COESIONE ITALIA 21-27 PIEMONTE
Cofinanziato dall'Unione europea
REGIONE PIEMONTE

Web Developer

HTML, CSS e Strumenti di Digital Marketing
(SEO, SEM, SEA)

Docente: Shadi Lahham

Intervento realizzato da
ITS
TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Responsive web pages

Media queries

Shadi Lahham - Web development

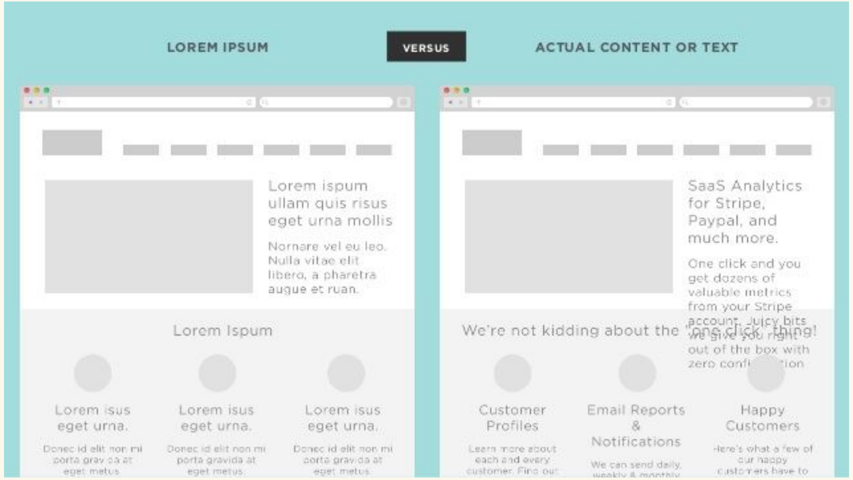
Content first

What is content

- Content is the reason users
    - Visit website
    - Downloads app
    - Provides email address
- Content encompasses a variety of media
    - Graphics, video, audio,
    - Social media communication
    - Anything used to tell a story or communicate an idea
- Content is storytelling
    - A user-friendly digital framework enables users to make a series of choices that helps them tell their story

Why content-first

- A non content design approach
  - Risks creating useless templates
  - Requires more reworks
  - Increases project costs
  - Miscommunications with clients
  - Impacts the User Experience


LOREM IPSUM
VERSUS
ACTUAL CONTENT OR TEXT
Lorem ispum ullam quis risus eget urna mollis
Nornare vel eu leo. Nulla vitae elit libero, a pharetra augue et ruan.
Lorem Ispum
Lorem isus eget urna.
SaaS Analytics for Stripe, Paypal, and much more.
One click and you get dozens of valuable metrics from your Stripe account. Juicy bits we give you right out of the box with zero configuration
We're not kidding about the "one click" thing!
Customer Profiles
Email Reports & Notifications
Happy Customers

How to work content-first

1. Perfect the user experience
    - Think about the whole user experience rather than content in terms of individual pages.
    - Considering the content early the process can yield a better experience
2. Consider all channels upfront
    - Aim to have unified and consistent content across all channels & devices
    - Helps to spot opportunities and problems early on
3. Use content to define the layout and design
    - Understand how content can inform design
    - Don't spend time creating templates you won’t use
    - Focus on problem-solving for your users

How to work content-first

4. Use proto-content
   - Focus on getting the writing done
   - Work with a prototype
   - Use proto-content instead of non-contextual placeholders such as Lorem Ipsum
   - Use existing content, draft content, or sample content
5. Understand content and technological requirements early
   - Start discussions early across different disciplines: content, UX, design and development
   - Have discussions around the required technology to deliver content
   - Modify the development platform to accommodate the content or design the content with the limitations of the development platform in mind

Mobile first

Why mobile-first

- Mobile phones are an integral part of our lives
- More than 50% of all global web traffic comes from mobile phones
- Google looks at the mobile version before the desktop when ranking a site
- Users have a higher trust in websites that have an excellent mobile UX
- Easier to progress from more straightforward outline and functionality to complex solutions
- Mobile-first means being modern, useful and straightforward

How to work mobile-first

- Mobile-first coding
  - If you use a framework, make sure it's mobile-first
  - If you write custom code, always prioritize mobile
  - Code elements as mobile-first
- Intuitive and user-friendly interactions
  - Clear CTAs (Call to Action)
  - Super quick loading
- Optimise your content (mobile first usually implies content-first)
  - Relevant, easy to read quickly content (divide long text)
  - Scroll rather than click
  - Help users find what they are looking for quickly
  - Use fonts that display well on mobile

Responsive design

What is responsive design

- A website or application that automatically adjusts to the screen and adapts to any device
- Responsiveness is a feature of a web page
  - Is an outcome of specific web development techniques
  - Usually implies mobile-first but does not require it
- Achieved by deploying media queries that
  - Change the default CSS styles
  - Modify the layout

Responsive design and development

- Designers
  - Used to start from the desktop version
  - The mobile version was an afterthought
  - Today it's common to design for mobile-first
  - Design the desktop version by scaling up and adding features
- Developers
  - Used to develop for the desktop first
  - Mobile development is more painful and requires more knowledge, more testing and more creativity in problem solving
- Teams
  - More collaboration between multiple disciplines: Copy, UI, UX, developers
  - More coordination, adaptation and iteration

Unified strategy

Design & development strategy

i 3 punti principali per il processo di desgin e realizzazione di una pagina web

Unified design & development

1. Content First

- Identify and prioritize essential content, including "proto-content"
    - early, draft content used during design and development
- Structure content to highlight key information and actions
- Ensure clarity and relevance in messaging

2. Mobile First

- Design for mobile, ensuring core features and content are accessible
- Use a simplified, minimalistic approach to design and functionality
- Implement progressive enhancement: add more features for larger screens

Unified design & development

3.Responsive Design

- Develop a flexible layout using grids, flexible images, and media queries
- Ensure design adapts across various screen sizes and orientations
- Test across devices to maintain a consistent user experience

Unified design & development

Implementation Steps

- Content Audit
  - Identify critical content and create "proto-content" for early design stages
- Wireframing
  - Create mobile-first wireframes focusing on essential content and "proto-content"
- Design
  - Develop a responsive design that scales from mobile to desktop
- Development
  - Use responsive techniques to build adaptable layouts
- Testing
  - Conduct testing on multiple devices to ensure functionality and consistency

Unified design & development

Benefits

- Enhanced user experience across all devices
- Optimized performance and accessibility
- Efficient maintenance and scalability

Breakpoints

Common breakpoints

- Breakpoints are often defined in collaboration with UI/UX designers
- There are no defined standard for widths to target in media queries
- Any reasonable set of increments is enough to target most devices
- The aim is to have sufficient breakpoints to target smartphones, tablets, laptops, and desktops

Some of the most common widths used:

```
320px
480px
576px
768px
992px
1024px
1200px
```

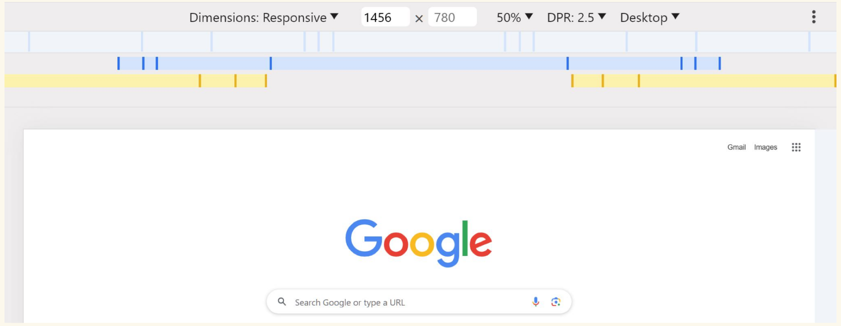
Example: Bootstrap breakpoints

```
/* 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) {}
```

Bootstrap is a front-end framework, shown here just as an example of breakpoint usage

Devtools & breakpoints

- Browser devtools have built-in support for media query breakpoints
- Easy to see which breakpoints are defined in the stylesheets and to test the page
- Also allow to simulate target devices
  - this is only an approximation
  - test on actual devices



[Simulate mobile devices]

Viewport meta tag

Viewport meta tag

- Located in the <head> of the HTML document
- Defines how a site should render in a web browser for mobile devices
- Makes media queries will work as intended
- Should always include the following line of HTML in the head of all documents

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

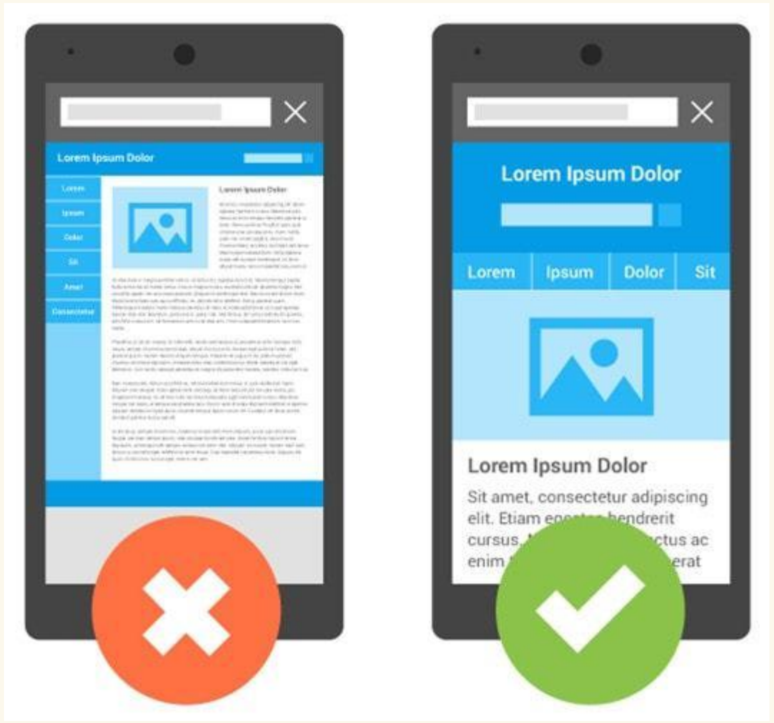
device-width
indicates the width should match with the viewport of the device

initial-scale
ensure the zoom will not be applied and the layout will always show on a 1:1 scale

Other settings can be used in the viewport tag but it is **not recommended** to change them

More details
Responsive Web Design Viewport

Viewport meta tag



Media queries

Media queries

- Allow the customization of web pages for specific devices
    - mobile phones, tablets, desktops, etc
- The HTML code is not changed, only the CSS style
- A media query is a logical expression: true or false
- If a media query is true, the related rules are applied to the target device

Media queries

```
@media screen and (min-width: 480px) {
 div {
   float: left;
   background: red;
 }
}
```

Media queries

```
@media screen and (min-width: 480px) {
 div {
   float: left;
   background: red;
 }

 div > p {
  margin: 50px;
  padding: 10px;
  border: 1px dashed black;
 }
}

div {
  border: 2px solid green;
}
```

Media queries - breakdown

`Media type` indichiamo a che tipologia di media andiamo ad applicare le regole

`all, screen, print`

`Media features` indichiamo la condizione per far si che le regole siano applicate all'interno di quella media query

`width, min-width, max-width, orientation, height, min-height, max-height, etc`

`There are many, but most are not used`

`The most used ones are min-width, max-width and orientation`

`Logical operators` possiamo usare operatori logici all'interno delle condizioni (and ,)...

`Logically represent and, or, not`

`The 'or' logic is represented by a comma (just like in CSS selectors)`

Media queries - media type

```css
/* sets a light blue background for the body element when viewed on screens */
@media screen {
  body {
    font-size: 16px;
    background-color: lightblue;
  }
}

/* sets the font size and color for printing, ensuring readability on paper */
@media print {
  body {
    font-size: 12pt;
    color: black;
    background-color: white; /* changing the background color to white for printing */
  }
}
```

Media queries - logical operators

AND logic: Both conditions must be true

```
@media screen and (min-width: 400px) and (orientation: landscape) {
 body {
   color: #31c78d;
 }
}
```

OR logic: At least one conditions should be true

```
@media screen and (min-width: 400px), screen and (orientation: landscape) {
 body {
   color: #00c3ff;
 }
}
```

Media queries - logical operators

Not logic: The 'not' operator negates the entire condition

```
@media not all and (orientation: landscape) {
 body {
   color: #dab928;
 }
}
```

Media queries - columns example

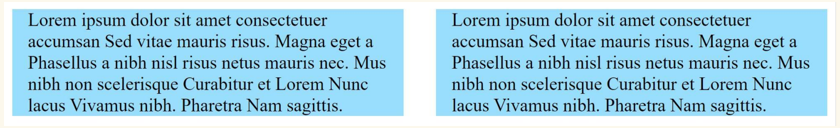
```css
.column {
 width: 48%;
 padding: 0 15px;
 box-sizing: border-box;
 background: #97ddff;
 float: left;
 font-size: 1.1em;
}

.container .column:first-child {
 margin-right: 4%;
}
```

HTML:

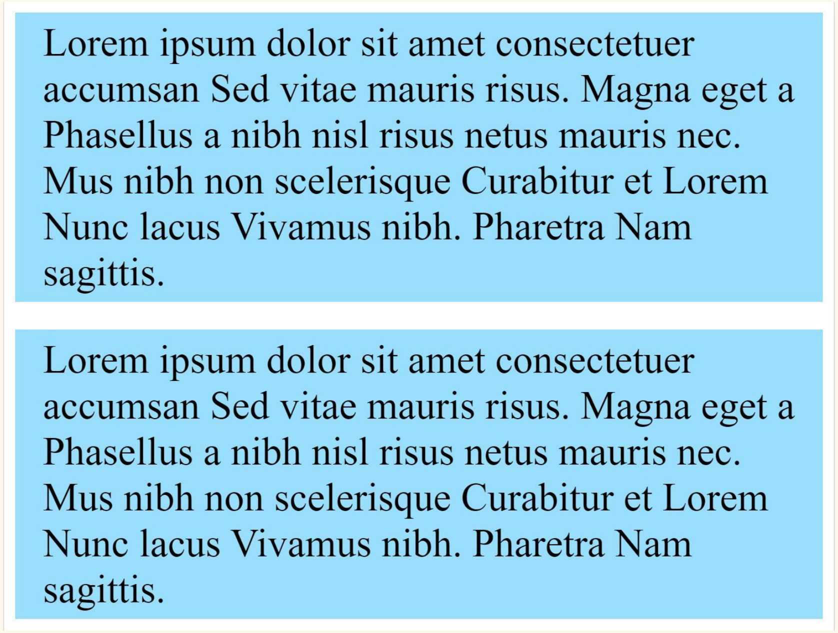
```html
<div class="container">
    <div class="column">Lorem ipsum ..</div>
    <div class="column">Lorem ipsum ..</div>
</div>
```

```css
@media screen and (max-width: 767px) {

  .column {
    width: 100%;
    padding: 5px 20px;
    float: none;
    font-size: 1.8em;
  }

  .container .column:first-child {
    margin-right: 0;
    margin-bottom: 20px;
  }

}
```

Media Queries - columns example


Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.


Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Lorem ipsum dolor sit amet consectetuer accumsan Sed vitae mauris risus. Magna eget a Phasellus a nibh nisl risus netus mauris nec. Mus nibh non scelerisque Curabitur et Lorem Nunc lacus Vivamus nibh. Pharetra Nam sagittis.

Media attribute

- It's possible to specify a media attribute in the link element
- This applies a whole stylesheet when the condition is true
- Possible to do but preferable to specify a single stylesheet with individual media queries

Example

```
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">
```

- It's also possible to have different stylesheets based on media type
- This might make sense for some use cases

Example

```
<link rel="stylesheet" type="text/css" href="screen.css" media="screen">
<link rel="stylesheet" type="text/css" href="print.css"  media="print">
```

Responsive images

Resolution switching

```
<img srcset="small-car-image.jpg 400w,
             medium-car-image.jpg 800w,
             large-car-image.jpg 1200w"
    sizes="(min-width: 1280px) 1200px,
            (min-width: 768px) 400px,
            100vw"
    src="medium-car-image.jpg" alt="Image of a Car">
```

srcset
accepts multiple images and widths
the browser chooses the most appropriate image

sizes
defines the space that the image will take up on the screen

Resolution switching

large-car-image.jpg

medium-car-image.jpg

small-car-image.jpg

Art direction

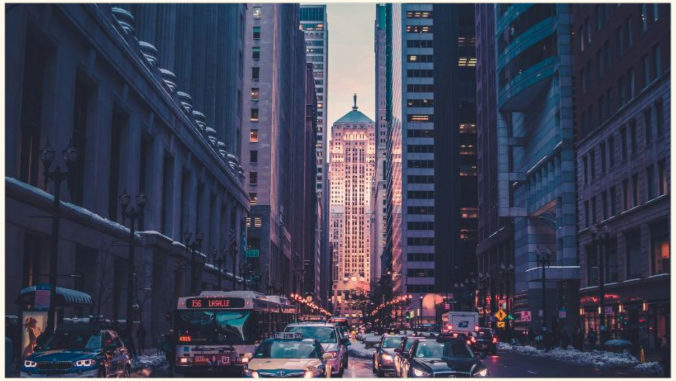
```html
<picture>
  <source media="(max-width: 600px)" srcset="portrait-image.jpg">
  <source media="(min-width: 601px)" srcset="landscape-image.jpg">
  <!-- fallback for browsers that do not support <picture> -->
  <img src="fallback-image.jpg" alt="fallback image">
</picture>
```

Notes:

`<picture>` is supported by modern browsers

`<source>` allows for fine control on when and which image is used

Art direction



landscape-image.jpg

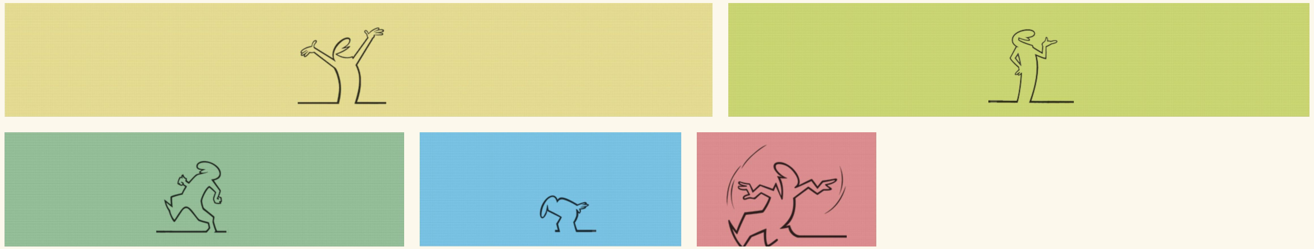
portrait-image.jpg

fallback-image.jpg

Your turn

1.Boxed in

- Requirements
  - Create a fully responsive page
  - The page should show a different image at each breakpoint
  - Pick 5 good breakpoints and change images using media queries
  - Make interesting changes at each breakpoint using the CSS properties that you know
  - Be creative and make a cool sequence of images
  - One idea would be to create a sequence of pictures showing a person in a shrinking space
  - **note:** you don't need the "Responsive images" section to solve this



2.Responsive menu

- Requirements
    - In *02-responsive-menu* create 2 subfolders called *responsive-menu-a* and *responsive-menu-b*
    - Each folder should have a *readme.md*, *style/style.css* and *index.html*
    - For *responsive-menu-a*, use max-width media queries only
    - For *responsive-menu-b*, use min-width media queries only
    - Ensure the menus are semantic, responsive, and include at least 7 items
- References
    - How To Create a Responsive Top Navigation Menu
    - Responsive Menu Concepts
    - How to make a responsive navigation menu
    - Learn from these but write your own code, don't copy
    - Your menu can be with or without Javascript, but it must work

Bonus

3.Full responsive page

- Create a page that has
  - A header, footer, an aside sidebar, and at least two main columns
  - A responsive menu with at least 5 items
  - The header and menu should stay in place when the page scrolls
- Requirements
  - The page should be fully responsive
  - It should use semantic HTML5 elements
  - The page should contain meaningful content
    - Design the page content-first
    - Don't use placeholder text
  - Try to add as many responsive elements as you can
  - Test the page on Chrome, Edge, Firefox and a few mobile browsers (iOS or android)

References

Viewport and media queries

Responsive Web Design Viewport

Responsive Web Design Basics | Web Fundamentals

Beginner's guide to media queries - Learn web development

CSS Media Queries

CSS3 Media Queries - Examples

Responsive images

References

Content first, mobile first, responsive design

Content First, Design Second

Why content-first design makes better websites

Why you should design the content first for better experiences?

What is Mobile First Design?

50 Examples of Responsive Web Design