



Finanziato
dall'Unione europea
NextGenerationEU



*Ministero dell'Istruzione
e del Merito*



Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



SOFTWARE DEVELOPER

Fondamenti di UX/UI Design e HTML CSS

Docente: Loredana Frontino

Titolo argomento: Responsive e media query

Responsive

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Cos'è il responsive design

È un approccio che permette di visualizzare le pagine web su tutte le tipologie di dispositivo, rispetto a dimensione e risoluzione dello schermo, garantendo una buona usabilità.

Il responsive web design è un approccio alla progettazione che tiene conto della gamma di dispositivi e delle loro dimensioni, consentendo l'adattamento automatico allo schermo, indipendentemente dal fatto che il contenuto venga visualizzato su un tablet, un telefono, un televisore o un orologio.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Cos'è il responsive design

Il termine responsive design, descrive l'uso di griglie fluide, immagini fluide e query multimediali per creare contenuti reattivi.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Adaptive design

Approccio differente di gestione dell'adattamento di un layout all'interfaccia.

In questo caso non si creano elementi fluidi ma vengono creati direttamente dei layout differenti in base ai dispositivi.



Responsive design

Come creare siti responsive?

- a) Utilizzando unità di misura relative quando serve
- b) Creando layout fluidi
- c) Sfruttando le potenzialità di flexbox o grid
- d) Utilizzando le media query

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Inserire il metatag viewport

Viewport

Attraverso il metatag viewport, da inserire nell'head della pagina HTML, è possibile andare a definire come questa si comporta in relazione alla variazione dell'area della pagina



```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Dimensione minima
della viewport

Il livello di zoom della
pagina al caricamento
iniziale

Unità di misura relative

Assolute

px

pt

Relative

em

rem

%

vh-vw

Assolute

px

pt

Unità di misura scalabile e definita a partire dall'attuale font-size impostata nell'elemento padre.

Se il font è 16px , 1em corrisponderà a 16px

Relative

em

rem

%

vh-vw

Utilizzati per definire una gerarchia di dimensioni all'interno di un elemento.

Utile per il responsive, ma con possibilità di essere variabile nel passaggio tra un elemento e l'altro.

Assolute

px

pt

Unità di misura relativa e definita a partire dall'attuale font-size impostata a livello di progetto, nell'elemento root della pagina. Se il font è 16px , 1rem corrisponderà a 16px

Relative

em

rem

%

vh-vw

Basandosi sull'unità di misura stabilita come proprietà generale, è utile considerarla per la gestione del responsive, se viene impostata la variazione del parametro al cambio del breakpoint.

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Assolute

px

pt

Unità di misura relativa che va a calcolare la width in percentuale rispetto alla dimensione dell'elemento padre che la contiene.

Relative

em

rem

%

vh-vw

Utile per dividere gli spazi all'interno di un'area, infatti viene utilizzata per la definizione delle colonne in bootstrap.

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Assolute

px

pt

Sono acronimi rispettivamente di “viewport height” e “viewport width”.

1vh rappresenta 1% dell'altezza della finestra del browser e 1vw l'1% della larghezza della finestra del browser.

Relative

em

rem

%

vh-vw

Utile quando si stanno gestendo posizioni rispetto allo schermo, larghezze e dimensioni rispetto alla viewport. Da prendere in considerazione quando si vuole definire un parametro che occupa un certo spazio in altezza.

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Layout fluidi

Utilizzando le unità di misura prima descritte andare a definire dei layout che si adattino alla dimensione del contenuto della pagina.

Inoltre è utile anche andare a settare i parametri come min-width e max-width per definire le dimensioni massime e minime che si vogliono dare agli elementi (queste preferibilmente con unità di misura assolute)

Fluid layout

width: 63%

width: 32%

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

```
<div class="flex-container">  
  <div class="child-box">  
    <p>A</p>  
  </div>  
  <div class="child-box">  
    <p>B</p>  
  </div>  
  <div class="child-box">  
    <p>C</p>  
  </div>  
</div>
```

```
.flex-container {  
  display: flex;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà base:

flex-direction

justify-content

align-items

flex-wrap

```
.flex-container {  
  display: flex;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà base:

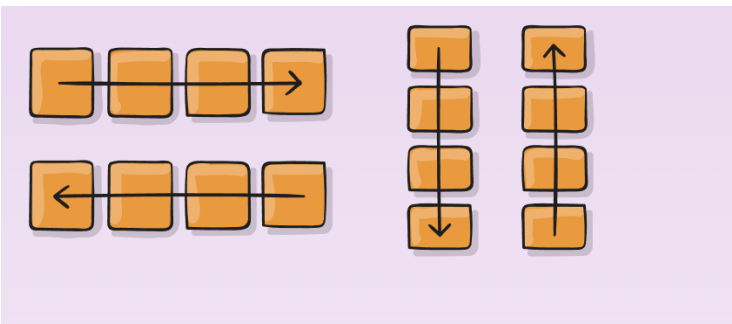
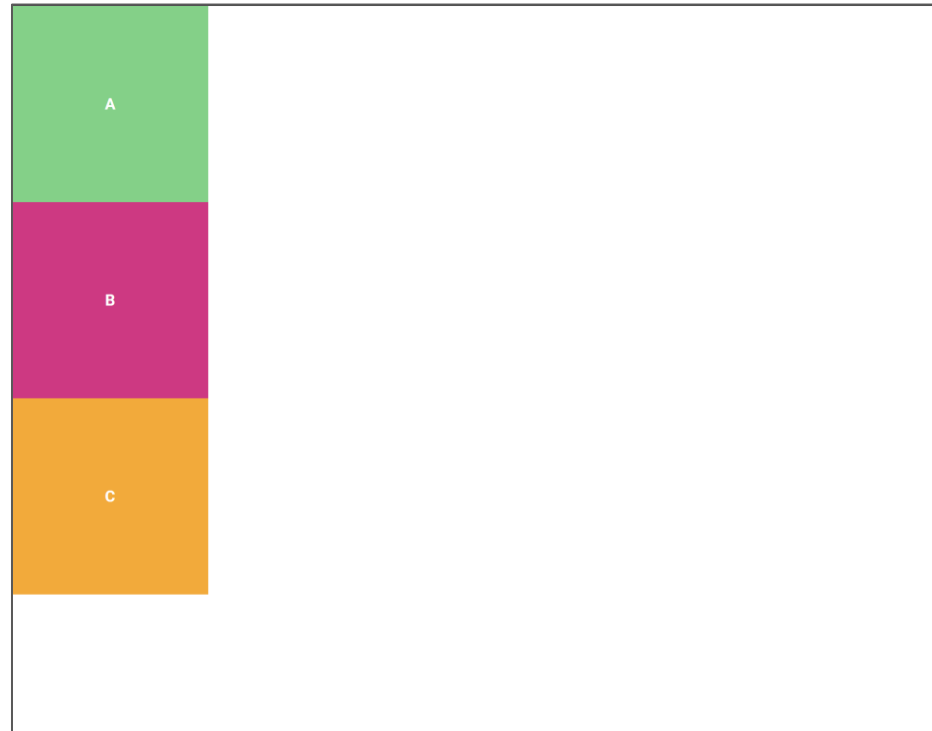
flex-direction

justify-content

align-items

flex-wrap

```
.flex-container {  
  display: flex;  
  flex-direction: column;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà base:

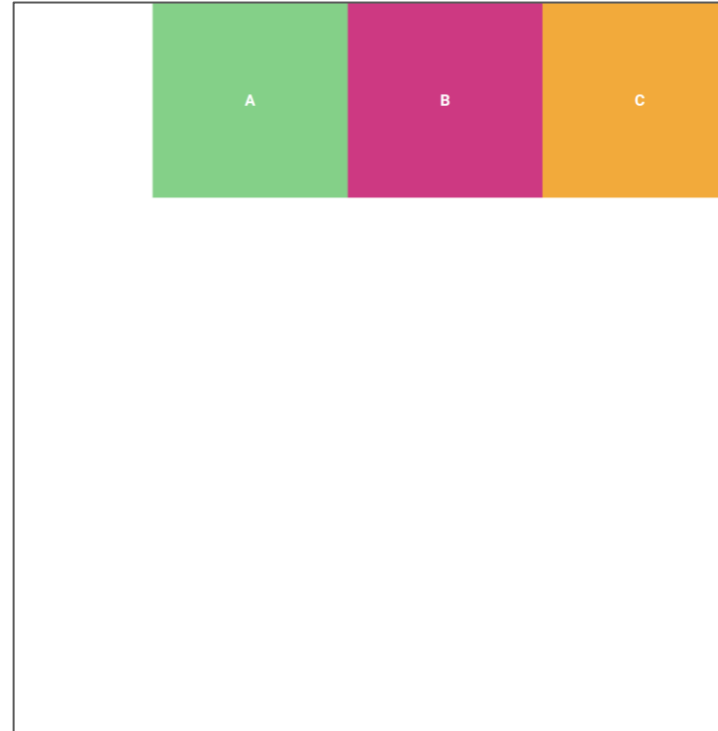
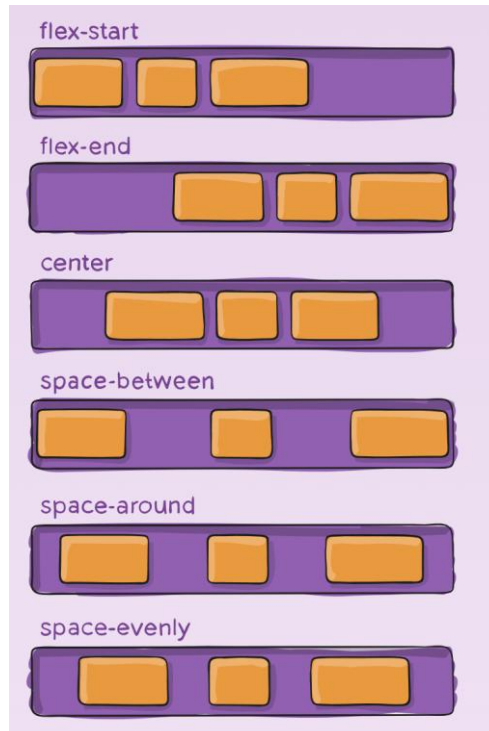
flex-direction

justify-content

align-items

flex-wrap

```
.flex-container {  
  display: flex;  
  justify-content: end;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà base:

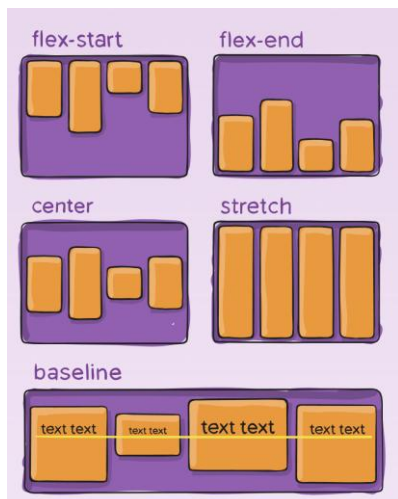
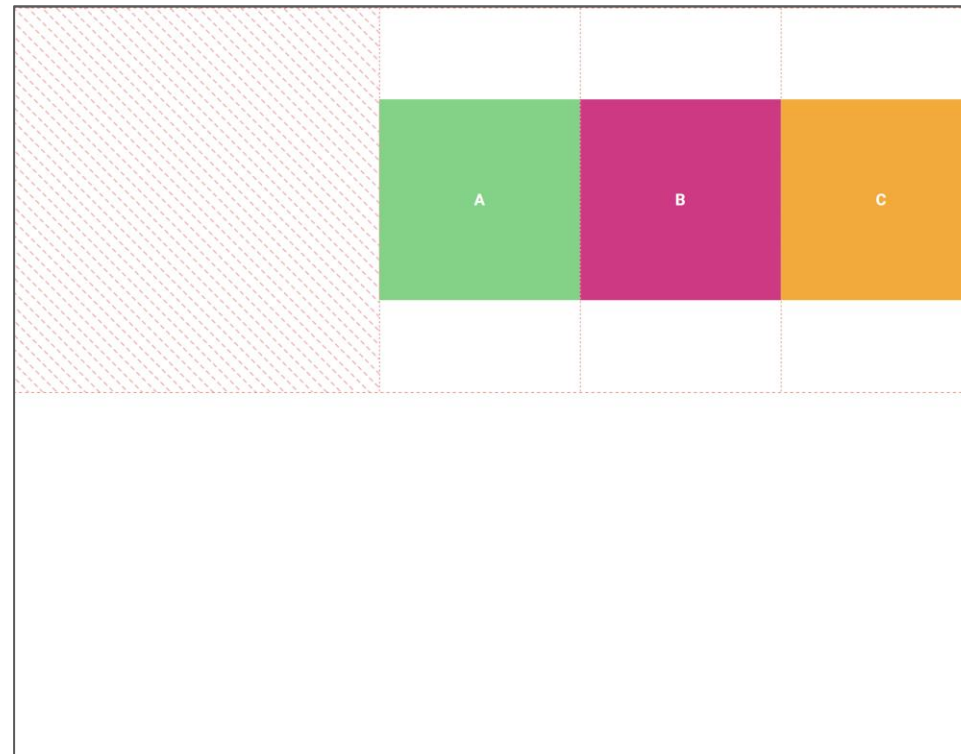
flex-direction

justify-content

align-items

flex-wrap

```
.flex-container {  
  display: flex;  
  align-items: center;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà base:

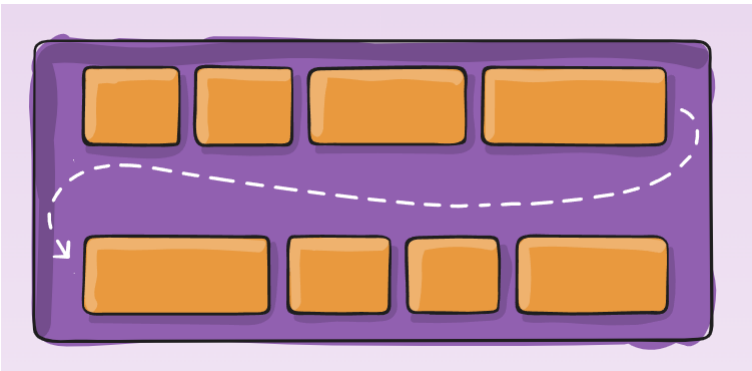
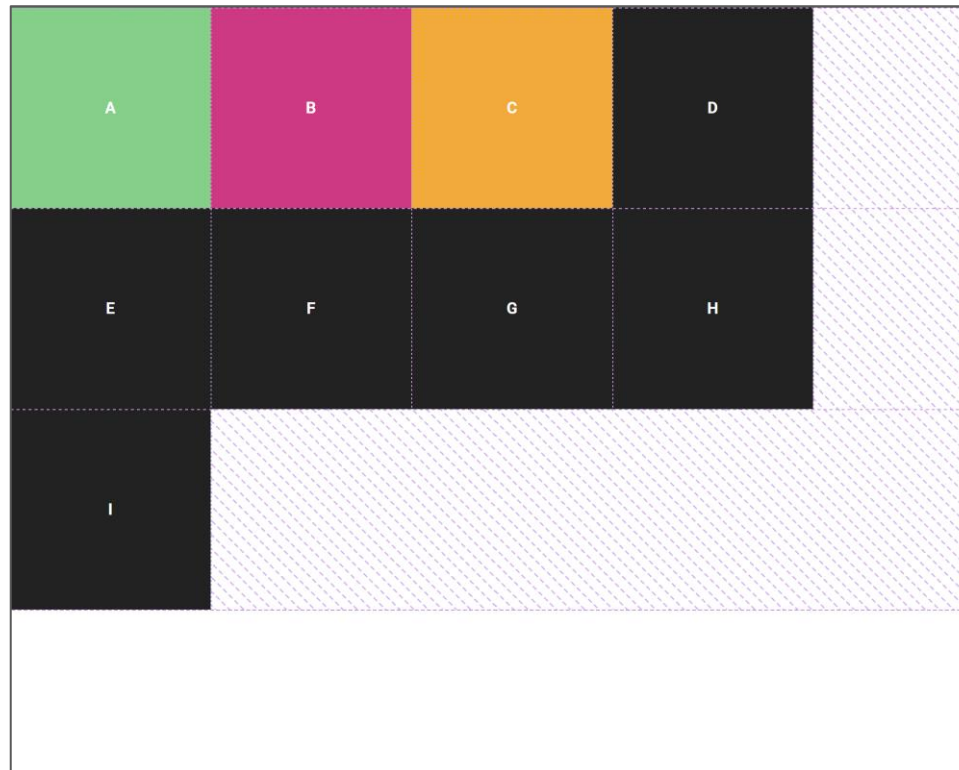
flex-direction

justify-content

align-items

flex-wrap

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
}
```



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Display flex

Proprietà dei figli

order

flex-grow

flex-shrink

flex-basis

align-self

```
.child-flex:first-child {  
  order: 1;  
  flex-grow: 3;  
  flex-shrink: 3;  
  flex-basis: auto;  
  align-self: end;  
}
```

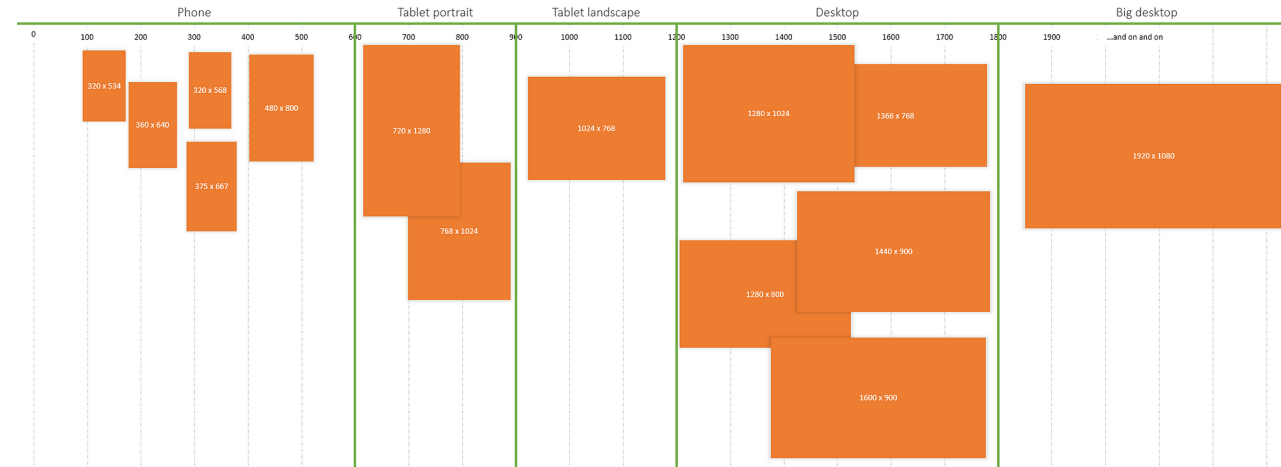
Breakpoints

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Cosa sono i breakpoints

Sono punti specifici di larghezza dello schermo (o a volte altezza) in corrispondenza dei quali il layout e lo stile del tuo sito web cambiano per adattarsi meglio a diverse dimensioni di dispositivi (desktop, tablet, smartphone, ecc.).

In pratica, sono delle condizioni che attivano regole CSS diverse. Ad esempio, potresti avere un layout a tre colonne per schermi larghi e passare a un layout a una colonna per schermi più stretti, utilizzando un breakpoint per definire quando questa transizione deve avvenire.

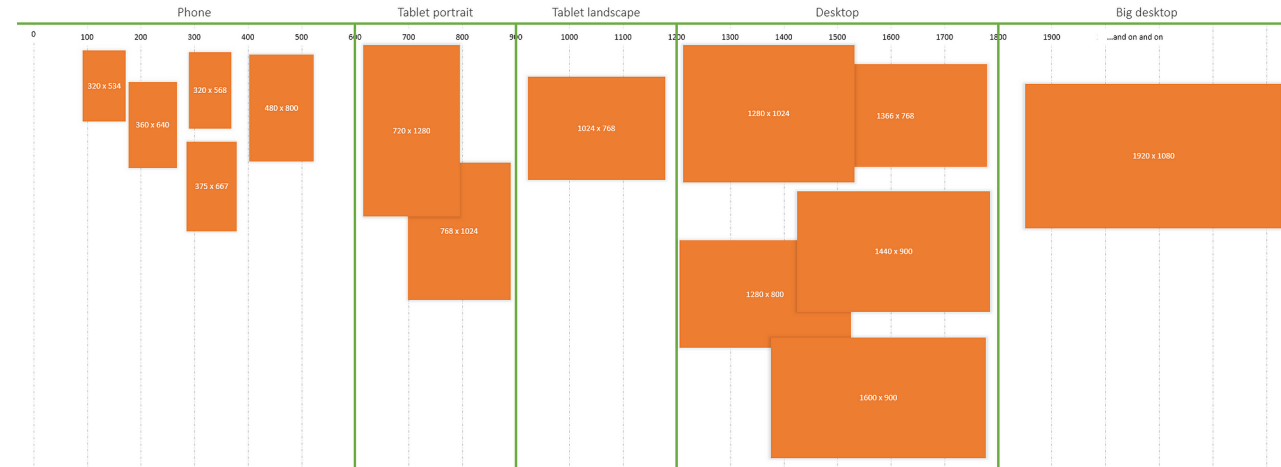


Fondamenti di UX/UI design e HTML CSS – Modulo 14

Cosa sono i breakpoints

Sono quindi soglie di dimensioni dello schermo che innescano modifiche nel codice CSS per rendere un sito web responsive.

Per implementarli si utilizzano le **media query CSS** di tipo **viewport**.

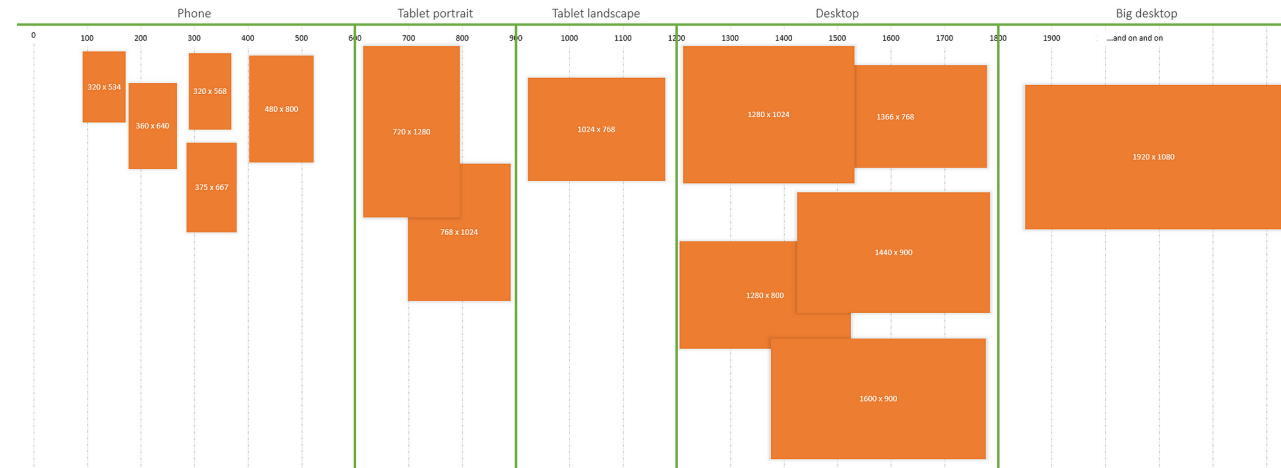


Fondamenti di UX/UI design e HTML CSS – Modulo 14

Breakpoints come definirli

Solitamente vengono definiti in accordo con il team di design.

Non esiste uno standard definito per le larghezze da scegliere come target nelle media query, l'obiettivo è avere breakpoints sufficienti per indirizzare smartphone, tablet, laptop e desktop.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Breakpoints comuni

320px

480px

576px

768px

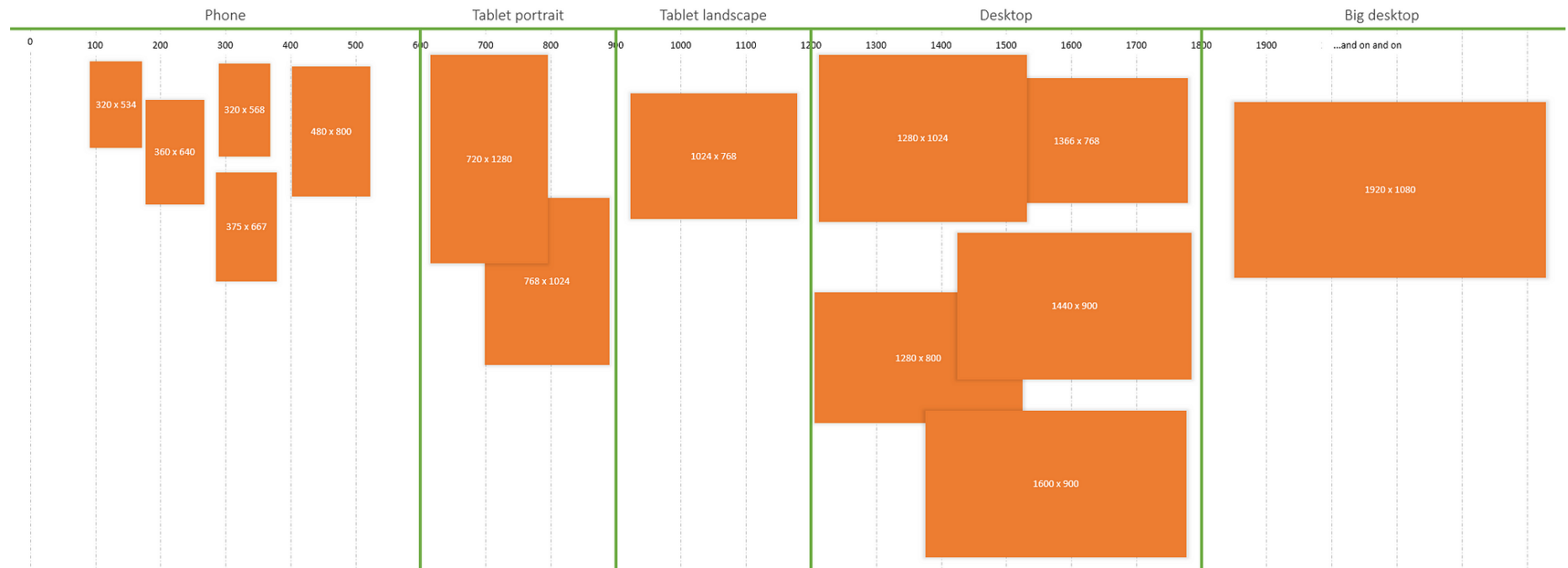
992px

1024px

1200px

1440px

1920px



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Esempio di breakpoints di Bootstrap

```
/* Extra small devices (portrait phones, less than 576px) */  
/* No media query for `xs` since this is the default in Bootstrap */  
/* Small devices (landscape phones, 576px and up) */  
@media (min-width: 576px) {}  
  
/* Medium devices (tablets, 768px and up) */  
@media (min-width: 768px) {}  
  
/* Large devices (desktops, 992px and up) */  
@media (min-width: 992px) {}  
  
/* Extra large devices (large desktops, 1200px and up) */  
@media (min-width: 1200px) {}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Logiche di organizzazione

Ricordare sempre lo stile a cascata, quindi le proprietà impostate per ultime nel file avranno la priorità.

Cercare di non aggiungere troppi breakpoints, che possono complicare inutilmente il codice e rendere difficile la manutenzione. Concentrarsi su quelli essenziali che rispondono meglio ai dispositivi più comuni utilizzati dagli utenti target.

Evitare duplicazione e ripetizione di media query.

```
@media (min-width: 576px) {}
```

```
/* Medium devices (tablets, 768px and up) */
```

```
@media (min-width: 768px) {}
```

```
/* Large devices (desktops, 992px and up) */
```

```
@media (min-width: 992px) {}
```

```
/* Extra large devices (large desktops, 1200px and up) */
```

```
@media (min-width: 1200px) {}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Diversi browser e sistemi operativi

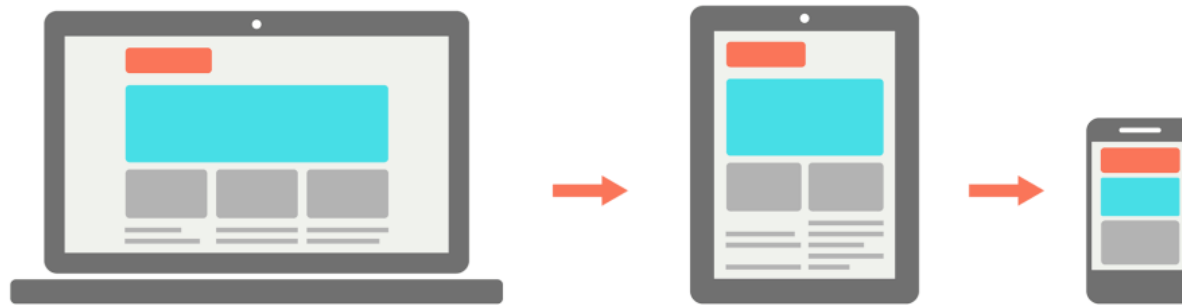
Fare attenzione quando si sviluppa al comportamento che possono avere i differenti browser alle proprietà che vengono definite.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Logiche di organizzazione

Mobile first o content first?



Responsive Web Design

Mobile First Web Design



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Logiche di organizzazione

Mobile first?

Il design mobile-first implica la creazione di stili predefiniti per i dispositivi mobili, e l'aggiunta di media query per adattare il layout ai dispositivi più grandi.

A livello di design questo approccio permette di organizzare e definire per primi gli elementi importanti di un sito, agendo dove le risorse e lo spazio sono limitati.

```
@media (min-width: 576px) {}
```

```
/* Medium devices (tablets, 768px and up) */
```

```
@media (min-width: 768px) {}
```

```
/* Large devices (desktops, 992px and up) */
```

```
@media (min-width: 992px) {}
```

```
/* Extra large devices (large desktops, 1200px and up) */
```

```
@media (min-width: 1200px) {}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Logiche di organizzazione

Content first?

Si dà priorità alla facilità di lettura del contenuto e all'accessibilità da tutte le dimensioni di schermo.

Secondo questo approccio il layout dovrebbe adattarsi al contenuto e non il contrario.

```
/* Extra large devices (large desktops, 1200px and down) */
```

```
@media (max-width: 1200px) {}
```

```
/* Large devices (desktops, 992px and down) */
```

```
@media (max-width: 992px) {}
```

```
/* Medium devices (tablets, 768px and down) */
```

```
@media (max-width: 768px) {}
```

```
/* Medium devices (mobile, 576px and down) */
```

```
@media (max-width: 576px) {}
```

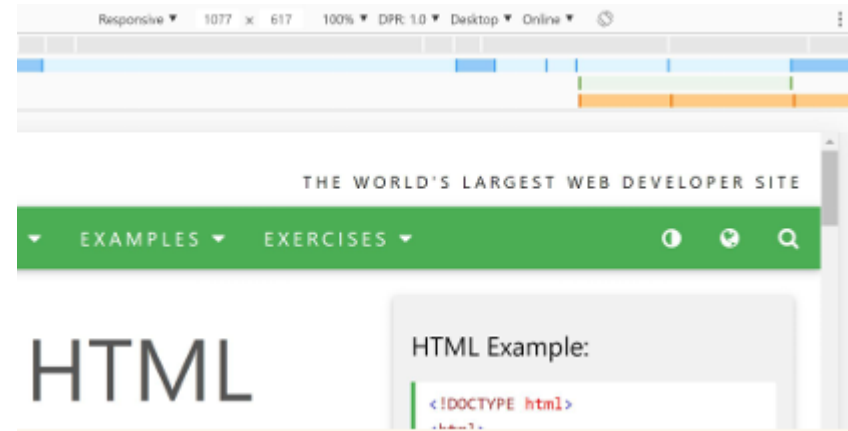

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Debugging con i breakpoints

Gli strumenti di sviluppo del browser hanno il supporto integrato per i breakpoints delle media query.

È facile vedere quali breakpoints sono definiti nei fogli di stile e testare la pagina.

Consentono anche di effettuare una simulazione dei dispositivi di destinazione, però è solo una simulazione, in quanto i test conviene farli sui dispositivi reali.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Esercitazioni

Esercizio 1

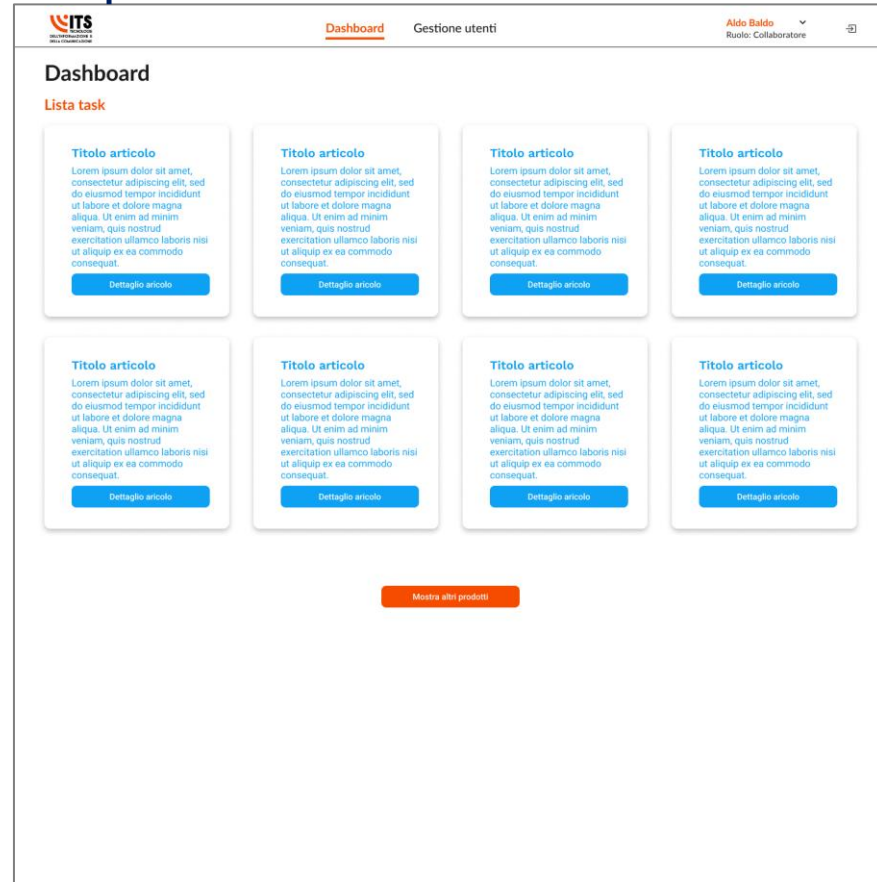
Creare un nuovo file HTML con un CSS esterno associato

- Inserire un header e dei blocchi di contenuto disposti a colonne all'interno della pagina riprendendo la struttura responsive gestita a pagina seguenete, creando anche delle card con contenuto a piacere all'interno ma che abbiano un box shadow e border radius definiti.
- Poi gestire le variazioni al responsive aggiungendo i breakpoint necessari.

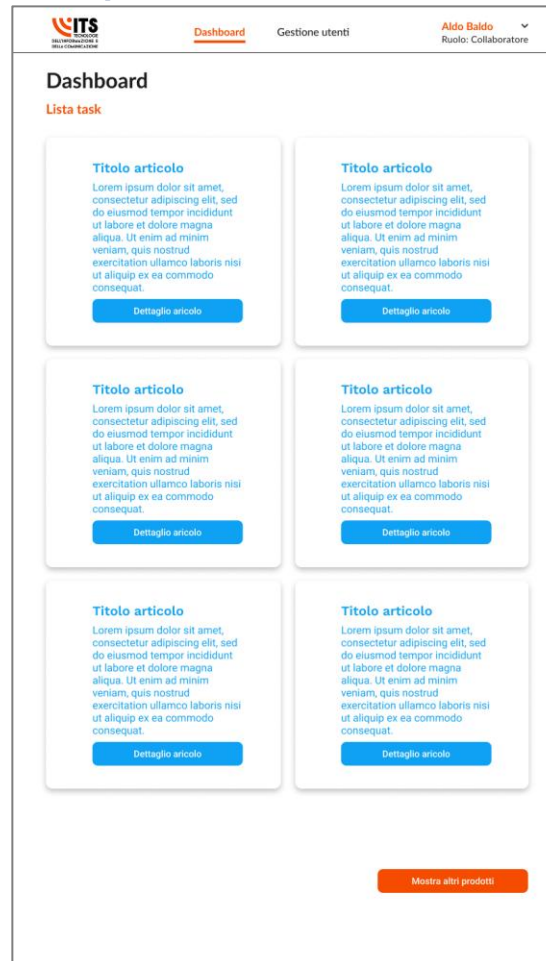
Fondamenti di UX/UI design e HTML CSS – Modulo 14

Esercizio 1

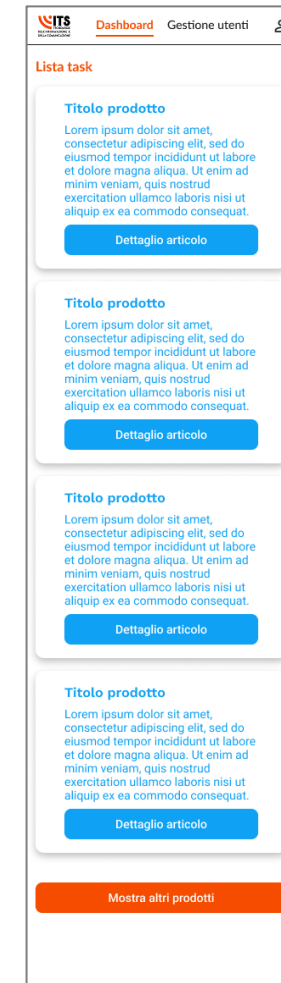
1440px



1024px



320px



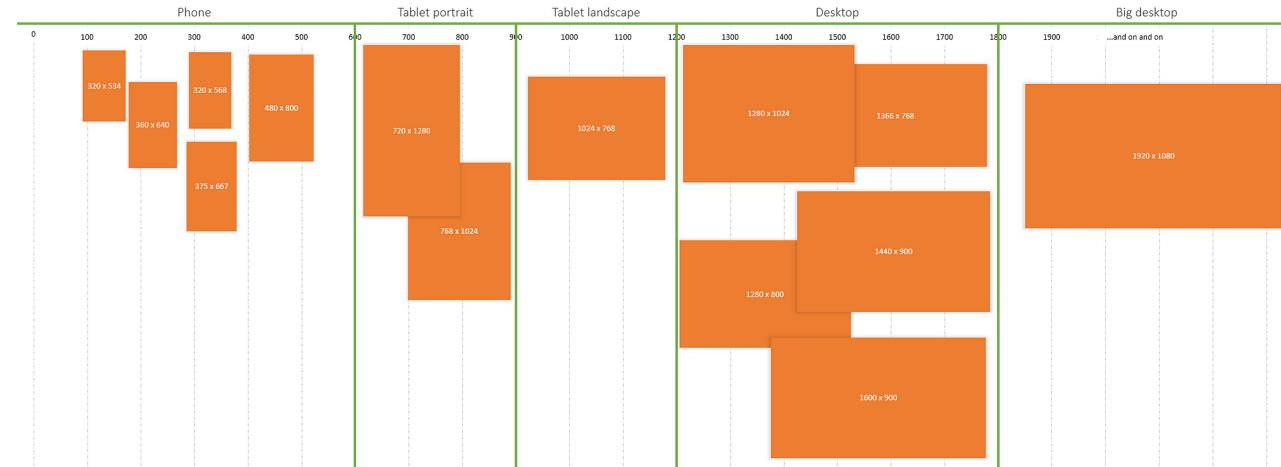
Media queries

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Media query

Sono delle espressioni logiche che vengono applicate al CSS, per permettere di personalizzare le pagine web per dispositivi specifici (telefoni, tablet, desktop, etc.).

Se una condizione espressa da una media query è vera, allora le relative regole verranno applicate.



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Media query - elementi

Media type

all | screen | print | speech

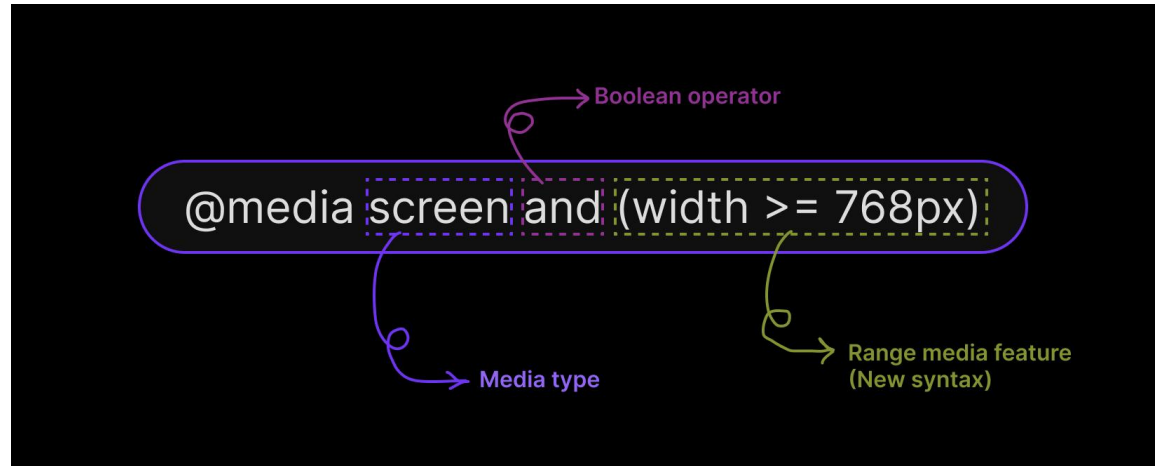
Media features

Width | min-width | max-width | orientation | height |
min-height | max-height | ...

Le più usate sono min-width | max-width | orientation

Logical operators

and | or (rappresentato anche con ,) | not



Fondamenti di UX/UI design e HTML CSS – Modulo 14

Esempi di media queries

```
/* (landscape phones, 576px and up) */
```

```
@media screen and (min-width: 576px) {  
  span {  
    color: #ff0000;  
  }  
}
```

```
/* (tablets, 768px and up and orientation landscape) */
```

```
@media screen and (min-width: 768px) and (orientation: landscape){}
```

```
/* (tablets, 768px and up or orientation landscape) */
```

```
@media screen and (min-width: 768px), (orientation: landscape){}
```

```
/* (theme dark) */
```

```
@media screen and (prefers-color-scheme: dark) {}
```

Fondamenti di UX/UI design e HTML CSS – Modulo 14

Media query – prefers-color-schema

Se si vuole applicare un tema e modificare i colori a un progetto impostato con variabili di root, basterà definire le variabili nel tema di default (light) e successivamente aggiungere una media query prefers-color-scheme: dark per settare i parametri di root che cambiano

```
:root {  
  color: blue  
}
```

```
@media (prefers-color-scheme: dark) {  
  :root {  
    color: purple  
  }  
}
```


Fondamenti di UX/UI design e HTML CSS – Modulo 14

Esercitazioni

Esercizio 2

Utilizzando l'HTML e il CSS dell'esercizio precedente:

- Gestire la combinazione colori per la modalità dark, andando ad impostare delle variabili nella root che poi cambiano quando le impostazioni del browser vengono settate sul tema dark.
- Trovare la proprietà nell'inspect di chrome che permette di effettuare questa operazione.

Link utili

[Responsive design](#)

[Tecniche di responsive](#)

[Viewport](#)

[Responsive o mobile first?](#)

[Media queries list](#)

[Prefers-color-scheme](#)

[Responsive images](#)

[Esempi di responsive web sites](#)