

SWEN1005

MOBILE WEB PROGRAMMING

Session Four

PAGE LAYOUT, BODY, TABLE VS DIV VS FLEXBOX, LISTS

Page Layout

- Impossible to separate the use of HTML and CSS at this point
- The most important question for each page when determining the layout is: “What is the viewer looking for?”
- Then serve that to the user is simple natural way
- Plan on paper first

Page Layout – Some guidelines

- Simplicity
- Visual Hierarchy
- Navigability
- Consistency
- Accessibility
- Conventionality
- Credibility
- User-Centricity

Simplicity

- Visitors are on your page to find some specific piece of information
- Don't make it harder for visitors to do what they're trying to accomplish.
- From a usability and user experience perspective, keep it simple
 - Typefaces. The typefaces you choose should be legible at the very least, use no more than three
 - Colors. Don't use too many
 - Graphics. Only use them if they'll help a user complete a task or perform a specific function

Visual Hierarchy

- Arrange and organize your website elements so that visitors naturally gravitate toward the most important elements first.
- The goal is to lead visitors to complete a desired action, but in a way that feels natural and enjoyable.
- By adjusting the: position, color, size

Navigability - Tips

- A visitor should be able to arrive on your site and not have to think extensively about where they should click next -- moving from point A to point B should be as pain-free as possible.
- Here are a few tips for optimizing your site's navigation:
 - Keep the structure of your primary navigation simple
 - Include navigation in the footer of your site.
 - Use breadcrumbs on every page (except for the homepage) so people are aware of their navigation trail.

Navigability - Tips

- More tips
 - Include a search box near the top of your site so visitors can search by keywords.
 - Don't offer too many navigation options on a page.
 - Don't dig too deep. In most cases, it's best to keep your navigation to no more than three levels deep.
 - Include links within your page copy, and make it clear where those links lead to.
 - Keep it consistent.

Consistency

- The overall look and feel of your site should be consistent across all of your site's pages.
 - Backgrounds
 - Color schemes
 - Typefaces
 - Even the tone of your writing
- Not every page should have the same exact layout, the layout depends on the information being presented.

Accessibility

- Your site needs to be compatible with the different devices (and operating systems, and browsers) that your visitors are using
- Your website should be responsive
- Adding alt-text to all of your images (so visitors who can't see images in their browsers can still understand what's on the page)
- Should have an identical look and feel across platforms

Conventionality

- There are certain web design conventions which, over the years, internet users have become increasingly familiar with. Such conventions include:
 - Having the main navigation be at the top (or left side) of a page
 - Having a logo at the top left (or center) of a page
 - Having the logo be clickable so it always brings a visitor back to the homepage
 - Having links change color/appearance when you hover over them

Conventionality

- Take advantage of the fact that you already know what types of web experiences they're familiar with
- One of the most common examples of conventionality in web design: Using a shopping cart icon on an ecommerce site.

Credibility

- Web design conventions can help give your site more credibility.
- Credibility (a.k.a. the amount of trust your site conveys) can go a long way.
- Be clear and honest about the product/service you're selling on the site.
- Have a pricing page.

User Centricity

- Usability and user experience hinge on the preferences of the end users.
- Conduct user testing, gather feedback, and make changes based on what you've learned.
- Here are a few user testing tools to get you started:
 - **Crazy Egg.** Use this tool to track multiple domains under one account and uncover insights about your site's performance using four different intelligence tools -- heat map, scroll map, overlay, and confetti.
 - **Loop11.** Use this tool to easily create usability tests -- even if you don't have any HTML experience.

User Centricity

- The User Is Drunk. Pay Richard Littauer to get drunk and review your site. Don't believe me? We tried it. [Check it out.](#)
- [\(Read this for even more helpful tools.\)](#)
- [According to Vitamin T](#), 68% of visitors fail to convert because they don't think you care about their experience. So as a final bit of usability/UX wisdom, start caring more! Put yourself into the shoes of your site's visitors and keep them in mind every step of the way.
- What other principles do you think make for exceptional website design and usability?

User Centricity

- Usability and user experience hinge on the preferences of the end users.
- Conduct user testing, gather feedback, and make changes based on what you've learned.
- There are several website testing tools you can use, the website below lists a few:
 - <https://blog.hubspot.com/marketing/user-testing-tools#sm.0001h5orqb178eebfuriqq3kqc29c>

Session Five

HOMEPAGE LAYOUT

Homepage Layout

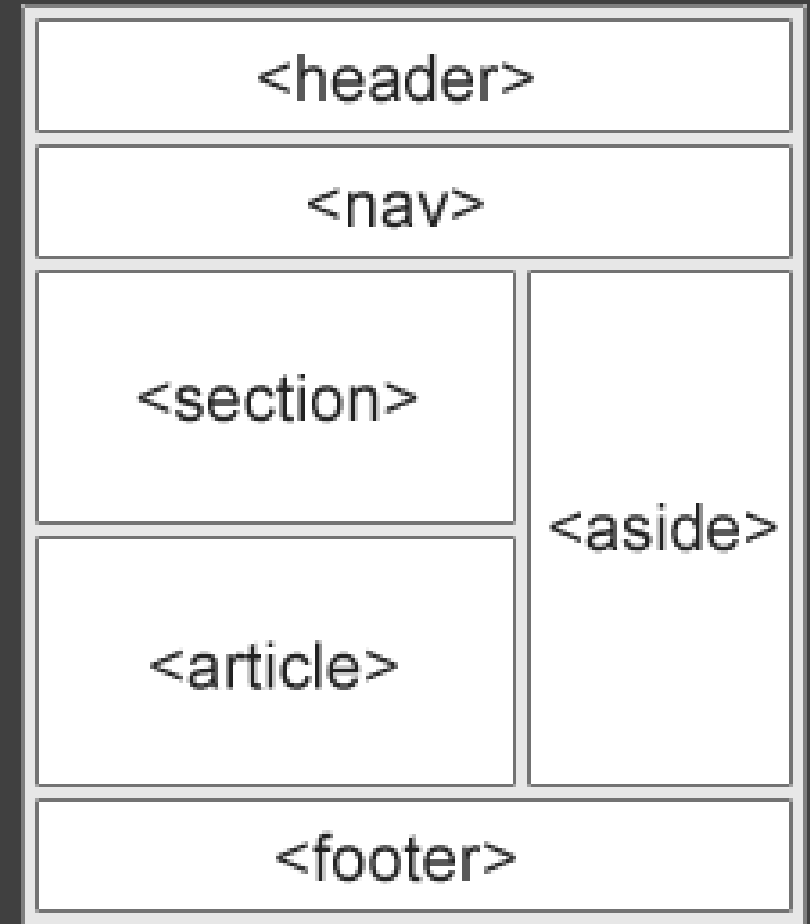
- After designing your page on paper, the next step is to implement that design.
- It must be noted that if your homepage includes graphics you should gather your graphics first.
- Which elements should we use to build our page layout

Semantic Elements

- These are the elements that define the meaning of content more so than the structure of that content.
- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of non-semantic elements:
 - `<div>` and `` - Tells nothing about its content.
- Examples of semantic elements:
 - `<form>`, `<table>`, and `<article>` - Clearly defines its content.

HTML5 offers new semantic elements

- These elements define different parts of a web page:
 - `<article>` `<aside>`
 - `<details>` `<figcaption>`
 - `<figure>` `<footer>`
 - `<header>` `<main>`
 - `<mark>` `<nav>`
 - `<section>` `<summary>`
 - `<time>`



<article>

- The <article> element specifies independent, self-contained content.
- An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
- Examples of where an <article> element can be used:
 - Forum post
 - Blog post
 - Newspaper article

<article> - Example

```
<article>
```

```
  <h1>What is an IPO Diagram?</h1>
```

```
  <p> The input-process-output (IPO) model, or  
input-process-output pattern, is a widely used  
approach in systems analysis and software  
engineering for describing the structure of an  
information processing program or other  
process.</p>
```

```
</article>
```

<aside>

- The <aside> element defines some content aside from the content it is placed in (like a sidebar).
- The aside content should be related to the surrounding content.

Example

```
<p> HTML5 semantic elements are supported in all modern  
browsers.</p>
```

```
<aside>
```

```
    <h4> Modern Browsers</h4>
```

```
    <p> Firefox, Opera, IE, Chrome.</p>
```

```
</aside>
```

<details>

- The <details> tag specifies additional details that the user can view or hide on demand.
- The <details> tag can be used to create an interactive widget that the user can open and close. Any sort of content can be put inside the <details> tag.
- The content of a <details> element should not be visible unless the open attribute is set.

<details> - Example

```
<!DOCTYPE html>
<html>
<body>
<details>
  <summary>Adapted from w3schools</summary>
  <p> - on February 13<sup>th</sup> 2017</p>
  <p>The content however was modified to make it more meaningful. You
  should try it on your own at w3schools.com</p>
</details>
<p><b>Note:</b> The details tag is not supported in Internet
  Explorer.</p>
</body>
</html>
```

<details> - Browser display

► Adapted from w3schools

Note: The details tag is not supported in Internet Explorer.

▼ Adapted from w3schools

- on February 13th 2017

The content however was modified to make it more meaningful. You should try it on your own at w3schools.com

Note: The details tag is not supported in Internet Explorer.

<figure> and <figcaption>

- The purpose of a figure caption is to add a visual explanation to an image.
- In HTML5, an image and a caption can be grouped together in a <figure> element:

<figure>

<figcaption> Fig1. - The Pulpit Rock, Norway. </figcaption>

</figure>

- The element defines the image, the <figcaption> element defines the caption.

<header>

- The <header> element specifies a header for a document or section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.

<header> - Example

```
<article>
```

```
  <header>
```

```
    <h1>What Does WWF Do?</h1>
```

```
    <p>WWF's mission:</p>
```

```
  </header>
```

```
  <p>WWF's mission is to stop the degradation of our planet's natural  
environment,
```

```
    and build a future in which humans live in harmony with  
nature.</p>
```

```
</article>
```

<footer>

- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You may have several <footer> elements in one document.

<footer> - Example

```
<footer>
```

```
  <p>Posted by: Hege Refsnes</p>
```

```
  <p>Contact information: <a ref="mailto:someone@example.com">  
  someone@example.com</a>.</p>
```

```
</footer>
```

<main>

- The <main> tag specifies the main content of a document.
- The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.
- There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

<main> - Example

<main>

<h1>Web Browsers</h1>

<p>Google Chrome, Firefox, and I.E are the top browsers today.</p>

<article>

<h1>Google Chrome</h1>

<p>Google Chrome: a free, open-source web browser by Google, released in 2008.</p>

</article>

<article>

<h1>Internet Explorer</h1>

<p>Internet Explorer: a free web browser from Microsoft, released in 1995.</p>

</article>

<article>

<h1>Mozilla Firefox</h1>

<p>Firefox is a free, open-source web browser from Mozilla, released in 2004.</p>

</article>

</main>

<mark>

- The <mark> tag defines marked text.
- Use the <mark> tag if you want to highlight parts of your text.

- Example

```
<body>
```

```
  <p>Do not forget to buy <mark>milk</mark> today.</p>
```

```
</body>
```

Do not forget to buy milk today.

<nav>

- The <nav> tag defines a set of navigation links.
- Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.
- Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

<nav> - Example

```
<nav>
```

```
  <a href="/html/">HTML</a> |
```

```
  <a href="/css/">CSS</a> |
```

```
  <a href="/js/">JavaScript</a> |
```

```
  <a href="/jquery/">jQuery</a>
```

```
</nav>
```

[HTML](#) | [CSS](#) | [JavaScript](#) | [jQuery](#)

<section>

- The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document.

<section>

<h1>What is HTML5?</h1>

<p>HTML5 is the latest version of Hypertext Markup Language.</p>

</section>

<section>

<h1>What is CSS3?</h1>

<p>This is the latest evolution of the Cascading Style Sheets language.</p>

</section>

<Summary>

- The <summary> tag defines a visible heading for the <details> element. The heading can be clicked to view/hide the details.

<summary> - Example

recall from the <details> tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<details>
```

```
  <summary> Adapted from w3schools </summary>
```

```
    <p> - on February 13th 2017</p>
```

```
    <p>The content however was modified to make it more meaningful. You  
      should try it on your own at w3schools.com</p>
```

```
</details>
```

```
<p><b>Note:</b> The details tag is not supported in Internet Explorer.</p>
```

```
</body>
```

```
</html>
```

<summary> - Example

recall from the <details> tag

► Adapted from w3schools

Note: The details tag is not supported in Internet Explorer.

▼ Adapted from w3schools

- on February 13th 2017

The content however was modified to make it more meaningful. You should try it on your own at w3schools.com

Note: The details tag is not supported in Internet Explorer.

<time>

- The <time> tag defines a human-readable date/time.
- This element can also be used to encode dates and times in a machine-readable way so that user agents can offer to add birthday reminders or scheduled events to the user's calendar, and search engines can produce smarter search results.

<time>

- `<p>We open at <time>10:00</time> every morning.</p>`
- `<p>I have a date on <time datetime="2008-02-14 20:00">Valentines day</time>.</p>`
- The time element does not render as anything special in any of the major browsers.

We open at 10:00 every morning.

I have a date on Valentines day.

Session Two

RESPONSIVE PAGE LAYOUT - GRID-VIEW

Grid View

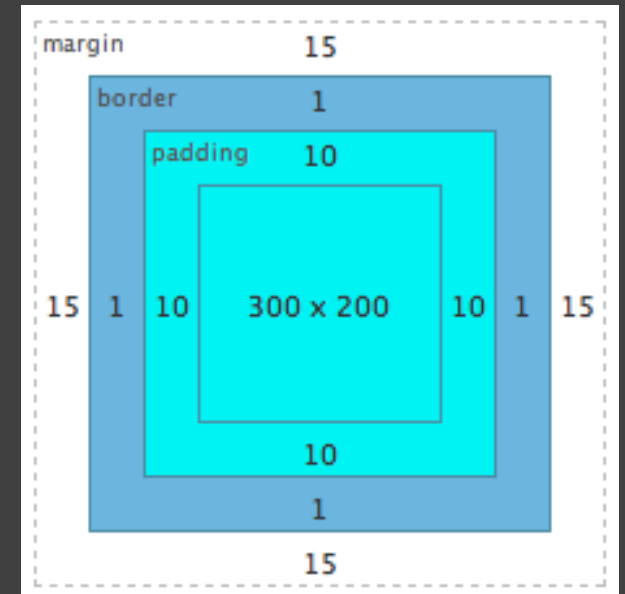
- This means that the page is divided into columns.
- Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.
- A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window. Try it.
 - https://www.w3schools.com/css/tryresponsive_grid.htm

Why use grid view

- Makes it easier to position elements on the page
- This makes it easier to control your layout for various width sizes. Using Media Queries.
- Allows for a fluid design across devices

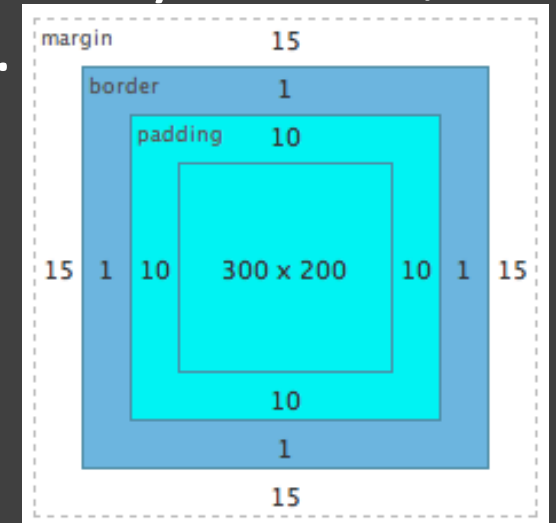
Let's look at box sizing first

- The CSS3 box-sizing property allows us to include the padding and border in an element's total width and height.
- Previously, the width and height of an element is calculated like this:
 - $\text{width} + \text{padding} + \text{border} = \text{actual width of an element}$
 - $\text{height} + \text{padding} + \text{border} = \text{actual height of an element}$



Let's look at box sizing first

- When you set the width/height of an element, the element often appear bigger than you have set (
 - That's because the element's border and padding are added to the element's specified width/height).
- So, web developers specify a smaller width value than they wanted, because they had to include the padding and borders.



Let's look at box sizing first

- The box-sizing property solves this problem.
- The CSS3 box-sizing property allows us to include the padding and border in an element's total width and height.
- If you set `box-sizing: border-box;` on an element padding and border are included in the width and height:

```
*{  
    box-sizing: border-box;  
}
```


Back to Grid View

- First ensure that all HTML elements have the box-sizing property set to border-box.
 - This makes sure that the padding and border are included in the total width and height of the elements.
- We want to use a responsive grid-view with 12 columns, to have more control over the web page.
- First we must calculate the percentage for one column: $100\% / 12 \text{ columns} = 8.33\%$

Back to Grid View

- Then we make one class for each of the 12 columns, class*=“col-”
 - Where: *= is used to select elements whose attribute value contains a specified value.
 - So that all elements with a class attribute value that contains “col-” will be styled

Grid View - Styling

```
<style>
* {
  box-sizing: border-box;
}
[class *= "col-"]
{
  float: left;
  padding: 15px;
  border: 1px solid red;
}
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
```

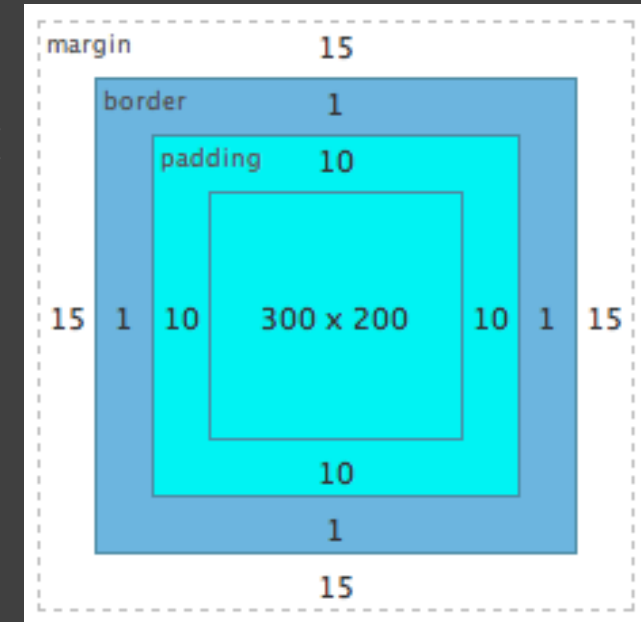
```
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

</style>
```

Grid View - Styling

```
<style>
*{
  box-sizing: border-box;
}
[class*="col-"]
{
  float: left;
  padding: 15px;
  border: 1px solid red;
}
```

- Each column should float to the left
- The padding will be set at 15 pixels
- There will be a red border around each element, 1 pixel wide



Grid View - Styling

- The col- number sets the width property of each column as a percentage of the available width.
 - For example:
 - col-3 – uses 25% of the window
 - col-6 – uses 50% of the window
 - col-9 – uses 75% of the window
 - col-12 – uses 100% of the window
- .col-1 {width: 8.33%;}
 - .col-2 {width: 16.66%;}
 - .col-4 {width: 33.33%;}
 - .col-5 {width: 41.66%;}
 - .col-7 {width: 58.33%;}
 - .col-8 {width: 66.66%;}
 - .col-10 {width: 83.33%;}
 - .col-11 {width: 91.66%;}

Grid View - Use

- Each row should be wrapped in a `<div>` The number of columns inside a row should always add up to 12.
- `<div class="row">`
 - `<div class="col-3">...</div> <!-- 25% -->`
 - `<div class="col-9">...</div> <!-- 75% -->``</div>`

Grid View – Clearing the flow after

- The columns inside a row are all floating to the left, and are therefore taken out of the flow of the page, and other elements will be placed as if the columns do not exist. To prevent this, we will add a style that clears the flow:

- `.row::after`

```
{
  content: " ";
  clear: both;
  display: table;
}
```

Grid View - Styling

```
<style>
...
.row::after {
    content: "";
    clear: both;
    display: table;
}
[class*="col-"]
{
...
}
```

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```


Grid View - implementation

```
<body>
<div class="header">
  <h1>Chania</h1>
</div>
<div class="row">
<div class="col-3">
  <ul>
    <li>The Flight</li>
    <li>The City</li>
    <li>The Island</li>
    <li>The Food</li>
  </ul>
</div>
<div class="col-9">
  <h1>The City</h1>
  <p>Chania is the capital of the Chania
region on the island of Crete. </p>
  <p>Resize the browser window to see
how the content respond to the
resizing.</p>
</div>
</div>
</body>
```

Grid View - Layout

Chania

- The Flight
- The City
- The Island
- The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

Resize the browser window to see how the content respond to the resizing.

Perspective

- The first thing we want to do when we are about to design our page layout is to choose the right semantic elements.
- Since we are designing for all devices we need to use responsive styling methods:
 - Grid View
 - Media Query

Media Query

- Media query is a CSS technique introduced in CSS3.
- It uses the @media rule to include a block of CSS properties only if a certain condition is true

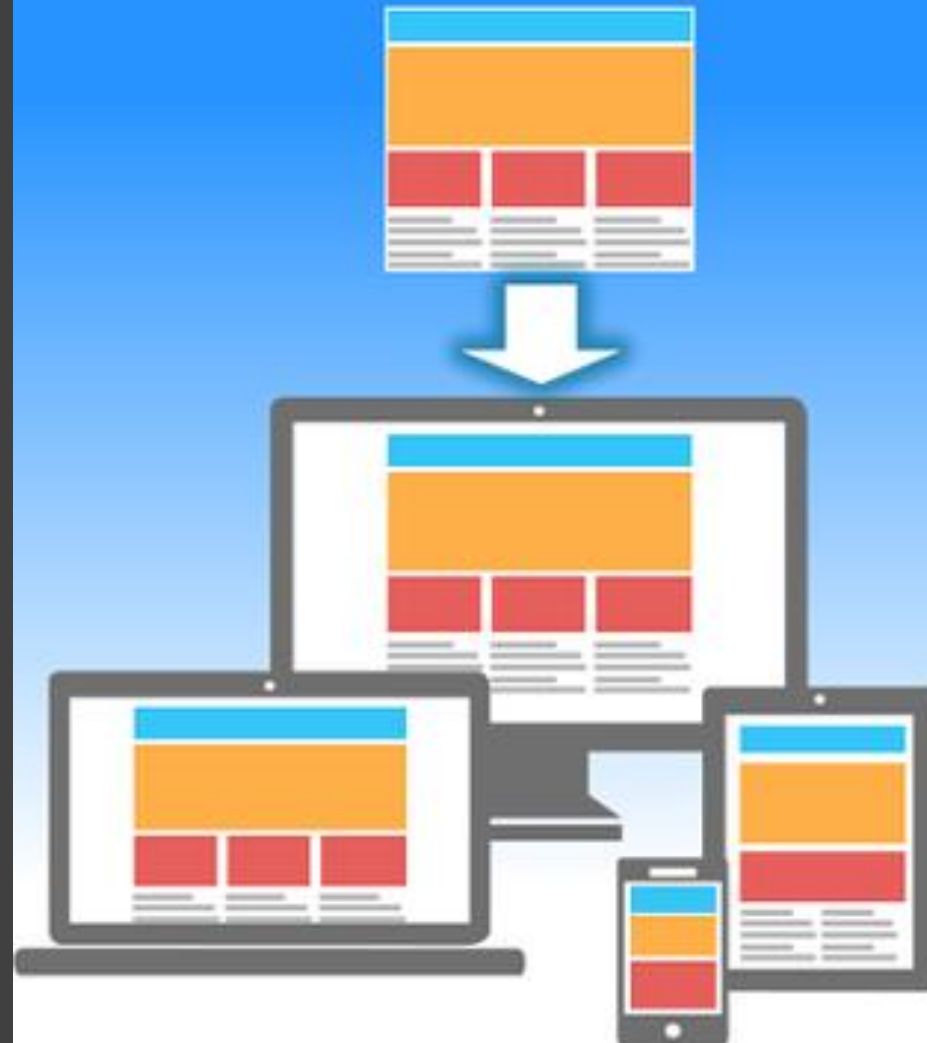
- Example

```
@media only screen and (max-width: 500px)
{
    body {
        background-color: lightblue;
    }
}
```

Responsive
across devices

RESPONSIVE

Universal design which
reflows across displays



What do you think happens here?

```
/* For desktop: */  
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

```
@media only screen and (max-  
width: 768px) {  
    /* For mobile phones: */  
    [class*="col-"] {  
        width: 100%;  
    }  
}
```

Design for mobile first

- Mobile First means designing for mobile before designing for desktop or any other device
 - This will make the page display faster on smaller devices
- **Don't** change the design as the width gets **smaller** than 768px
- **Change** the design when the width gets **larger** than 768px
- This will make our design Mobile First.

Mobile first breakpoint

```
/* For mobile phones: */  
[class*="col-"]  
{  
    width: 100%;  
}
```

```
@media only screen and (min-  
width: 768px) {  
    /* For desktop: */  
    .col-1 {width: 8.33%;}  
        ⋮  
    .col-8 {width: 66.66%;}  
    .col-9 {width: 75%;}  
    .col-10 {width: 83.33%;}  
    .col-11 {width: 91.66%;}  
    .col-12 {width: 100%;}  
}
```


For mobile, tablets and desktop

```
* For mobile phones: */
```

```
[class*="col-"] {  
    width: 100%;  
}
```

```
@media only screen and (min-  
width: 600px) {
```

```
    /* For tablets: */
```

```
    .col-m-1 {width: 8.33%;}
```

```
    .col-m-2 {width: 16.66%;}
```

```
    ...
```

```
    .col-m-11 {width: 91.66%;}
```

```
    .col-m-12 {width: 100%;}
```

```
}
```

```
@media only screen and (min-  
width: 768px) {
```

```
    /* For desktop: */
```

```
    .col-1 {width: 8.33%;}
```

```
    .col-2 {width: 16.66%;}
```

```
    .col-3 {width: 25%;}
```

```
    ...
```

```
    .col-11 {width: 91.66%;}
```

```
    .col-12 {width: 100%;}
```

```
}
```

DaleFRANKLYN

Welcome

Who am I

My Mobile Apps

Improve your customers Web experience...

**I ASSESS, DESIGN and CREATE
mobile responsive websites.**

Do you need a Mobile App?

**I design and create mobile apps for
all platforms**

Let me help you

Design and create mobile apps

- Tailor-made to meet on your needs
- Locally made locally maintained

Make your website mobile responsive

- Improve readability across devices
- Improve user experience

Web experience Consultation

- Free website assessment
- Free website improvement recommendations

DaleFRANKLYN

Welcome

Who am I

My Mobile
Apps

Improve your customers Web experience...

**I ASSESS, DESIGN and CREATE
mobile responsive websites.**

Do you need a Mobile App?

**I design and create mobile apps
for all platforms**

Let me help you

Design and create mobile apps

- Tailor-made to meet on your needs
- Locally made locally maintained

DaleFRANKLYN

Welcome

Who am I

My Mobile Apps

Improve your customers Web experience...

**I ASSESS, DESIGN and CREATE
mobile responsive websites.**

Do you need a Mobile App?

Do you need a Mobile App?

**I design and create mobile apps
for all platforms**

Let me help you

Design and create mobile apps

- Tailor-made to meet on your needs
- Locally made locally maintained

Make your website mobile responsive

- Improve readability across devices
- Improve user experience

Web experience Consultation

- Free website assessment
- Free website improvement recommendations

Quick Review

- Design Layout on paper
- Choose semantic elements
- Use Grid View to create layout
- Set breakpoints with Media Query

Questions?

THE END