



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Programmazione Web

Contenuti del Corso

Esercitazione 1

24.03.2025

DOCENTE

Paolo Fosci

EMAIL

paolo.fosci@unibg.it

TITOLARE DEL CORSO

Prof. Giuseppe Psaila

Docente: Paolo Fosci

Per qualsiasi domanda o dubbio sugli esercizi scrivetemi una mail a:

- paolo.fosci@unibg.it

Nell'oggetto delle mail **usate il prefisso** [\[PW25\]](#)

per essere identificati come studenti del corso di Programmazione Web
(e ricevere una risposta quanto più immediata, **anziché finire nello spam**)

NON SCRIVETEMI SU TEAMS!

Calendario Provvisorio (possibile di emendamenti)

1. **Lunedì 24 Marzo : HTML + CCS**
2. **Lunedì 24 Marzo : Javascript**
3. **Lunedì 31 Marzo : Server Side Script**
4. **Lunedì 31 Marzo : JQuery**

**Oggetto del
1° progetto**

5. Lunedì 05 Maggio : Node JS

A. Venerdì 09 Maggio : Seminario su usabilità e accessibilità (da confermare)

6. Lunedì 12 Maggio : Database
7. Lunedì 19 Maggio : Python + DB
8. Lunedì 26 Maggio : Bootstrap (probabilmente)

Per passare l'esame di Programmazione WEB, è necessario

- Passare un esame scritto: Min 9pt – Max 16pt
(test a risposta chiusa
su tutto il programma del Prof. Psaila)
- Realizzare un primo Progetto di gruppo: Min 4.5pt – Max 8pt
**Il Progetto per ogni gruppo è deciso
dal docente e verrà comunicato più avanti!**
- Realizzare un secondo Progetto di gruppo: Min 4.5pt – Max 8pt
legato al primo Progetto
(stesso gruppo del primo Progetto)
- Per realizzare i progetti avete un anno di tempo (fino all'inizio del Corso nel 2026!)

- I gruppi devono essere formati da 2 o 3 persone
- Per comunicare la composizione dei gruppi usare il form disponibile sul canale «Esercitazioni» di Teams (il form sarà attivo dalle 18:00 fino al 28 aprile a mezzanotte)
- Le comunicazioni del gruppo verso il docente devono avvenire **necessariamente:**
 - con tutti i membri in CC
 - Con **[PW25]** e il nome del gruppo come prefissi nell'oggetto della mail
 - **Altrimenti le mail saranno cestinate**

Link al form per indicare il
gruppo di lavoro:

<https://forms.office.com/e/DT1aHeJpBJ>

PW25 Gruppi di lavoro



Strumenti di lavoro (per la prima parte di corso):

- Altervista.org (online free) - <https://it.altervista.org/>
- Notepad++ (free)
- HTMLPad (30 runs trial) - <https://www.htmlpad.net/>
- Visual Studio (free, complex) - <https://visualstudio.microsoft.com/it/>
- Dreamweaver (to buy)

Documentazione : **W3School**

- HTML : <https://www.w3schools.com/html/default.asp>
- CSS : <https://www.w3schools.com/css/default.asp>
- JS : <https://www.w3schools.com/js/default.asp>

Per la prossima lezione:

Attivate un profilo Altervista

- Dati sintentici
https://en.wikipedia.org/wiki/Synthetic_data
- Usabilità
<https://it.wikipedia.org/wiki/Usabilit%C3%A0>
- Accessibilità
[https://it.wikipedia.org/wiki/Accessibilit%C3%A0 \(web\)](https://it.wikipedia.org/wiki/Accessibilit%C3%A0_(web))
Legge Stanca - [https://it.wikipedia.org/wiki/Legge Stanca](https://it.wikipedia.org/wiki/Legge_Sanca)



RTFM

Read The F*cking Manual

- Linguaggio di Markup
- Composto da <Tag> di apertura e </Tag> di chiusura
 - Ma esistono anche <Tag/> autoconclusivi
- Tra i tag possiamo avere sia testo che altri tag (valgono le regole per le parentesi)
- All'interno dei tag possiamo avere degli attributi nella forma *attributo= "valore"*

- I tag non hanno un valore semantico specifico!
- Il valore semantico viene dato da come li usiamo e dal foglio di stile... (per cui se eliminiamo il foglio di stile...)

MA...

- I tag sono stati introdotti per gestire esigenze editoriali specifiche e andrebbero usati di conseguenza. Quindi...

Quindi:

- Una pagina HTML è tra i tag `<body> ... </body>`
- Il tag `<nav>` si usa per gestire la navigazione
- Il tag `<header>` si usa per definire l'intestazione della pagina
- Il tag `<footer>` si usa per definire il fondo della pagina
- I tag `` e `` si usano per le liste, rispettivamente non ordinate (`unordered`) e ordinate (`ordered`)
- Il tag `<table>` si usa per definire i dati tabellari
- Il tag `<div>` si usa per contenere tutti quegli elementi generici che danno struttura e schema alla pagina
- **MA SOPRATTUTTO NON USARE `<table>` per definire lo schema di una pagina**

- I fogli di stile (CSS) servono a dare forma e significato ai tag
- Per ogni tag dobbiamo considerare vari elementi quali:
 - l'area occupata
 - il bordo (che può avere dimensioni)
 - Il contenuto
 - Il margine (spazio tra il bordo e il confine dell'area occupata – in altre parole, lo spazio minimo con l'elemento successivo)
 - Il padding (spazio tra il bordo e il contenuto del tag)

- Sebbene in ogni tag si possa definire lo stile in-line (cioè tramite l'attributo *style*) questa pratica è fortemente scoraggiata ed è preferibile usare un file a parte per il foglio di stile da includere nella pagina HTML
- Il foglio di stile ci dice dove posizionare i tag
- Il «floating» è quella caratteristica che spinge i tag a sx (left) o dx (right) in base allo spazio disponibile

- I fogli di stile (CSS) servono a dare forma e significato ai tag
- Per ogni tag dobbiamo considerare vari elementi quali:
 - l'area occupata
 - il bordo
 - Il contenuto
 - Il margine (spazio tra il bordo e il confine dell'area occupata)
 - Il padding (spazio tra il bordo e il contenuto del tag)

Gli elementi di una pagina HTML si possono referenziare in un foglio di stile tramite:

- Il nome del tag. Es:

div {...}

- Un pattern di tag. Es:

div ul {...}

- L'identificatore univoco di un tag, tramite l'attributo «*id*». Es:

#p2 {...}

- La classe di un tag, tramite l'attributo «*class*»

.giallo {...}

- Un qualsiasi pattern combinando tag, id e classi

- Con i fogli di stile possiamo definire quindi posizione e aspetto di ogni elemento, anche quando la pagina cambia dimensioni
- Oppure quando avviene qualche evento

- Ovviamente, in una pagina HTML, con il suo foglio di stile tutto va organizzato in maniera tale da fornire all'utente una pagina gradevole, funzionale, e accessibile
- Quindi abbiate uno stile coerente, pulito, essenziale
- ED EVITATE LE ARLECCHINATE (aka PAGLIACCiate)!