

# Responsive web pages

CSS media queries

Shadi Lahham - Programmazione web - Frontend - HTML e CSS

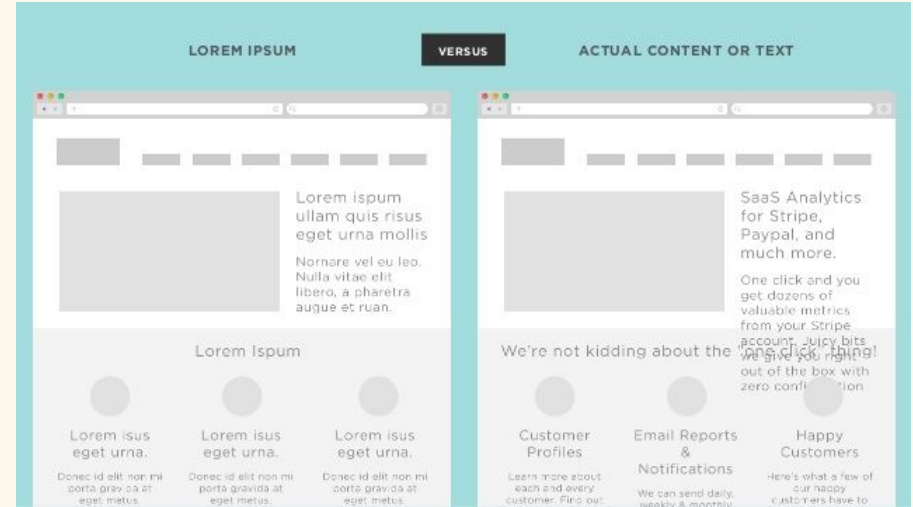
Content first

# 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
  - Risks miscommunications with clients
  - Impacts the User Experience



# How to work content-first

## 1. Prefect 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

# 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) {}
```

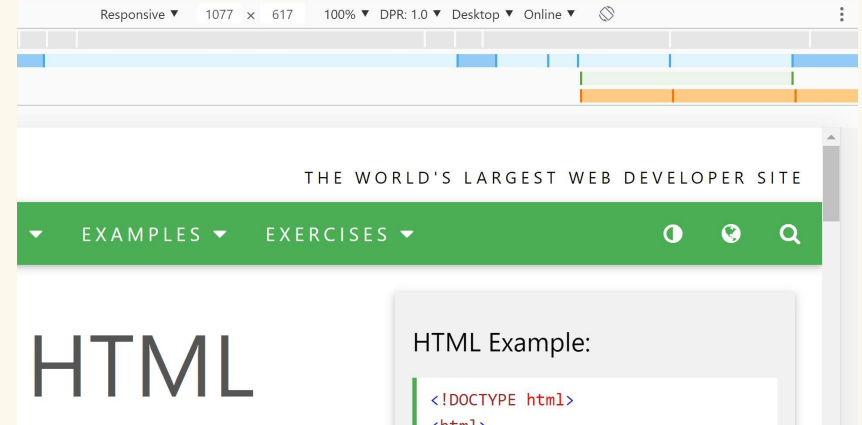
```
/* Large devices (desktops, 992px and up) */  
@media (min-width: 992px) {}
```

```
/* 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

# Debugging with 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





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**

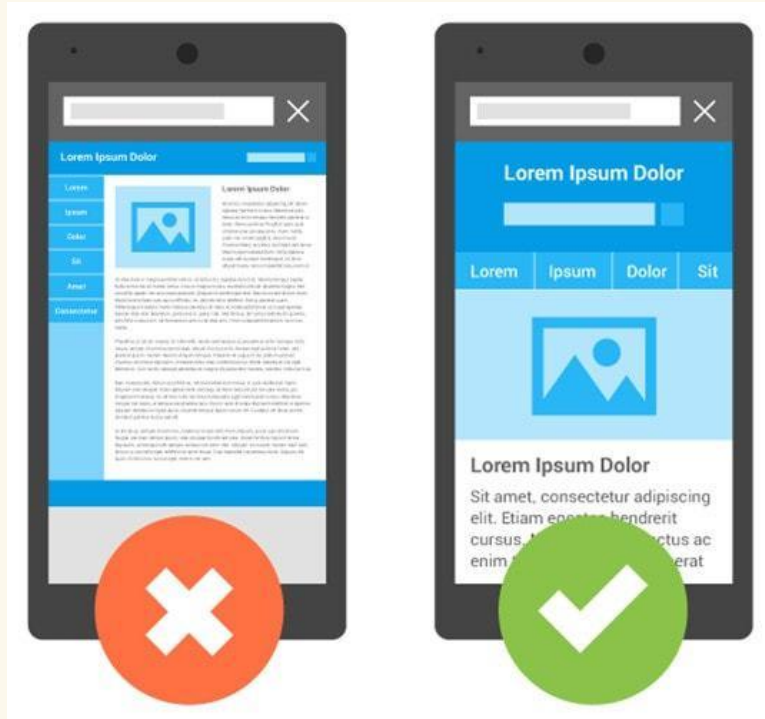
ensure the zoom will not be applied and the layout will always show on a 1:1 scale

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 - breakdown

## Media type

all, screen, print, speech

## Media features

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

Logically represent and, or, not

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

# Media Queries - example

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

# 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

```
.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:

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

```
@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 consectetur 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 consectetur 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 consectetur 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 consectetur 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

```

```

## **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

# Art direction

```
<picture>  
  <source media="(min-width:1024px)" srcset="image-large.jpg">  
  <source media="(min-width:768px)" srcset="image-medium.jpg">  
    
</picture>
```

## Notes:

`<picture>` is supported by [modern browsers](#)

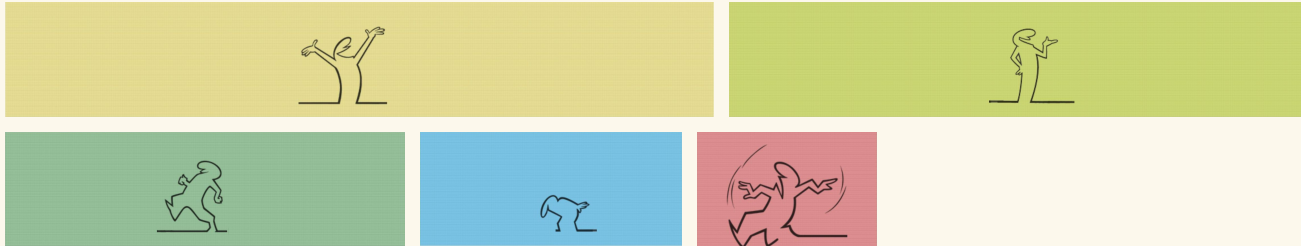
`<source>` allows for fine control on when and which image is used

Your turn



# 1.Boxed in

- Requirements
  - Use a series of images to represent a figure in an increasingly closed space
  - Only one image appears on screen at each breakpoint
  - Choose 5 sensible breakpoints. Use media queries to switch the image at breakpoints
  - Make interesting gradual changes between one breakpoint and another
  - Use the various CSS properties that you know
  - Be creative and try to push your skills to create a nice sequence



## 2.Responsive menu

- Requirements

- In 02-responsive-menu create 2 subfolders called *responsive-menu-a* and *responsive-menu-b*
- Each subfolder should have a readme.md, style/style.css and index.html
- The solution for responsive-menu-a should only use max-width media queries
- The solution for responsive-menu-b should only use min-width media queries
- The menus should be semantic (use correct elements), responsive and contain at least 7 items

- Notes

- Use the following reference links to learn, but don't copy the code directly, create your own menus
- You can create a version without Javascript or include Javascript for opening the menu
- You don't need to explain what the Javascript code does, but it has to work
- [How To Create a Responsive Top Navigation Menu](#)
- [Responsive Menu Concepts](#)

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, IE11 and a mobile browser (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](#)