

Web Design

prof.ssa Cristina Iurissevich
cristinaiurissevich@abacatania.it

Indice lezione:

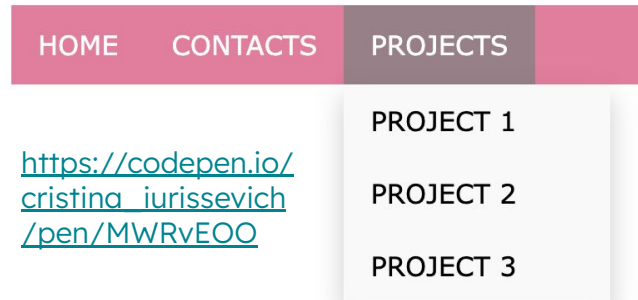
1. Menù
2. Bootstrap
3. UI
4. Usabilità
5. Accessibilità
 - a. Principi
 - b. Accessibilità e Design Inclusivo

Menu

Menu HTML / CSS



https://codepen.io/cristina_iurissevich/pen/PogKige



https://codepen.io/cristina_iurissevich/pen/MWRvEOO

Menu con hamburger menu:

<https://codepen.io/alvarotrigo/pen/MWEJEWG>
<https://codepen.io/erikterwan/pen/EVzeRP>

Bootstrap

Framework CSS

I fogli di stile CSS sono difficili da organizzare, mantenere e riutilizzare. Anche piccole modifiche di stile richiedono la scrittura di nuove regole CSS.

Le classi pronte all'uso sono i principali elementi costitutivi di tutti i framework CSS. Consentono di applicare regole di stile predefinite agli elementi HTML, come margini, colori di sfondo e altri.

Alcuni framework includono componenti predefiniti come menu, schede o tabelle. L'uso di questi componenti ti consente di creare interfacce user-friendly senza un eccessivo lavoro.

I framework CSS rendono il flusso di lavoro pulito e gestibile.



Tailwind CSS

<https://tailwindcss.com/>



BULMA.io

<https://bulma.io/>



Bootstrap

<https://getbootstrap.com/>

Bootstrap

— — —

Bootstrap è attualmente il framework CSS più utilizzato.

Twitter ha introdotto questo framework nel 2011 per rendere il responsive web design facilmente accessibile agli sviluppatori.

Le librerie software di Bootstrap sono un insieme di strumenti per costruire siti web responsive, cioè quei siti il cui layout delle pagine web si regola in maniera automatica, tenendo conto cioè delle caratteristiche del dispositivo utilizzato, sia esso desktop, smartphone o tablet.

Bootstrap framework è tra i progetti open source più popolari esistenti e permane un ottimo supporto e aggiornamento.

<https://getbootstrap.com/>

Responsive

Bootstrap ha guidato l'assalto alla progettazione statica del web e ha contribuito a stabilire il design mobile-first come standard del settore.

Breakpoint della larghezza dello schermo di Bootstrap:

- telefoni in landscape (576px)
- tablet (768px)
- laptop (992px)
- schermi extra large del desktop (1200px).)

Collegare Bootstrap

Nei progetti web possono essere inseriti i CSS e JS di Bootstrap.

Devono essere inseriti i tag <link> nel <head> per il CSS:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

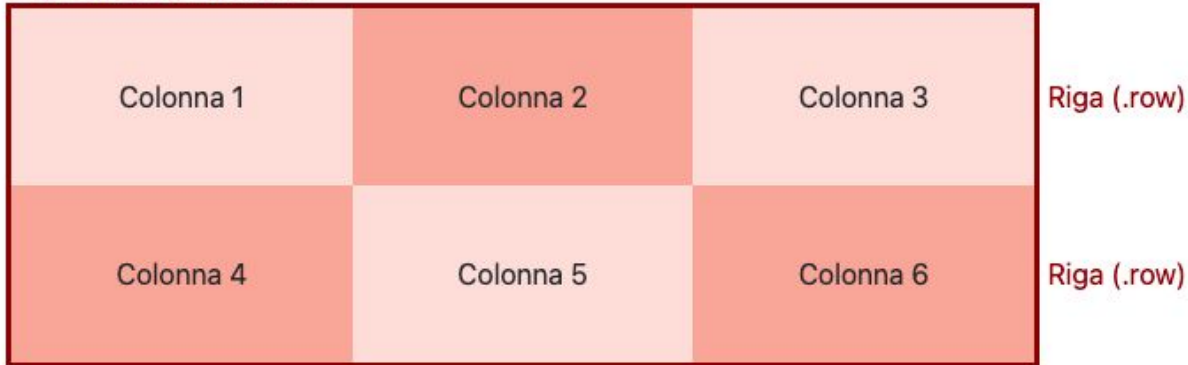
Possono essere collegati anche Javascript e Popper (per ulteriori funzionalità).

Grid System

Bootstrap si rivela uno strumento particolarmente utile per creare il layout di base del proprio progetto web anche grazie al bootstrap grid system. Composto da quattro elementi: contenitore, riga, colonna e breakpoint.

Il contenitore rappresenta la parte più esterna del grid system. È possibile crearne uno aggiungendo a un tag `<div>` la classe “container”. Ciascun contenitore a sua volta può includere una o più righe (`<div>` con classe “row”), che servono ad “avvolgere” la parte più interna (e flessibile) del grid system: le colonne (`<div>` con classe “col”).

Contenitore (.container)



```
<div class="container">
  <div class="row">
    <div class="col">Colonna 1</div>
    <div class="col">Colonna 2</div>
    <div class="col">Colonna 3</div>
  </div>
  <div class="row">
    <div class="col">Colonna 4</div>
    <div class="col">Colonna 5</div>
    <div class="col">Colonna 6</div>
  </div>
</div>
```

Grid System

Il grid system di Bootstrap divide il contenuto di ciascuna riga in 12 parti. Possiamo quindi specificare la larghezza di ciascuna colonna in dodicesimi, associando alla classe “col” un suffisso numerico che va da 1 a 12.

Contenitore (.container)

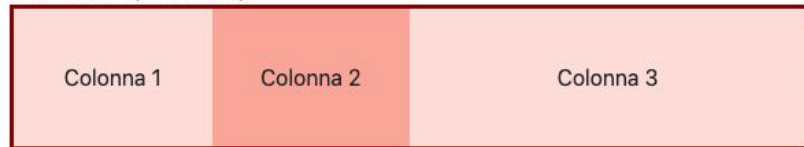


Riga (.row)

```
<div class="container">
  <div class="row">
    <div class="col-3">Colonna 1</div>
    <div class="col-3">Colonna 2</div>
    <div class="col-6">Colonna 3</div>
  </div>
</div>
```

Il grid system di Bootstrap divide il contenuto di ciascuna riga in 12 parti. Possiamo quindi specificare la larghezza di ciascuna colonna in dodicesimi, associando alla classe “col” un suffisso numerico che va da 1 a 12.

Contenitore (.container)



Riga (.row)

“Colonna 1” e “Colonna 2” occupano 3 unità ciascuna, mentre “Colonna 3” ne occupa 6. Totale: 12. Ovviamente “Colonna 3” sarà larga quanto le prime due, occuperà il 50% del contenitore indipendentemente dalla larghezza del viewport.

```
<div class="container">
  <div class="row">
    <div class="col-3">Colonna 1</div>
    <div class="col-3">Colonna 2</div>
    <div class="col-6">Colonna 3</div>
  </div>
</div>
```

Classi

In Bootstrap, le classi sono dei nomi predefiniti che applicano stili CSS a elementi HTML. Per esempio, la classe `btn` definisce un bottone con uno stile specifico, mentre la classe `text-center` allinea il testo al centro.

Le classi vengono applicate direttamente agli elementi HTML attraverso l'attributo `class`.

Le principali classi di Bootstrap sono:

- Contenitori (`bootstrap container`): impiegati per definire la struttura e l'allineamento del layout
- Griglia (`bootstrap grid`): per posizionare i contenuti sulla pagina con `.row` ed altre classi
- Form (`bootstrap form`): per creare form con elementi come `input`, `label`, pulsanti, selezioni e molto altro
- Bottoni (`bootstrap button`): per creare pulsanti con varie dimensioni, colori e stili
- Barra di navigazione (`bootstrap navbar`) per la creazione di link di navigazione.

UI

UX

La "User Experience", spesso abbreviata in "UX", è la qualità dell'esperienza che ha una persona quando interagisce con uno specifico design.

Il design dell'esperienza utente (UXD) è il processo che i team di design utilizzano per creare prodotti che forniscono esperienze significative e rilevanti agli utenti. Ciò comporta la progettazione dell'intero processo di acquisizione e integrazione del prodotto, compresi gli aspetti di branding, design, usabilità e funzione.

Il Dr. Donald Norman, un ricercatore di scienze cognitive, ha definito per primo l'importanza del design incentrato sull'uso, un processo di progettazione basato sulle esigenze e gli obiettivi dell'utente.

UX Design

User research +



End to End experience +



Content design +



Customer journey, use case +



UI Design

+ Typography, colours



+ Visual experience

+ Web/App UI Layouts



+ Graphics: look & feel



UI

Il design dell'interfaccia utente rappresenta il processo di costruire interfacce grafiche, concentrandosi sull'aspetto, stile, colori e tipografia.

Principi basilari:

- Contrasto
- Allineamenti
- Ripetizione
- Prossimità

Contrasto

Il suo scopo è organizzare la pagina. Se due elementi non sono esattamente gli stessi bisogna renderli differenti l'uno dall'altro.

- Creare focus
- Mettere in evidenza gli elementi dominanti
- Creare una gerarchia

Allineamenti

Organizzare il layout in modo tale che risulti essere ordinato.

- Creazione di linee immaginarie di lettura delle informazioni
- Estetica elegante

Ripetizione

Creare consistenza e aggiungere interesse visivo.

- Elementi visivi omogenei (font, dimensioni, colori, forme, ...)

Prossimità

Organizzare in gruppi le varie parti dell'interfaccia. Tutti gli elementi correlati devono essere raggruppati insieme di modo da creare informazioni coese.

- Migliore organizzazione
- Struttura chiara

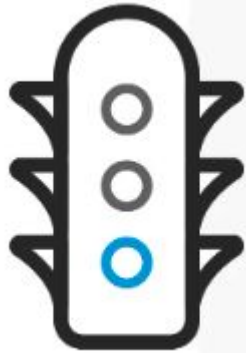
Usabilità

Le 10 euristiche di usabilità - Jakob Nielsen e Rolf Molich

— — —

- Visibilità dello stato del sistema.
 - Allineamento tra il mondo del sistema e quello reale.
 - Controllo da parte dell'utente e sua libertà.
 - Coerenza e standard.
 - Prevenzione degli errori.
 - Riconoscimento piuttosto che memorizzazione.
 - Flessibilità ed efficienza d'uso.
 - Design estetico e minimalista.
 - Aiutare l'utente nel riconoscere, diagnosticare e rimediare gli errori
- Aiuto e documentazione

1. Visibilità dello stato di sistema



Il sistema deve costantemente informare l'utente su ciò che sta accadendo, fornendo il feedback adatto in tempi ragionevoli.

- Comunicare chiaramente agli utenti qual'è lo stato del sistema
- Dare feedback immediati all'utente (o quantomeno il più rapidamente possibile).
- Creare fiducia attraverso una comunicazione aperta e continua.

2. Allineamento tra il sistema e il reale



Il sistema dovrebbe usare lo stesso linguaggio dell'utente, con parole, frasi e concetti familiari, piuttosto che usare termini orientati al sistema dovrebbe seguire le convenzioni del mondo reale.

- Assicurati che gli utenti possano comprendere il significato di una parola.
- Non dare mai per scontato che la tua comprensione di parole o concetti corrisponda a quella dell'utente.

4. Coerenza e standard



Gli utenti non devono chiedersi se parole, situazioni o azioni diverse fanno riferimento alle stesse cose. Si dovrebbero seguire le convenzioni stabilite dagli standard ufficiali o di fatto.

- Mantenere la coerenza all'interno di un singolo prodotto o di una famiglia di prodotti (coerenza interna).
- Seguire le convenzioni del settore stabilite (coerenza esterna).
- Migliora l'apprendimento mantenendo entrambi i tipi di coerenza: interna ed esterna.

5. Prevenzione degli errori



Un progetto che previene gli errori è meglio dei messaggi di errore.

- Evita errori fornendo vincoli utili e buone impostazioni predefinite.
- Previeni gli errori senza richiedere all'utente di memorizzare funzionamenti, supportando l'annullamento e indirizzando correttamente i percorsi.

6. Riconoscimento piuttosto che memorizzazione



Rendere visibili gli oggetti, le azioni e le opzioni. Le istruzioni su come usare il sistema dovrebbero essere visibili o reperibili con facilità ogni volta che è necessario senza che l'utente debba ricorrere alla sua memoria.

- Consenti alle persone di riconoscere le informazioni nell'interfaccia, piuttosto che costringerle a ricordarle ("richiamarle").
- Offri aiuto nel contesto, invece di fornire agli utenti un lungo tutorial.
- Ridurre le informazioni che gli utenti devono ricordare.

7. Flessibilità ed efficienza d'uso



I tasti rapidi - non visibili da utenti principianti - spesso possono velocizzare l'interazione per utenti esperti, è utile far sí che il sistema sia adatto ad entrambi.

- Fornire acceleratori come scorciatoie da tastiera e/o gesti tattili.
- Fornire la possibilità di personalizzare contenuti e funzionalità per i singoli utenti.
- Consentire la personalizzazione, in modo che gli utenti possano effettuare selezioni su come desiderano che il prodotto funzioni.

8. Design estetico e minimalista



Le schermate non dovrebbero contenere informazioni irrilevanti o usate raramente. Ogni componente informativo in una schermata compete con gli altri, e diminuisce la loro visibilità relativa.

- Mantieni il contenuto e il design visivo dell'interfaccia utente focalizzati sui contenuti essenziali.
- Non lasciare che elementi non necessari distraggano gli utenti dalle informazioni di cui hanno veramente bisogno.
- Dai la priorità ai contenuti e alle funzionalità per supportare gli obiettivi primari.

10. Aiuto e documentazione



Può essere necessario fornire degli aiuti e della documentazione di riferimento. Questo tipo di informazioni devono essere semplici da ricercare e da utilizzare.

- Assicurarsi che la documentazione di aiuto sia facile da trovare.
- Quando possibile, presentare la documentazione nel contesto giusto, cioè nel momento in cui l'utente potrebbe averne bisogno.
- All'interno della documentazione elencare i passi concreti da compiere.

Esercizio 9

Esercizio usabilità

Valuta seguendo il PDF con le 10 euristiche i seguenti siti:

- <https://www.abacatania.it/>
- <https://www.regione.sicilia.it/>

Accessibilità

I principi di accessibilità - w3.org

— — —

- **Percepibile**
- **Utilizzabile**
- **Comprensibile**
- **Robusto**

Le aree di necessità che dobbiamo tenere in considerazione sono:

- Visuale
- Motoria
- Uditiva
- Convulsioni (alcuni individui possono essere colpiti da luce, movimento, sfarfallio, ...)
- Apprendimento (l'apprendimento e le disabilità cognitive possono influenzare l'accessibilità)

Percepibile

Le informazioni e i componenti dell'interfaccia utente devono essere presentati agli utenti in modi in cui essi possano percepirli attraverso la vista, l'udito e/o il tatto.

- Alternative testuali
- Media temporizzati
- Adattabile
- Distinguibile

Utilizzabile

I componenti e la navigazione dell'interfaccia utente devono essere utilizzabili attraverso differenti modalità di input.

- Accessibile da tastiera
- Adeguata disponibilità di tempo
- Convulsioni e reazioni fisiche
- Navigabile
- Modalità di input

Comprensibile

Le informazioni e le operazioni dell'interfaccia utente devono essere leggibili e comprensibili con etichette chiare e istruzioni.

- Leggibile
- Prevedibile
- Assistenza nell'inserimento

Robusto

Il contenuto deve essere abbastanza robusto per essere interpretato in maniera affidabile da una grande varietà di browser e dispositivi, comprese le tecnologie assistive.

- Compatibile

Accessibilità e Design Inclusivo

Ci sono molti modi per rendere accessibile un sito web.

Per iniziare, ecco alcuni elementi a cui fare attenzione durante la progettazione:

- Header Tags
- Link corretti
- Scelta dei colori e riferimento
- Contenuti

Header tags

Progettare e utilizzare correttamente le intestazioni (da H1 a H6). Utilizzare H1 solo una volta per pagina.

Gli utenti con disabilità visive più gravi possono accedere al tuo sito utilizzando un display o un terminale Braille aggiornabile, che dipende dagli screen reader.

Link corretti

Progettare correttamente i link. Evitare i link "clicca qui" sparsi all'interno del testo.

I link migliori hanno una **descrizione di testo** prima del collegamento e quindi un nome univoco per il link. Prendi in considerazione l'idea di offrire un segnale visivo (come un'icona PDF) sul link per chiarire cosa offrirà.

Usa le **sottolineature sui link** all'interno del testo (aiutano le persone daltoniche a distinguerli).

Evidenzia i collegamenti del menu al passaggio del mouse (**effetto hover**) per aiutare a localizzare il cursore.

Scelta dei colori e riferimento

Attenzione alla **scelta dei colori**. Il daltonismo è una disabilità incredibilmente comune e la tavolozza sbagliata può rendere difficile per una persona daltonica leggere il testo o navigare il sito.

È necessario assicurarsi di fornire **alti livelli di contrasto** tra testo e sfondo, anziani o ipovedenti possono avere difficoltà a vedere il testo a meno che il contrasto non sia elevato.

Non riferirsi solo al colore per eventuali istruzioni (es. "Fai clic sul pulsante rosso" non è utile per una persona daltonica). Le forme possono aiutare a guidare gli utenti piuttosto che fare affidamento solo sul colore.

Contenuti

Attenzione ai contenuti dinamici. Non riprodurre automaticamente i video (può essere problematico con lo screen reader).

Offrire **trascrizioni per file audio**. Gli utenti con problemi di udito non possono utilizzare il software per leggere le voci quindi è utile includere una trascrizione. Allo stesso modo, nei video, offrire didascalie per i non udenti.

Creazione di **contenuti leggibili**. Più semplice è il linguaggio, più facile sarà la lettura per gli utenti con difficoltà di apprendimento.

Primo elaborato per esame:

Portfolio personale

Portfolio_Personale.pdf