



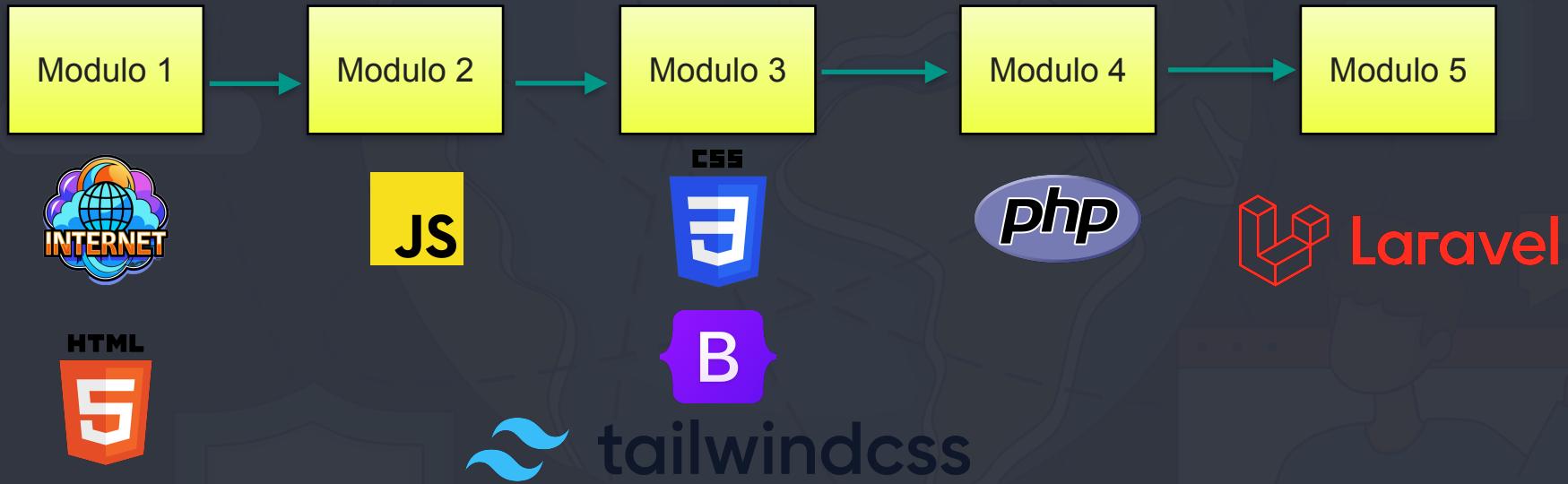
Sviluppatore Full Stack con Laravel, PHP, Javascript



Francesco Mansi
Web Specialist

Programma del Corso

(101 o Zero to Hero)





Modulo 1

Internet & HTML

Capitolo 1

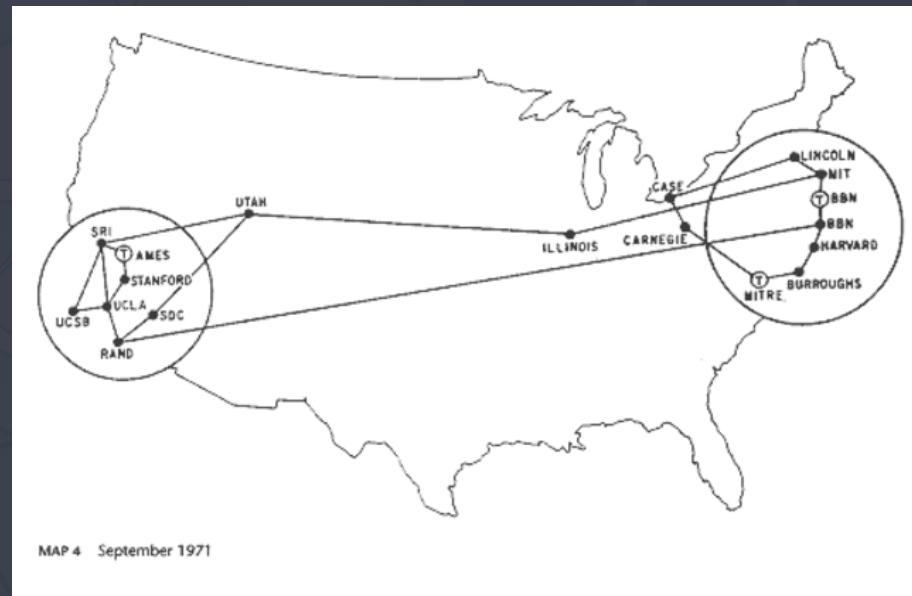
La Storia di Internet

1969 - A.R.P.A., un ente governativo americano per la difesa, sviluppa ARPANET (Una rete militare finalizzata allo scambio di informazioni, un sistema che doveva essere veloce e sicuro).

Utilizzando 4 poli universitari:

- UCLA (Università California Los Angeles)
- SRI (Istituto di Ricerca)
- UCSB (Università California Santa Barbara)
- UUSC (Università Utah ella School Computing)

1972 - Prima connessione nazionale in ARPANET.



La Storia di Internet

- Architettura Client/Server
- TCP/IP (Transmission Control Protocol/Internet protocol)
- Ampiezza di banda 50Kbs (<https://www.youtube.com/watch?v=OGwlwZs9PxE>)
- Era possibile mandare dei pacchetti testuali.
- Il primo messaggio su ARPANET fu inviato da uno studente di nome Charley Kline alle 10:30 del 29 ottobre 1969 con scritto: "Lo" Fonte: <https://www.icann.org/en/blogs/details/the-first-message-transmission-29-10-2019-en>

Internet Bandwidth Comparisons

Connection	Speed		Transmit 1 GB		1 MB	10 K
	type	bps	B/sec	Days hh : mm : ss	mm : ss	Seconds
28.8 Modem	28.8 K	3.5 K		3 : 10 : 51 : 01.62	4 : 51.27	2.8444
56K Modem	56 K	6.8 K		1 : 18 : 36 : 31.69	2 : 29.80	1.4629
DS0	64 K	8.0 K		1 : 12 : 24 : 32.00	2 : 08.00	1.2500
2 Channel ISDN	128 K	15.6 K		18 : 38 : 28.86	1 : 05.54	0.6400
Common DSL	512 K	62.5 K		4 : 39 : 37.22	: 16.38	0.1600
Cable	1 M	122.1 K		2 : 23 : 09.93	: 8.39	0.0819
T1	1.544 M	188.5 K		1 : 32 : 43.43	: 5.43	0.0531
T2	6.312 M	770.5 K		: 22 : 40.89	: 1.33	0.0100
Ethernet	10.000 M	1.220 M		: 14 : 18.99	: 0.84	0.0082
T3	44.736 M	5.461 M		: 3 : 12.01	: 0.19	0.0018
OC-1	51.840 M	6.328 M		: 2 : 45.70	: 0.16	0.0016
Fast Ethernet	100.00 M	12.207 M		: 1 : 25.90	: 0.08	0.0008
OC-3	155.52 M	18.984 M		: : 55.23	: 0.05	0.0005
OC-12	622.08 M	75.937 M		: : 13.81	: 0.01	0.0001

La Storia di Internet

1983: 213 macchine connesse ad Arpanet iniziano ad usare i protocolli TCP/IP, e l'Università del Wisconsin crea il **DNS (Domain Name System)** per facilitare la comunicazione a host conosciuti

Ma sempre nel 1983 ARPA esaurì il suo scopo: lo stato chiuse l'erogazione di fondi pubblici, la sezione militare si isolò, necessitando di segretezza assoluta a protezione delle proprie informazioni, e nacque perciò MILNET.

1990: Con il passare del tempo, l'esercito si disinteressò sempre più del progetto fino ad abbandonarlo e rimase sotto il pieno controllo delle università, diventando un utile strumento per scambiare le conoscenze scientifiche e per comunicare.

Arpanet, formata da 300 mila macchine, lascia il posto ad **Internet**. Cominciano a svilupparsi le **backbone** (dorsali ad alta velocità, vedi micelio funghi). Presso il CERN di Ginevra, è sviluppato il protocollo per lo scambio di immagini oltre che di dati (ipertesto), attraverso il linguaggio HTML.

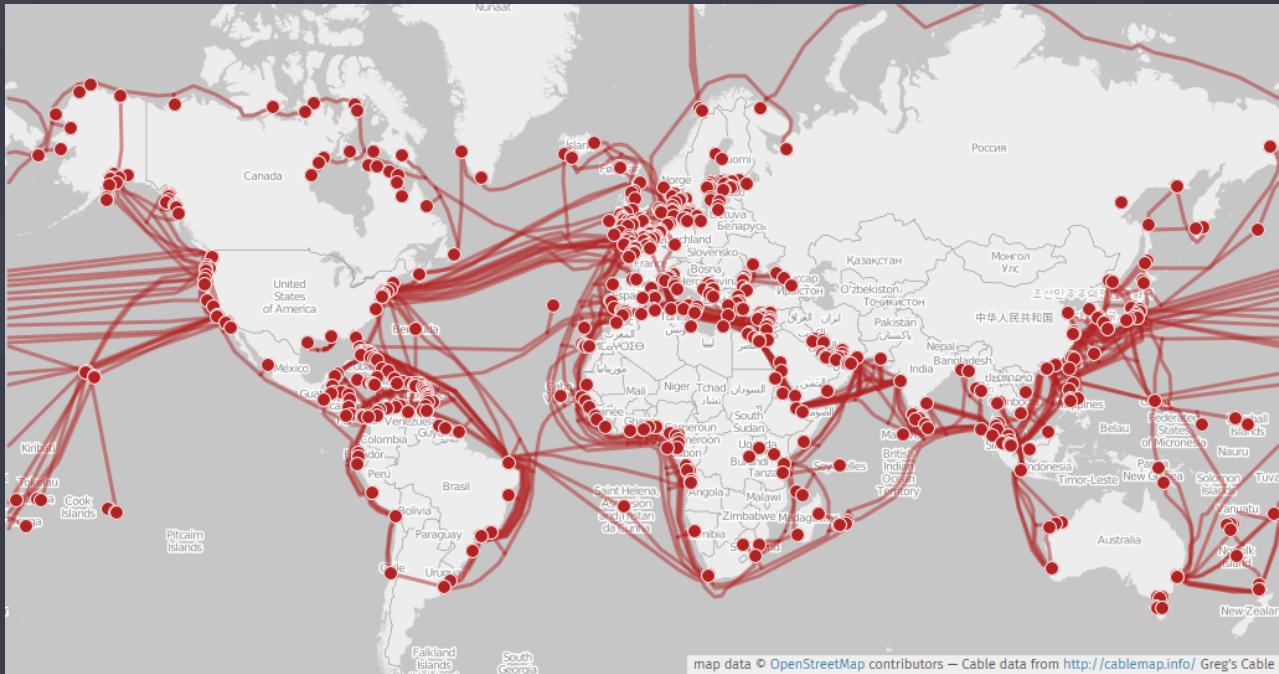
Backbone



Curiosità

- [Squali attaccano internet](#)
- [Video GeoPOP](#)

Mappa Backbone



|| WWW

- World Wide Web o W3
- Nato nel 1991 al CERN di Ginevra
- Il Suo creatore è Tim berners-Lee
- Definì il protocollo HTTP (HyperText Transfer protocol)
- Definì il linguaggio **HTML** (HyperText Markup language)
- Nel 1993 fù scritto il primo **Browser** chiamato MOSAIC
- Primo sito web: <http://info.cern.ch/hypertext/WWW/TheProject.html>



The NCSA Mosaic Resource Guide

Welcome! This page is designed to give you convenient access to some of the many resources on the World Wide Web. NCSA Mosaic information can be found on the [NCSA Mosaic for Microsoft Windows Home page](#). Please refer to our home page for the latest information about news and announcements, Win32s information, external viewers, sample background images & audio schemes, on-line documentation and many other related topics.

Information about using Mosaic's many features can be found in the Mosaic Help file. Click "Contents" in the Help menu of Mosaic to open the file. The Help file also contains a glossary of terms you may find helpful while configuring Mosaic or surfing the Internet. If you need help configuring your system for a wansock, check the readme.wn file in the Mosaic directory or ask your Internet access provider for assistance.

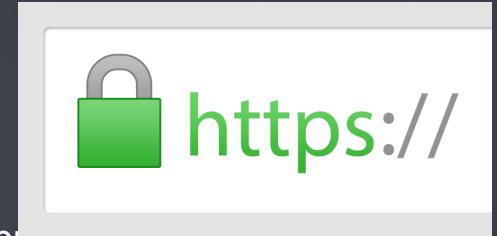
A few topics you may find interesting and helpful:

NCSA Mosaic Licensing

NCSA Mosaic is [copyrighted](#) by The Board of Trustees of the [University of Illinois \(UI\)](#), and ownership remains with the UI. The UI grants you a license without a fee the use of Mosaic software for personal, academic, research, government and internal business purposes. If you are

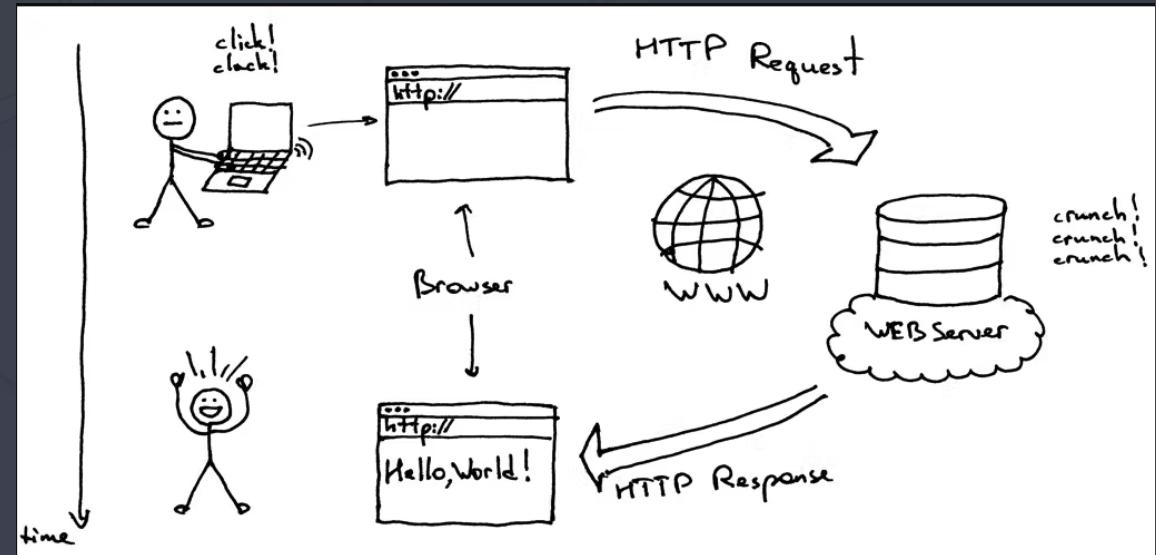
|| WWW

- **HTTP (ora HTTPS):** Un server HTTP generalmente resta in ascolto delle richieste dei client sulla porta 80 usando il protocollo TCP a livello di trasporto.
- **HTML:** L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C).
<https://github.com/whatwg/html>
- URL: Uniform Resource Locator, è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa su una rete di computer,



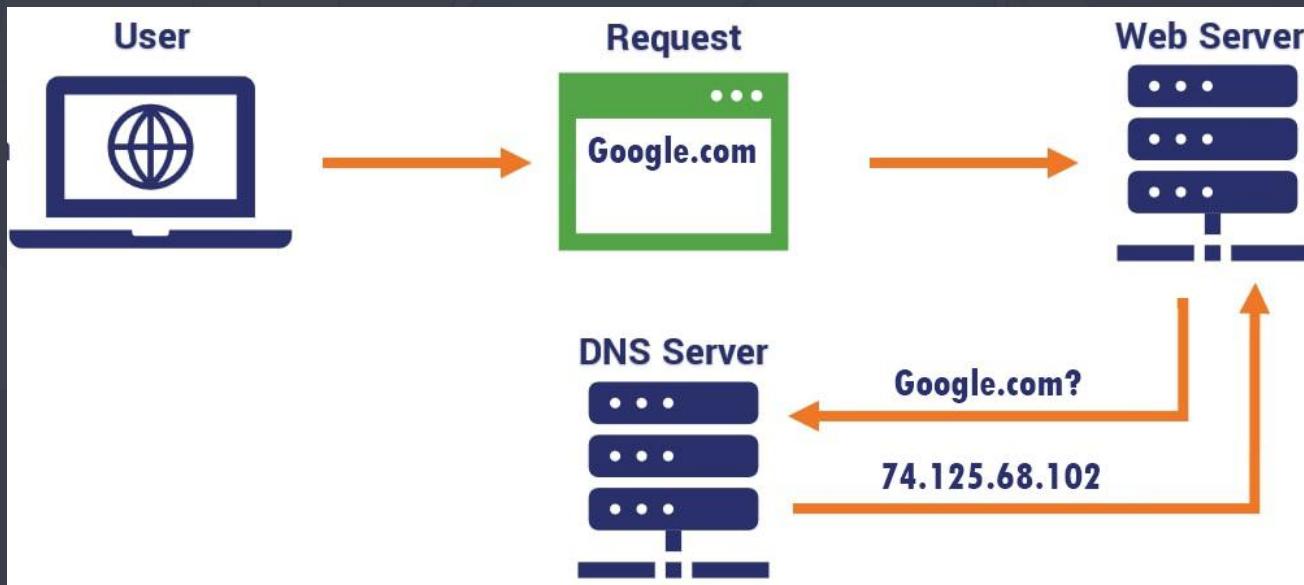
HTTP

- Modello Client/Server
- Client = Browser
- Server = Server Remoto
- Due tipi di messaggi:
 - Request = Richiesta
 - Response = Risposta



- HTTP è Stateless, ovvero il che significa che una volta termina la connessione tra il browser e il server viene persa. Si utilizzano i Cookie HTTP

HTTP - IP e DNS



Vediamo in Ispezione il sito <https://www.google.com/>

HTTP - Metodi

- Metodi:
 - GET: Ottenerne Una Risorsa, una pagina WEB. Ha sempre una Intestazione (HEADER), mai un Corpo (Body)
 - POST: Quando mandiamo una richiesta al server di creare una nuova risorsa, mandiamo un HEADER e un BODY con i dati sensibili da mandare.
 - PUT: Quando abbiamo già creato una risorsa e dobbiamo modificarla in totale, mandiamo un HEADER e un BODY;
 - PATCH: Quando dobbiamo modificare una parte della risorsa precedentemente creata, mandiamo un HEADER e un BODY;
 - DELETE: Quando dobbiamo eliminare una risorsa, mandiamo un HEADER e un BODY;

HTTP - Metodi

Create	→	POST
Read	→	GET
Update	→	PUT
Delete	→	DELETE

HTTP
Methods

HTTP - Status Code

- 1xx — Informational: Il server non ha completato completamente la richiesta, sta ancora pensando ed è in una fase di transizione.
- 2xx—Successful: Il server ha completato con successo la richiesta
- 3xx—Redirects: Questo blocco è per i reindirizzamenti, significa che hai richiesto un indirizzo ma sei stato inviato altrove
- 4xx—Client Errors: C'è qualche errore dalla tua parte
- 5xx—Server Errors: C'è qualche errore dalla parte del server

Tutti gli stati: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Mani in Pasta

Abbiamo bisogno di:

- Un programma per vedere le pagine internet
- Un programma per scrivere codice
- Un modo per dire al programma per vedere le pagine internet, come vogliamo vederle.

Vedere le pagine Internet: Bowser



Browser



Safari

Apple

MacOS, iOS



Firefox

Mozilla

MacOS, MS Windows, Linux OS,
Android OS



Chrome

Google

MacOS, MS Windows, Linux OS,
Android OS, Chrome OS



Edge new

Microsoft

MS Windows, MacOS, iOS
Android OS



Opera

Opera Software

MacOS, MS Windows, Linux OS,
Android OS



Vedere le pagine Internet: Browser

Un browser web è un'applicazione software progettata specificamente per recuperare, interpretare e visualizzare contenuti dal World Wide Web.

Funziona come un traduttore e interprete che converte il codice delle pagine web (HTML, CSS e JavaScript) in interfacce grafiche con cui possiamo interagire.

Il browser gestisce le comunicazioni con i server web, richiede i dati necessari tramite protocolli come HTTP o HTTPS, e assembla tutti gli elementi di una pagina - testi, immagini, video, script - in un'esperienza coerente e navigabile.

Oltre alla semplice visualizzazione, i browser moderni offrono funzionalità avanzate come gestione di password, sincronizzazione tra dispositivi, estensioni personalizzate e strumenti per sviluppatori, diventando vere e proprie piattaforme applicative multifunzionali.

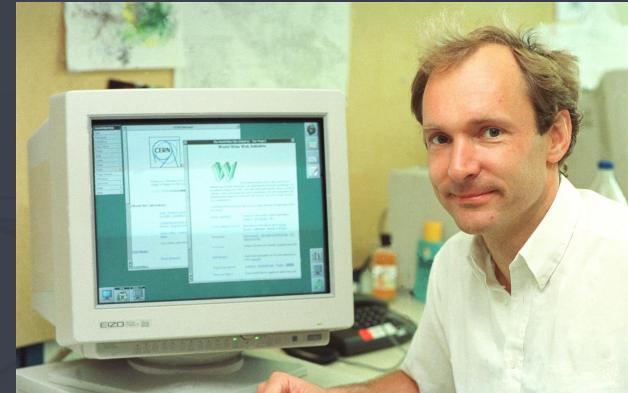
Un modo per scrivere: HTML

HTML e' un acronimo che sta per HyperText Markup Language.

E' stato sviluppato da Tim Berners-Lee e i suoi colleghi presso il CERN di Ginevra nel 1990, come parte del progetto World Wide Web.

E' uno dei tre linguaggi fondamentali per lo sviluppo frontend di un sito web, insieme a CSS e JavaScript, e ci permette di definire la struttura e il contenuto di una pagina web, come testo, immagini, video, collegamenti ipertestuali ed altri elementi.

Questa struttura che andiamo a creare in HTML fornisce quindi un'organizzazione logica e visiva al contenuto della pagina.



HTML, CSS e JS

L'HTML è uno dei tre linguaggi fondamentali per lo sviluppo frontend di un sito web, insieme a CSS e JavaScript, e ci permette di definire la struttura e il contenuto di una pagina web, come testo, immagini, video, collegamenti ipertestuali ed altri elementi.

Questa struttura che andiamo a creare in HTML fornisce quindi un'organizzazione logica e visiva al contenuto della pagina.



Index

È pratica comune (e una convenzione diffusa dei server) andare a cercare automaticamente un file chiamato *index* nella cartella principale, se non viene specificato diversamente, a prescindere dal tipo di file o dal linguaggio.

Chiamare il file principale *index.html* facilita l'identificazione del file principale di un sito web, rendendo intuitivo e chiaro quale sia la pagina principale del sito.

Non basta però aggiungere l'estensione HTML ad un file per scrivere HTML: abbiamo anche bisogno di utilizzare gli strumenti che ci mette a disposizione il linguaggio. In questo modo possiamo assegnare delle **etichette** che ci permettono di definire il contenuto di una pagina web e dare un ruolo ai nostri elementi, che prendono il nome di **tag**.

Come funziona HTML

Per cominciare a scrivere HTML, apriamo un editor di testo (anche blocco note) e salviamo un file con estensione .html:



Programma per scrivere: IDE

Un IDE (Integrated Development Environment) è un'applicazione software che offre agli sviluppatori uno spazio di lavoro integrato per scrivere, compilare, testare e debuggere codice.

In sostanza, è un software che riunisce tutti gli strumenti necessari per lo sviluppo di software in un'unica interfaccia.



Prima Pagina HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Francesco Mansi</title>
  </head>
  <body>
    <h1>Homepage </h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore</p>
  </body>
</html>
```

Esercizi

- Installare Google Chrome o un Browser equivalente
- Installare VS CODE
- Creare una pagina HTML
- Replicare la stessa pagina HTML del docente
- Validare il codice HTML a questo link: https://validator.w3.org/#validate_by_input



Modulo 1

Principi di UX/UI

Capitolo 2

Regola Numero 1 dello sviluppo web

Prima di scrivere codice:
PENSA

Principi di UX/UI

La progettazione di un sito web efficace richiede una solida comprensione di tre elementi fondamentali:

- Le fonti di ispirazione per stimolare la creatività;
- I mockup e Wireframe come strumenti di visualizzazione;
- I principi di User Experience (UX) e User Interface (UI) che guidano decisioni progettuali orientate all'utente.

Questo parte di corso esplorerà questi tre aspetti essenziali della progettazione web moderna, fornendo basi concrete per creare esperienze digitali intuitive, funzionali ed esteticamente piacevoli.

Il Valore dell'Ispirazione nel Design

Cercare ispirazione rappresenta un passaggio fondamentale nel processo creativo di ogni designer: non state copiando nulla.

L'esposizione a lavori eccellenti stimola la creatività, amplia gli orizzonti progettuali e aiuta a restare aggiornati sulle tendenze contemporanee.

Studiare soluzioni esistenti permette di comprendere come altri hanno risolto problemi simili, evitando di "reinventare la ruota" e costruendo invece su pratiche consolidate.

L'obiettivo non è mai la copia, ma la comprensione profonda dei principi che rendono efficace un design, per poi reinterpretarli in modo originale e contestualizzato al proprio progetto specifico.

Il Valore dell'Ispirazione nel Design

Il web offre numerose risorse di alta qualità per trovare ispirazione nel design.

Piattaforme come:

- Awwwards premiano e raccolgono i siti web più innovativi a livello mondiale, mentre
- Behance e Dribbble ospitano portfolio di designer professionisti con focus su UI/UX.
- Pinterest facilita la creazione di moodboard tematiche

Non va sottovalutata l'analisi dei competitor nel proprio settore, che può rivelare convenzioni efficaci e opportunità di differenziazione.

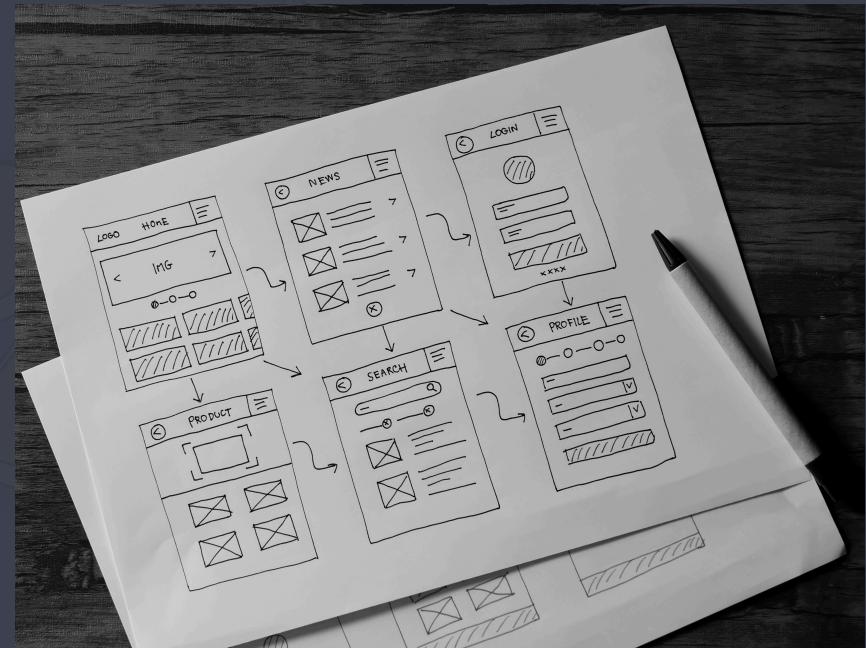
La chiave è sviluppare l'abitudine di osservare criticamente, analizzando perché certe soluzioni funzionano bene in determinati contesti.

Cos'è un Wireframe

Un wireframe è uno schema visivo semplificato che rappresenta la struttura di una pagina web o di un'applicazione.

Viene utilizzato nelle fasi iniziali di progettazione per definire la disposizione degli elementi principali, come intestazioni, testi, immagini, pulsanti e menu di navigazione.

Non si concentra sull'aspetto grafico, ma sulla funzionalità e sull'organizzazione dei contenuti, aiutando il team di sviluppo e design a condividere una visione chiara del layout prima di passare alla fase estetica e alla programmazione.

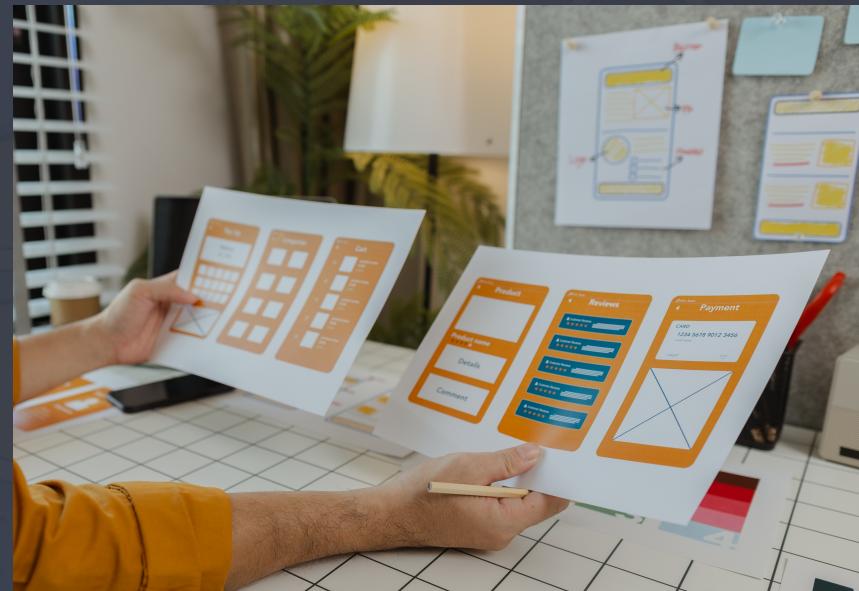


Cos'è un Mockup

Un mockup rappresenta una visualizzazione statica ma dettagliata di come apparirà un sito web o un'applicazione prima della sua effettiva realizzazione.

Diversamente da un semplice wireframe, il mockup riproduce fedelmente colori, tipografie, immagini e layout, offrendo una rappresentazione realistica del prodotto finale.

Questo strumento costituisce un ponte comunicativo essenziale tra designer, clienti e sviluppatori, permettendo di allineare aspettative e visioni prima di investire risorse nella fase di implementazione tecnica.



Perché Creare Mockup?

La creazione di mockup rappresenta una fase cruciale nel processo di design che offre molteplici vantaggi.

Innanzitutto, permette di visualizzare concretamente le idee, trasformando concetti astratti in rappresentazioni tangibili.

Facilita inoltre la raccolta di feedback precoce da stakeholder e potenziali utenti, riducendo significativamente il rischio di costose revisioni nelle fasi avanzate del progetto.

I mockup consentono anche di sperimentare diverse soluzioni di design in modo rapido ed economico, accelerando l'iterazione e il perfezionamento del prodotto finale.

I tool adatti alla creazione di mockup grafici sono:



UX e UI: Fondamenti e Distinzioni

User Experience (UX) e User Interface (UI) rappresentano due discipline complementari ma distinte nel campo del design digitale.

La UX si concentra sull'esperienza complessiva dell'utente nell'interazione con un prodotto, considerando aspetti come l'utilità, l'usabilità e la soddisfazione emotiva. Il design UX è guidato da ricerca, dati comportamentali e principi psicologici, con l'obiettivo di risolvere problemi reali degli utenti.

La UI, invece, si focalizza sugli aspetti visivi e interattivi dell'interfaccia: colori, forme, tipografia e layout che l'utente vede e con cui interagisce direttamente.

Entrambe sono essenziali per creare prodotti digitali di successo, operando in sinergia durante tutto il processo di progettazione.



Principi Fondamentali di UX

La User Experience si fonda su principi chiave che guidano ogni decisione progettuale.

- L'usabilità (usability) garantisce che il prodotto sia facile da usare e da apprendere, mentre l'accessibilità assicura che sia utilizzabile da tutti, incluse persone con disabilità.
- L'utilità (usefulness) verifica che il prodotto risolva effettivamente problemi reali degli utenti, e la desiderabilità aggiunge una dimensione emotiva positiva all'esperienza.
- La chiarezza (clarity) riduce al minimo confusione e frizioni cognitive, mentre la consistenza crea comportamenti prevedibili e familiari attraverso l'interfaccia.
- L'efficienza (efficiency), infine, permette agli utenti di raggiungere i propri obiettivi con il minimo sforzo necessario, rispettando il loro tempo e le loro energie.

Principi Fondamentali di UI

Il design dell'interfaccia utente si basa su principi visivi e interattivi che determinano l'efficacia comunicativa del prodotto.

- La gerarchia visiva organizza gli elementi secondo la loro importanza, guidando l'attenzione dell'utente.
- La leggibilità assicura che i testi siano facilmente fruibili in ogni contesto e dimensione di schermo.
- La coerenza stilistica e comportamentale crea un'esperienza prevedibile, mentre il feedback visivo conferma all'utente che le sue azioni sono state registrate dal sistema.
- Il concetto di affordance (invito all'uso) suggerisce visivamente come interagire con gli elementi dell'interfaccia, e il bilanciamento crea una distribuzione armoniosa degli elementi nello spazio.
- Il contrastò, infine, differenzia gli elementi secondo la loro rilevanza, creando punti focali strategici.



Ispirazioni, Mockup e UX/UI

Mockup, ispirazioni e principi UX/UI non sono elementi isolati ma componenti interconnesse di un processo di design olistico.



Esercizi

- Utilizzando Miro, creare lo stesso mockup fatto a lezione dal docente



Modulo 1

Introduzione HTML

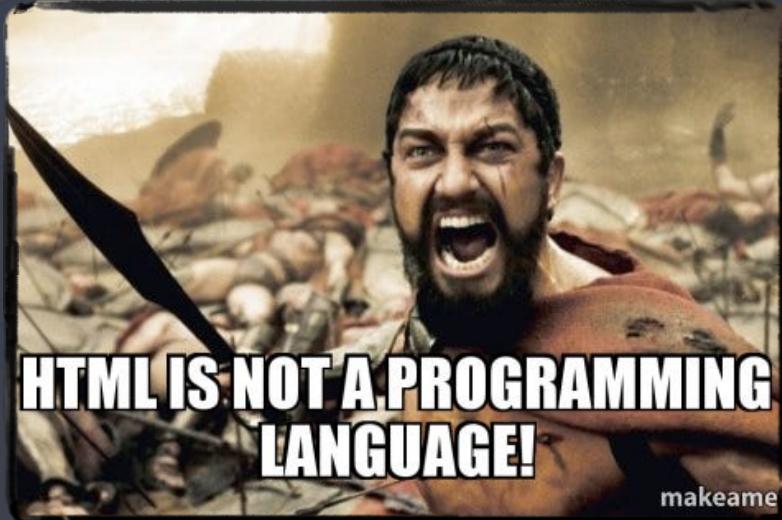
Capitolo 3

HTML

HTML è l'acronimo di HyperText Markup Language, sviluppato da Tim Berners-Lee nel 1990 presso il CERN di Ginevra.

È uno dei tre linguaggi fondamentali per lo sviluppo web, insieme a CSS e JavaScript.

Non è un linguaggio di programmazione ma un linguaggio di marcatura, che ci permette di definire la struttura e il contenuto delle pagine web.



Prima Pagina HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Francesco Mansi</title>
  </head>
  <body>
    <h1>Homepage </h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore</p>
  </body>
</html>
```

Struttura base

Ogni file HTML ha bisogno di una struttura base che include:

- `<!DOCTYPE html>`: Indica al browser che stiamo usando HTML5
- `<html>`: Tag radice che contiene tutta la pagina
- `<head>`: Contiene informazioni sulla pagina (metadati)
- `<body>`: Contiene tutto il contenuto visibile della pagina e i vari TAG

DOCTYPE

Il primo elemento che troviamo è una dichiarazione:



```
<!DOCTYPE html>
```



Utilizzare HTML5 come standard per il tuo sito web assicura che il codice sia interpretato in modo uniforme e corretto dai browser moderni, garantendo una migliore compatibilità e accessibilità del sito.

Subito sotto troviamo un primo TAG:

Tag

Politica e governo · Di tendenza

Ursula

24.900 post

Di tendenza nella seguente zona: Italia

#AscoltiTv

Sport · Di tendenza

Venezia

5.049 post

Di tendenza nella seguente zona: Italia

Buon Lunedì

1.320 post

Sport · Di tendenza

Conference

104.000 post

#foodblogger
72,5 mln post

#foodpornography
2,2 mln post

#foodphotography
96,9 mln post

#foodie
218 mln post

#foodstagram
106 mln post

#foodpics
47,9 mln post

#foodstylist
4,5 mln post

#foodblog
22,9 mln post

#foodstyling
13,8 mln post



I Tag HTML

I tag sono delle etichette racchiuse tra parentesi angolari che attribuiscono significato a specifici contenuti. Generalmente hanno un tag di apertura e uno di chiusura.

Alcuni tag non necessitano di un tag di chiusura separato perché non contengono alcun contenuto all'interno. Questi sono i self-closing tag.

```
<tag>contenuto</tag>
```

```
<tag/>
```

Il contenuto tra i tag è ciò che sarà visualizzato nella pagina, mentre i tag stessi dicono al browser come interpretare e mostrare quel contenuto.

Tag HEAD

Il tag `<head>` funziona come il cervello della nostra pagina web: contiene informazioni importanti ma non visibili direttamente.

Al suo interno troviamo:

- `<meta>`: Fornisce informazioni sul documento (codifica, viewport)
- `<title>`: Definisce il titolo della pagina che appare nella scheda del browser

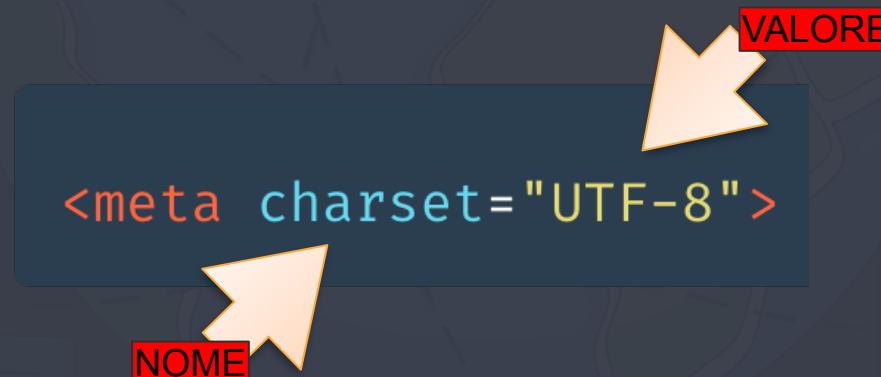
Questi elementi aiutano browser e motori di ricerca a capire meglio la nostra pagina.

```
<head>  
    <meta charset="UTF-8">  
    <title>Titolo della Pagina</title>  
</head>
```

Attributi HTML

Gli attributi forniscono informazioni aggiuntive agli elementi HTML e sono sempre specificati nel tag di apertura.

Sono formati da una coppia chiave-valore, dove la chiave è il nome dell'attributo e il valore specifica le informazioni da applicare all'elemento: id, class, type, style ecc..



Attributi HTML

Gli attributi forniscono informazioni aggiuntive agli elementi HTML e sono sempre specificati nel tag di apertura.

Sono formati da una coppia chiave-valore, dove la chiave è il nome dell'attributo e il valore specifica le informazioni da applicare all'elemento: id, class, type, style ecc..

Tag BODY

Il tag <body> contiene tutto ciò che gli utenti vedranno nella pagina web.

È come il corpo fisico della nostra pagina.

All'interno del body inseriamo testo, immagini, link, tabelle, moduli e tutti gli altri elementi visibili che compongono la nostra pagina web.

Ogni elemento che vogliamo mostrare all'utente deve essere posizionato all'interno di questo tag.

```
<body>  
  <p>Paragrafo</p>  
</body>
```

Contenitori Generici

Per raggruppare il contenuto senza aggiungere significato semantico si utilizzano `<div>` e ``.

- Il `<div>` è un contenitore a blocco, occupa tutto lo spazio orizzontale disponibile e gli elementi successivi vanno a capo. Viene usato quando vogliamo creare una DIVisione;
- Lo `` è un contenitore inline, occupa solo lo spazio del suo contenuto. Perfetto se vogliamo applicare uno stile solo ad una parola.

Sono come scatole vuote che possiamo riempire e organizzare come vogliamo.

```
<div>Tutta la pagina</div>
```

```
Ciao mi chiamo <span>Francesco</span>
```

Heading

I tag heading (`<h1>` fino a `<h6>`) definiscono i titoli e sottotitoli della pagina, dove `<h1>` è il più importante e `<h6>` il meno importante.

È consigliabile usare un solo `<h1>` per pagina e mantenere una gerarchia logica nei titoli per favorire SEO e accessibilità.

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

`<h1>Heading 1</h1>``<h2>Heading 2</h2>``<h3>Heading 3</h3>``<h4>Heading 4</h4>``<h5>Heading 5</h5>``<h6>Heading 6</h6>`

Paragrafi

Il tag `<p>` definisce un paragrafo di testo, creando automaticamente uno spazio sopra e sotto di esso.

È uno degli elementi più fondamentali per organizzare il testo in blocchi distinti, rendendo il contenuto più leggibile e strutturato.

```
<p>Paragrafo</p>
```

Liste

HTML offre due tipi principali di liste:

- : Liste non ordinate con bullet points
- : Liste ordinate con numeri

Entrambe utilizzano (list item) per definire i singoli elementi della lista.

1. Coffee
2. Tea
 - Green tea
 - Black tea
3. Milk

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Green tea</li>
      <li>Black tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ol>
```

Link

Il tag `<a>` (anchor) crea collegamenti ipertestuali che permettono agli utenti di navigare tra le pagine web.

```
<a href="google.it">Clicca Qui </a>
```

L'attributo `href` (Hypertext REference) contiene l'URL di destinazione, e il testo tra i tag è ciò che l'utente vede e può cliccare.

Immagini

Il tag inserisce immagini nella pagina web ed è un self-closing tag.

```

```

L'attributo `src` specifica la posizione dell'immagine (URL o percorso), mentre `alt` fornisce una descrizione testuale fondamentale per l'accessibilità.

URI vs URL

URL assoluto:

Collega a un'immagine esterna

Esempio: `src="https://www.w3schools.com/images/img_girl.jpg"`

URL relativo:

Collega a un'immagine interna al sito

Non include il nome di dominio

Esempio: `src="img_girl.jpg"` o `src="/images/img_girl.jpg"`

Consigliato per evitare problemi di dominio

Esercizi

- Realizza una pagina HTML con tutti gli elementi visti fino ad adesso.



Modulo 1

La semantica di HTML

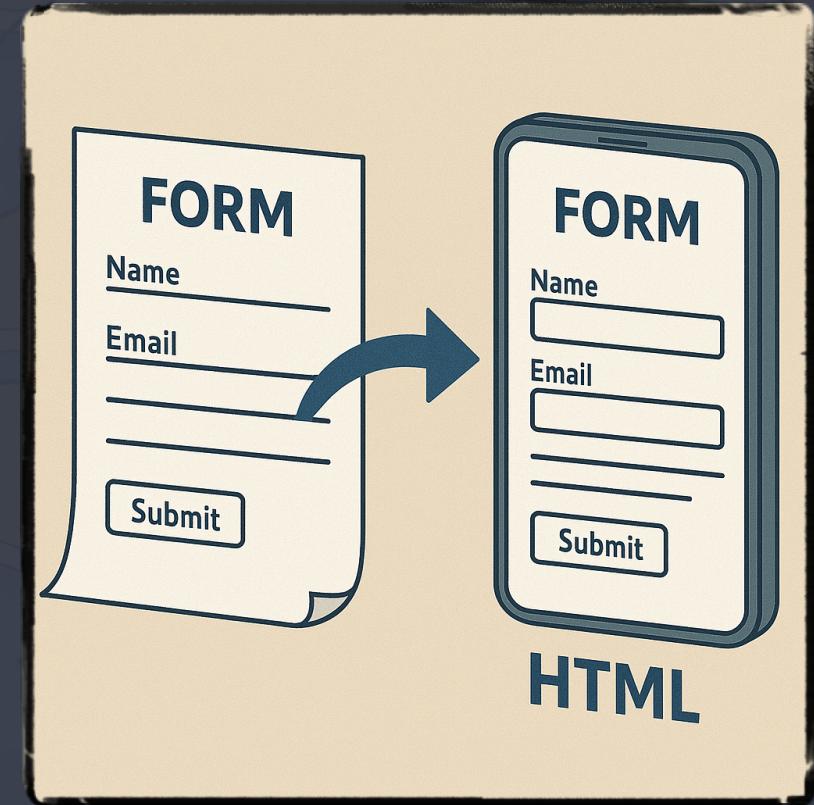
Capitolo 4

Il Tag FORM

Il tag `<form>` crea moduli interattivi per raccogliere input dagli utenti.

```
<form action="/invia-dati" method="post">  
    ← Elementi del form →  
</form>
```

L'attributo `action` specifica dove inviare i dati, mentre `method` determina il metodo HTTP da utilizzare (GET o POST).



Input

Il tag <input> crea diversi tipi di campi per l'inserimento dati nei form.

L'attributo type determina il tipo di campo (testo, password, checkbox, ecc.), mentre placeholder mostra un testo di esempio.

A questo link tutti quelli supportati ufficialmente:



<https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/input>

Label

Il tag <label> associa un'etichetta descrittiva a un campo di input.

```
<label for="nome">Nome:</label>  
  
<input type="text" id="nome">
```

L'attributo `for` collega la label all'input con l'id corrispondente, migliorando l'usabilità (cliccando sulla label si attiva il campo) e l'accessibilità.

Pulsanti

Il tag <button> crea pulsanti interattivi nei form.

```
<button type="submit">Invia</button>  
<button type="reset">Annulla</button>  
<button type="button">Fai Qualcosa</button>
```

I tipi principali sono: `submit` (invia i dati del form), `reset` (ripristina i valori predefiniti) e `button` (per azioni personalizzate con JavaScript).

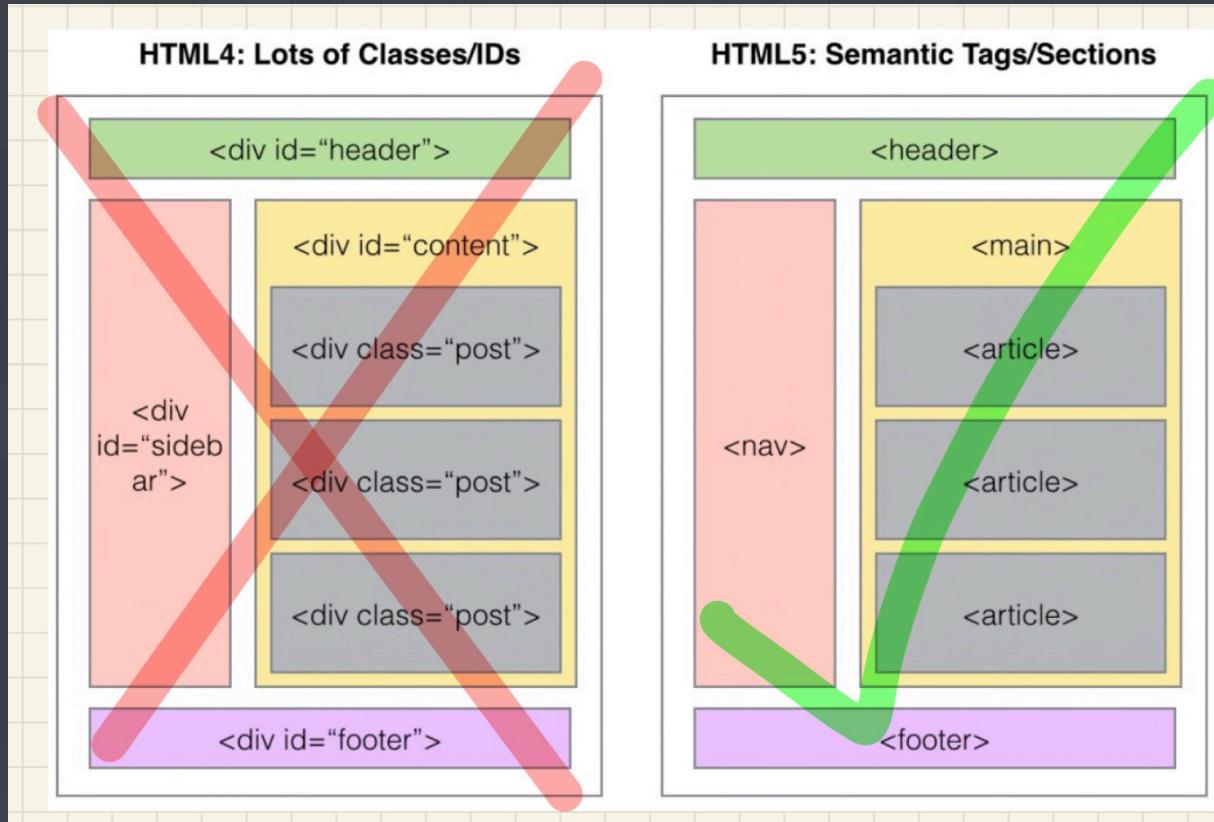
HTML5 e i TAG semantici

HTML5 ha introdotto tag semantici che descrivono chiaramente il loro significato sia agli sviluppatori che ai browser.

Questi tag (come `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<footer>`) aiutano a strutturare logicamente la pagina, migliorando accessibilità e SEO.

A differenza dei generici `<div>`, comunicano chiaramente la funzione di ciascuna sezione.

HTML5 e i TAG semantici



Tag di navigazione: NAV

Il tag <nav> definisce una sezione dedicata ai link di navigazione principale del sito.

```
<nav>  
    <a href="index.html">Home</a>  
    <a href="about.html">Chi siamo</a>  
    <a href="contact.html">Contatti</a>  
</nav>
```

Indica chiaramente ai browser e alle tecnologie assistive che i contenuti al suo interno sono parte della navigazione del sito.

Tag di Intestazione: HEADER

Il tag <header> rappresenta l'intestazione di una pagina o di una sezione specifica.

```
<header>  
    <h1>Il mio sito web</h1>  
    <nav><!-- Menu di navigazione --></nav>  
</header>
```

Tipicamente contiene elementi introduttivi come loghi, titoli, sottotitoli e spesso anche il menu di navigazione principale.

Contenuto Principale: MAIN

Il tag <main> identifica il contenuto principale della pagina, escludendo header, footer e barre laterali.

```
<main>
  <h2>Articolo del giorno</h2>
  <p>Contenuto principale della pagina ... </p>
</main>
```

Dovrebbe essere unico nella pagina e aiuta screen reader e motori di ricerca a individuare rapidamente il contenuto centrale.

Sezioni: SECTION e ARTICLE

`<section>` raggruppa contenuti correlati tematicamente, mentre `<article>` rappresenta contenuti autonomi e indipendenti.

```
<section>

    <h2>Notizie recenti</h2>

    <article>

        <h3>Titolo articolo</h3>

        <p>Contenuto dell'articolo ... </p>

    </article>

</section>
```

Un blog potrebbe avere una `<section>` "Post recenti" contenente vari `<article>` individuali.

Pie di pagina: FOOTER

Il tag <footer> definisce il piè di pagina di un documento o di una sezione.

```
<footer>  
  <p>© 2025 Il mio sito. Tutti i diritti riservati.</p>  
  <a href="privacy.html">Privacy</a>  
</footer>
```

Tipicamente contiene informazioni come copyright, contatti, link ai social media e altri collegamenti utili ma secondari.

E molti altri tag ...

```
<p>Questo è <strong>molto importante</strong> e questo merita  
<em>particolare enfasi</em>.</p>
```



Vai qui: <https://www.w3schools.com/tags/>

Esercizi

- Crea una pagina HTML completa rispettando tutti i principali criteri di semantica visti