section). Gli elementi possono essere ripetuti, e.g. più paragrafi p in una sezione section.

Ogni documento HTML può essere visto come un *albero* (una struttura gerarchica) di nodi (detti elementi HTML). Esistono n*odi parenti e nodi discendent*i

Ogni documento HTML ha sempre un nodo radice, obbligatorio, ovvero l'elemento html che racchiude tutti gli altri elementi. N.B. In HTML esistono alcuni elementi obbligatori (html , head , title e body). Tutti gli altri dipendono dallo sviluppatore.

I rapporti tra nodi parenti e nodi discendenti si chiamano **annidamenti**. Un elemento HTML può contenere un altro elemento HTML SOLO SE lo include completamente.

titolo:*Romanza*

~~ titolo: *Romanza* ~~

Esistono elementi vuoti, detti anche **milestone**, che non contengono alcun valore. Gli elementi vuoti servono per inserire interruzioni, o multimedia (immagini, video). Gli elementi vuoti possono essere rappresentati con una sintassi sintetica dove lo slash fa da bookmark per quello che viene dopo. es elemento **
** = break che obbliga un paragrafo ad andare a capo prima che finisca lo spazio.

Ad ogni elemento HTML possono essere associati uno o più **attributi**. La funzione dell'attributo è specificare ulteriori caratteristiche dell'elemento a cui è associato. Il valore dell'attributo può essere unico all'interno della pagina o condiviso con altri elementi.

L'attributo compare solo nel tag di apertura, non viene ripetuto nel tag di chiusura.

Gli attributi si scrivono tutti con il **tag di apertura + nome attributo+ = +" “+ spazio**; poi nelle virgolette metto quello che voglio. Non ci possono essere *id* che hanno tra virgolette la stessa cosa; mentre ci possono essere quanti elementi vogliamo con stesso valore di *classe*.
Gli attributi sono ‘aggettivi’: alcuni elementi si possono comportare diversamente da altri → es vorrei dare uno sfondo di colore diverso a paragrafi alternati: per identificare uno specifico elemento (utile per la grafica in questo caso) posso usare:

- **attributo** detto **id**, identifier. Consente di attribuire un identificativo univoco (il valore dell'attributo) all'elemento a cui è associato. Il valore deve essere unico all'interno dello stesso sito. Potenzialmente tutti gli elementi della pagina possono avere un id:

lo diamo coscientemente solo agli elementi che vorremmo modificare nella grafica, agli elementi che sono eccezioni, che si devono comportare diversamente da altri in termini di grafica.

- **attributo class**, anche questo può essere dato a tutti; il valore si ripete in tutti gli elementi che voglio raggruppare per una qualche ragione. Identifica un numero di elementi, modo per raggruppare o per dividere

HTML è case insensitive, ovvero posso digitare gli attributi sia in maiuscolo che in minuscolo (per convenzione scrivo in minuscolo sia attributi che tag). Anche l'indentazione del codice è una buona pratica (per la leggibilità del codice) ma non è obbligatoria. Gli ‘a capo’ servono a me per leggere meglio il codice ma non vengono letti da html (uso br oppure lo fa in autonomia quando chiude il tag)

- <html **lang**=it> →language, specifica la lingua della pagina; il valore è una lista controllata di valori dove en è inglese e it è italiano, sono acronimi per ogni lingua (importante per i device dei non vedenti per leggere la pagina nella lingua corretta e per le traduzioni automatiche.
- **href** (Hypertext REFerence) *sempre* associato all'elemento **a** (anchor) permette di creare un link ad una pagina es. ****nome che comparirà nella pagina****

→se rimandiamo l’utente a una pagina esterna uso la URL completa (URL assoluta); se lo rimando ad un’altra pagina interna al mio sito uso la URL parziale, senza http://

N.B. Gli attributi @id e @class saranno i nostri punti di ancoraggio per richiamare in CSS gli elementi HTML ai quali vogliamo associare uno stile grafico. Pertanto la scelta (degli elementi a cui associare un attributo, quale attributo associare e quale valore attribuire)

Altra buona pratica è aggiungere commenti nel codice per annotare cosa si sta facendo o cosa si intende fare. Il contenuto dei commenti non viene visualizzato dal browser. I commenti sono inclusi tra **<!-- -->**

HTML non include aspetti di *semantica*, e.g. non è possibile specificare se si sta parlando di una persona, di un luogo o di un oggetto che è (potenzialmente) descritto in altri documenti web. Per questi aspetti esistono altre tecnologie, dette Semantic Web, per arricchire una pagina web con descrizioni di concetti (reali e astratti) e relazioni tra concetti (e.g. avere autore).

Struttura della pagina: html , head , title , body.

Dentro a body abbiamo:

→Intestazione e menu: **header , h1 , nav , ul , li , a** **NB: header (intestazione, in body, visibile) è diverso da head (elemento parente di body, non visibile)**
L'elemento **header** normalmente è il primo elemento incluso in body ed include tutti gli elementi che compongono l'intestazione della pagina web, e.g. titolo, logo, menu.
-Il titolo principale della pagina va sempre espresso con l'elemento **h1** , (heading di livello 1).
-L'elemento **nav** include il menu di navigazione primaria (o principale) del sito, che collega la pagina in cui ci si trova alle altre pagine che compongono il sito.
Normalmente l'elenco delle pagine che compongono il menu (o qualsiasi elenco all'interno di una pagina HTML) è incluso in un elemento **ul** (unordered list), che a sua volta include tanti elementi **li** (list item) quante sono le pagine del menu.
Ogni elemento della lista (ovvero ogni pagina a cui si vuole linkare dal menu) include un elemento **a** (anchor), il cui attributo @**href** include la URL parziale della pagina html a cui si vuole rimandare. → il menu deve essere uguale per ogni pagina del sito.

L'elemento **li** può contenere diverse cose (anche icona) ma soprattutto deve contenere l'elemento a che serve per crear il link interno (e non esterno come a youtube)
-se mando l'user fuori dal sito faccio come prima
-se lo rimando a pagine del mio sito basta specificare nome del file perché io devo avere tanti file html tante quante le mie pagine del sito es 3 file html per home, about e documentazione

I file chiamateli con nome significativo, non index.html perchè quella è la home page ed è obbligatoria. I nomi dei file si scrivono sempre tutto minuscolo oppure cd_list o cdlist.

Note: * l'utilizzo e l'ordine degli elementi h1 , nav , ul non sono prescrittivi, ma sono una buona pratica utilizzata universalmente per lo sviluppo web. Infatti i motori di ricerca seguono questo schema per trovare e indicizzare le informazioni del sito web (e.g. cercano il titolo in h1 per dedurre il tema della pagina -- quindi le parole utilizzate sono importanti!).

* la home page di ogni sito web DEVE sempre essere chiamata index.html . Questo perchè il browser legge la lista di pagine html nella cartella del sito e interpreta sempre index.html come la pagina di partenza.

* href="about.html" implica l'esistenza di una pagina html chiamata about.html contenuta all'interno della stessa cartella della home page index.html . Se la pagina about.html non esiste, il browser ritorna un errore (404 not found)

* nel menu vanno inseriti link a pagine interne al sito. Non vanno mai inseriti link che rimandano a risorse esterne al sito (e.g. link a social network) e che quindi buttano fuori l'utente dal sito.

→Contenuto della pagina: **h2-h6 , main , aside , section , p**
I titoli successivi al titolo della pagina vanno espressi con headings di livello inferiore h2 , h3 , h4 , h5 , h6 . I numeri crescenti non rappresentano l'ordine dei titoli all'interno della pagina, bensì il livello di sottotitolo. Normalmente questi elementi sono figli di altri elementi contenitori (vedi i successivi main , aside , section) e mai figli diretti di body .
Il contenuto principale della pagina (generalmente centrale o allineato a sinistra) può essere inserito nell'elemento **main** , che viene disposto allo stesso livello, successivamente all'elemento header .

Contenuti secondari (e.g. menu di navigazione secondaria o contestuale, link a post su social network, note a margine) possono essere inseriti nell'elemento **aside** . Il significato di main e aside è puramente semantico, poichè il comportamento all'interno della pagina (ovvero come vengono visualizzati) non cambia. Main e aside possono contenere blocchi di testo, chiamati section , che separano logicamente testi di natura diversa. Section può anche essere utilizzato in alternativa a main e aside; section può contenere titoli h1-h6 , paragrafi p e altri elementi.

footer= piede di pagina. L'elemento footer racchiude le informazioni a piede di pagina. Si trova come ultimo elemento figlio di body . Può contenere testo strutturato in sezioni o paragrafi,
o immagini (e.g. loghi di sponsor). Contiene informazioni e link sui contatti del sito (e.g. tecnico, commerciale, social network), i crediti del sito (e.g. grafica, contenuti), e l'attribuzione del sito (il proprietario del sito). Assieme a header , questi elementi sono generalmente fissi, ovvero si ritrovano identici in ogni pagina che compone il sito.
Per esempio:

```
<footer>
  <p>Sito realizzato da <a href=”http:/example.org”> me </a> </p>
</footer>
```

Multimedia: **img**
Immagini e altri multimedia (video, audio) possono essere contenuti a qualsiasi livello della pagina, sempre all'interno di elementi figli di body . L'elemento img consente di inserire un file immagine all'interno della pagina. L'attributo @**src** , similmente a @href include la URL parziale al file che si vuole inserire in quel punto, oppure i passaggi tra cartelle. L'elemento img è un elemento vuoto, che può essere rappresentato con la sintassi abbreviata.
Per esempio:

non c’è un tag di chiusura ma solo uno slash. @src è un link multimediale come @href (link ipertestuale).

Note:
"imgs/logo.jpg" implica che l'immagine logo.jpg è contenuta in una cartella chiamata imgs , la quale si trova nella stessa cartella (quindi allo stesso livello)
della pagina HTML in cui viene richiamata l'immagine (index.html).

L'attributo **alt** serve per mettere un'eventuale alternativa all'immagine nel caso non fosse vista o non può vederla, come una descrizione. Qui posso usare il testo libero, diversamente dagli altri attributi ma non posso usare le stesse virgolette (posso usare le singole se di solito uso le doppie); quello che scrivo in alt non viene visualizzato. Le dimensioni dell'immagine vengono messe da CSS.

Alcuni effetti tipografici possono essere rappresentati tramite appositi elementi html, e.g. corsivo **i** (o em), grassetto **b** (o strong), sottolineato **u** .

Altri effetti tipografici più sofisticati possono essere realizzati tramite l'utilizzo di CSS. Per esempio: i è una forzatura del font utilizzato rendendolo obliquo. In tipografia (web e tradizionale), il corsivo non corrisponde all'obliquo, ma ha un proprio disegno, con pesi e bilanciamento degli spazi e delle grazie studiato ad hoc. Tramite CSS è possibile specificare quando utilizzare il corsivo vero o l'obliquo.

Link utili
Il miglior modo di imparare HTML è fare pratica e vedere tanti esempi.
Ecco una lista di tutorial (mainly in english!) per prendere confidenza con HTML: *
Mozilla - Introduction to HTML spiegazione delle basi di HTML, esempi e active

learning * W3School HTML sito con tutorial, esempi ed editor per interagire/sperimentare ad ogni lezione. * learn HTML spiegazioni con esempi e test interattivi
Impareremo buona parte degli elementi del vocabolario HTML strada facendo. Per una lista completa degli elementi HTML vedi la Reference List di Mozilla, che include tutti gli elementi HTML raggruppati per organizzazione logica della pagina (head / body) e tipologia di contenuti (testo, multimedia

- [ ] Pubblicare versione nuova sito con br sistemato