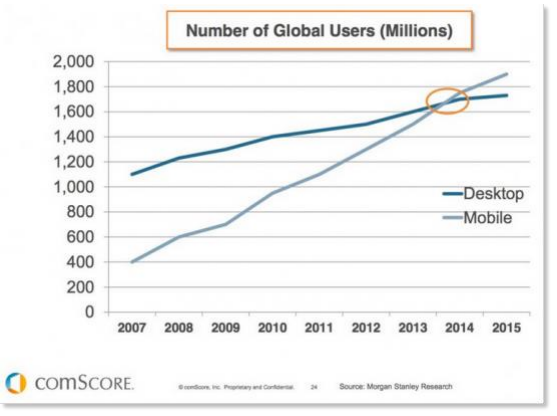
Mobile First Design

WEB DEVELOPMENT FUNDAMENTALS

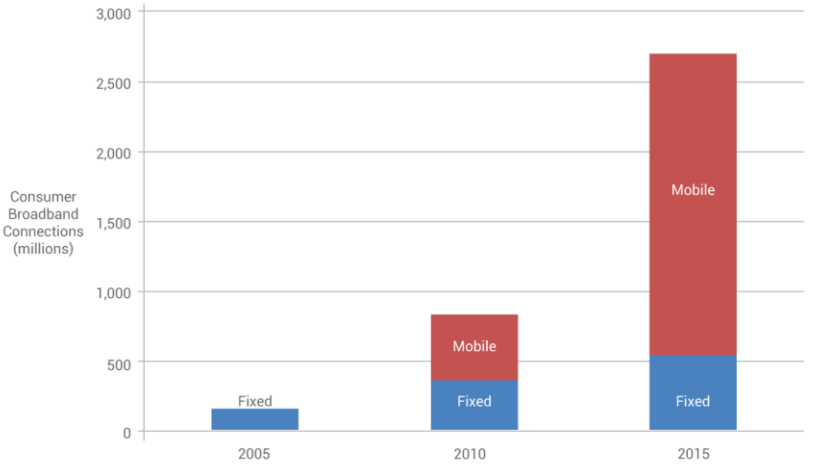
Mobile device growth

- Why go mobile?



2

Mobile Data Usage is Huge



3

Customer satisfaction

72%

of mobile users say it's important to them that websites are mobile-friendly...

...and...

96%

have visited a site that doesn't work well on their device

Users are

5 x

more likely to abandon the task they are trying to complete, if the site isn't optimised for mobile use

...and...

67%

of consumers say that a mobile friendly site makes them more likely to buy a product

4

Strategies for Mobile Development

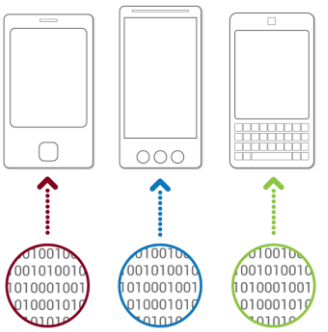
- Go native
  - Create native apps using Java / Objective C / C# / C++
- Embedded HTML
  - Create embedded apps using HTML using PhoneGap (or similar)
- Filter at the server
  - Create separate mobile and desktop websites
- Responsive website design
  - Create a single website using responsive web design

5

Go Native

Native apps are written to work on a specific mobile OS

- **PROS:**
  - Faster, more responsive
  - No network connection required
  - Can have icon on home screen
  - Can use phone hardware (camera, accelerometer, compass, etc.)
- **CONS:**
  - Different development skills required for each platform
  - Different, often unclear, requirements for each platform's store
  - Harder to maintain and update app

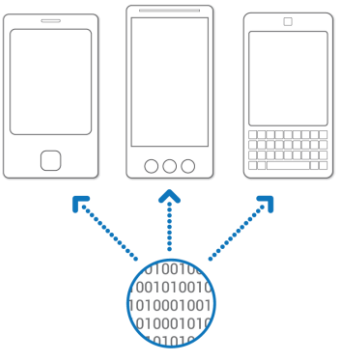


6

Embedded HTML

Package HTML / CSS / JS into a native app

- **PROS:**
  - Looks like a native app to the end users
  - Same source code can be used across multiple platforms
  - Works offline
  - Can use phone hardware (compass, camera, accelerometer, etc.)
- **CONS**
  - Slower and less capable than a native app
  - More effort to deploy and maintain than a website
  - Requires users to download, update, etc.



7

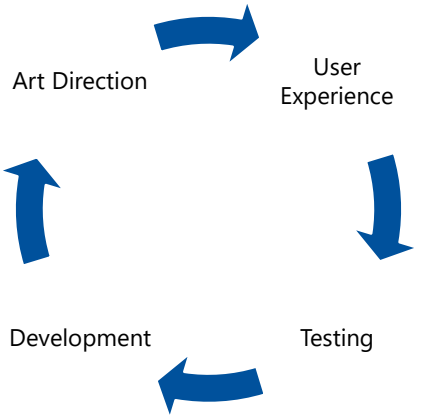
Filter at the Server

Two separate websites, possibly on separate domains. Good approach for web apps.

- **PROS:**
  - Easier to code / separation of concerns
  - Content can be completely different
  - Interaction style can be different (e.g. touch vs. mouse)
- **CONS:**
  - How to reuse content across both sites?
  - What about intermediate-sized devices?
  - Probably need to automatically redirect users
  - What if the user doesn't want to be redirected?

8

Rethinking how we work

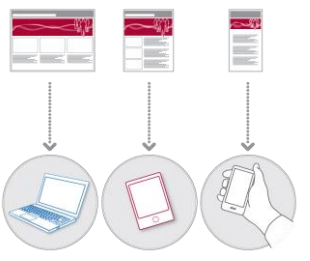


9

Responsive Design

Create a single site that re-styles itself to suit the device

- **PROS:**
  - Mobile-first development demands a focus on key use cases
  - Progressive enhancement will benefit all users, not just mobile
  - Relatively future-proof
  - Much less duplication of content
- CONS:
  - A good multi-device experience requires excellent JavaScript and CSS skills
  - Being efficient for mobile bandwidth requires discipline
  - Downloading large pages, images, scripts, CSS, etc.
  - Require some kind of fallback / poly-fill for older browsers



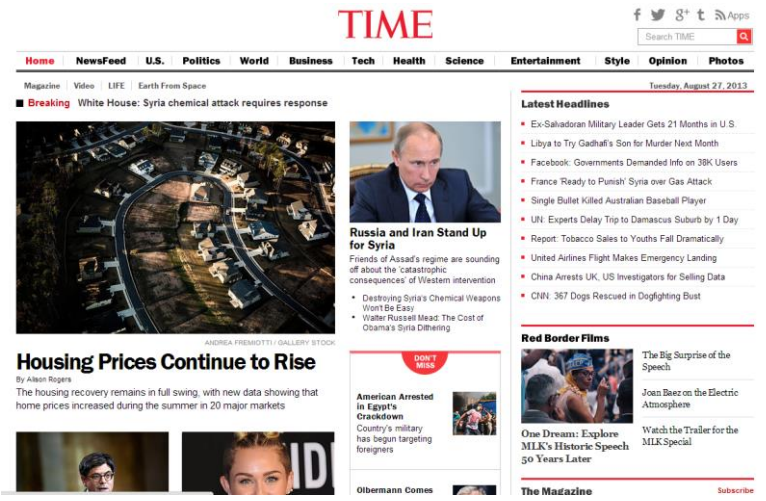
10

Going Responsive – Two strategies

- On a static page mobile devices are challenged
  - A small device 320x480 shows 20% of a page
  - With 80% needing to be reached via pinch and zoom
  - The content is not optimised for mobile bandwidth
    - Delays kill user interest!
- There are two strategies for a responsive website either:
  - To refactor existing solutions progressively to use HTML5 & CSS3
    - Build from 1024x768 down to smaller device settings
  - To rethink development from a mobile first perspective
    - Rebuilding from the smallest device up
    - Considering the information first and presenting it differently
- The remainder of the course focuses on mobile first development

11

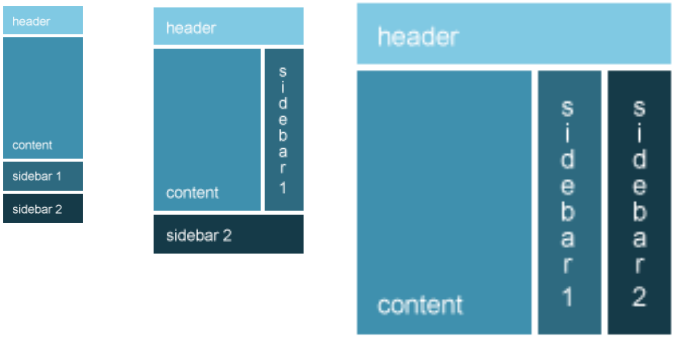
A Mobile First Example – Time Magazine



12

Responsive is not just smaller text

- Responsive development is about re-envisioning content



13

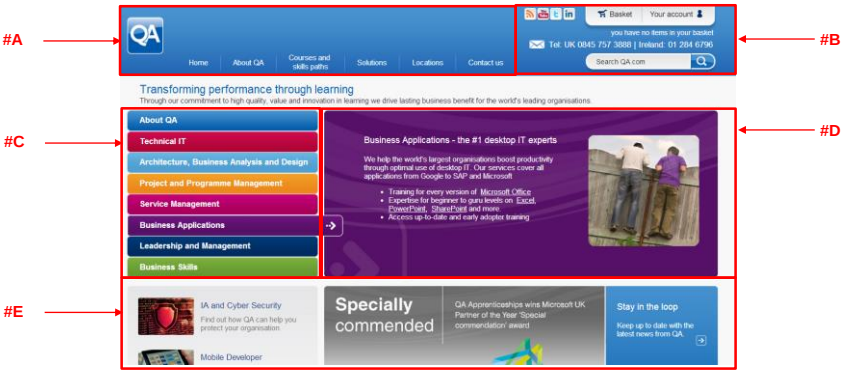
Mobile First and Responsive Design

- Web design and UI development grew out of DTP concepts
  - The web has evolved beyond the desktop screen
  - Now a multi-device, multi-platform world
- This involves a shift in design methodology and paradigm
  - Mobile-first development demands a focus on key use cases
    - Placing the Mobile experience as a key consideration
  - Responsive development places UI layout in the hand of the client
    - Progressive enhancement will benefit all users, not just mobile
    - Uses CSS3 media queries and is relatively future-proof
    - Much less duplication of content – but still too much!
    - Aim is to create efficient content for mobile bandwidth
      - This requires discipline

14

Focusing on Information

- Consider the following Desktop layout
  - We need to consider the areas of intent and how to present them



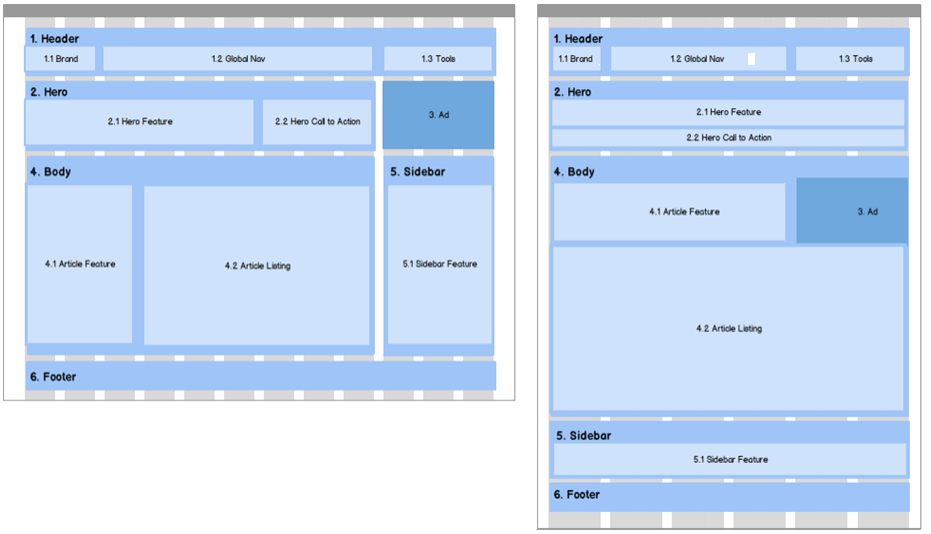
15

The Fluid Grid

- A flexible grid that allows content to be structured consistently and predictably, but also allows for the size and positioning of the elements to change depending on the available size of the rendering device's display
- It is part of the responsive design toolbox, but is not the only one
- Like a grid, it is made of vertical and horizontal alignments
- It concerns itself with reflow of the layout for when the screen size changes rather than changing the size of the grid units or the atomic grid unit

16

The Fluid Grid



17

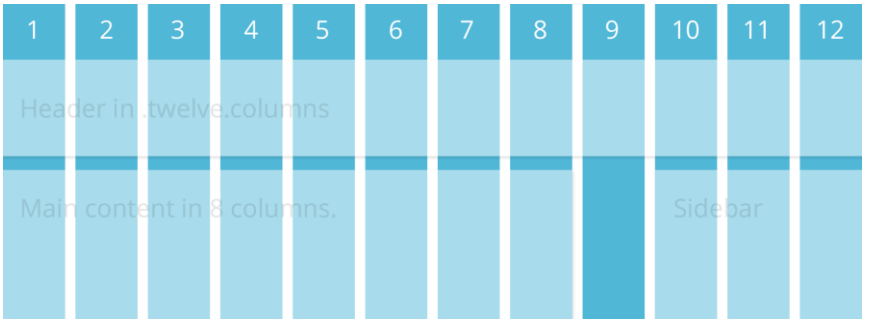
What are Grid Systems and Frameworks?

- Grids are typically CSS frameworks designed to simplify
  - Page layout
  - Cross Browser compatibility
  - Increasingly responsive development patterns
- They are serious pieces of CSS work put together by developers
  - Rigorous software principals of reuse and modularisation are applied
- Providing reusable patterns and tools for:
  - Layout
  - Navigation
  - Typography
  - RIA plugins
- When used by CSS developers aware they make everything easier!

18

Using Foundation – The Grid

- The basis of Foundation is its Grid – it consists of 12 columns
  - The grid is metaphorical, there is no actual grid
  - We use it to space and control areas of content



19

Content Hierarchy

- When you reduce the number of columns, you can't just move the remaining content onto the next line
- You will need to design your layout with a series of grids reflecting the different number of columns that can be sensibly displayed on various devices
- You need to reflect your desired content hierarchy at different resolutions

20

Content Hierarchy

- Before you start designing, you need to;
  - Make decisions about who your audience are
  - Make decisions about the kind of devices and screen resolutions you are going to support
- Once you have made those decisions, you should think about your content hierarchy and what will be important to different users

21

Content Hierarchy

- Example Hierarchy
  - In this example, there are six groups of elements which are listed in order of importance between the header and the footer
    - Group 1: Header
    - Group 2: Featured Content
    - Group 3: Advertising
    - Group 4: Main Content
    - Group 5: Sidebar
    - Group 6: Footer

The order of importance will be debatable, and must be determined in conjunction with various stakeholders

22

Content Hierarchy

- In our example header, we have three elements
  - Element 1: Branding
  - Element 2: Global Navigation
  - Element 3: Tools

- The hierarchy for the header might look like this
  - 1 Header
  - 1.1 Branding
  - 1.2 Global Navigation
  - 1.3 Tools

23

Content Hierarchy

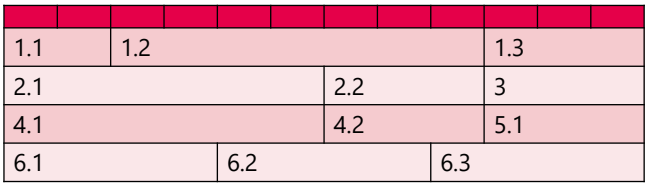
- The full hierarchy might look something like this example

1 Header
1.1 Branding
1.2 Global Navigation
1.3 Tools
2 Featured Content
2.1 Featured Article
2.2 Call to action
3 Advert
4 Main content
4.1 Article
4.2 Article Listing
5 Sidebar
5.1 Sidebar Feature
6 Footer
6.1 Global Navigation
6.2 About
6.3 Legal

24

Grid Layout

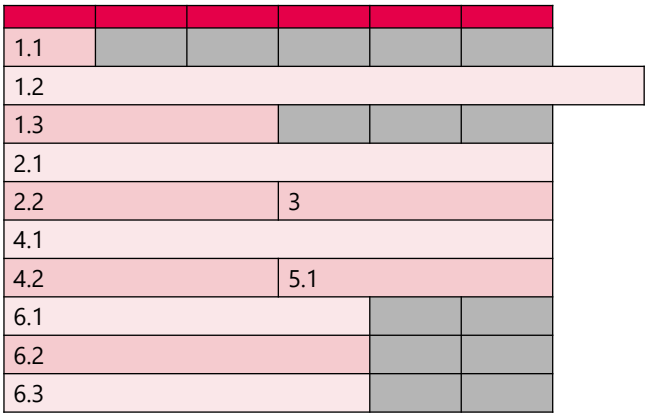
Desktop



25

Grid Layout

Reflow (Tablet) – How not to do it



26

Grid Layout

Reflow (Tablet)

1.1	1.2			1.3	
2.1		2.2		3	
4.1					
4.2					
5.1					
6.1		6.2		6.3	

1.1	1.2			1.3	
2.1			2.2		
3					
4.1					
4.2					
5.1					
6.1		6.2		6.3	

27

Grid Layout

Reflow (Smartphone)

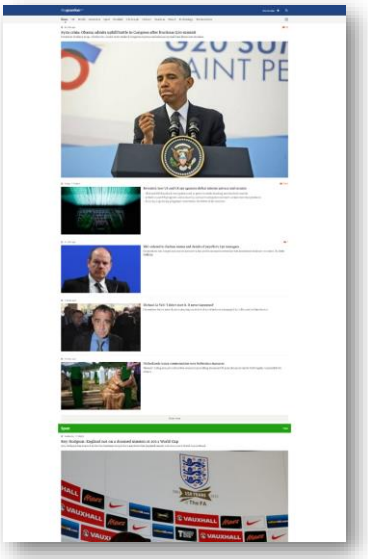
1.1
1.2
1.3
2.1
2.2
3
4.1
4.2
5.1
6.1
6.2
6.3

A mobile screen typically has even less space than a tablet screen

You will want to be cautious about what you show

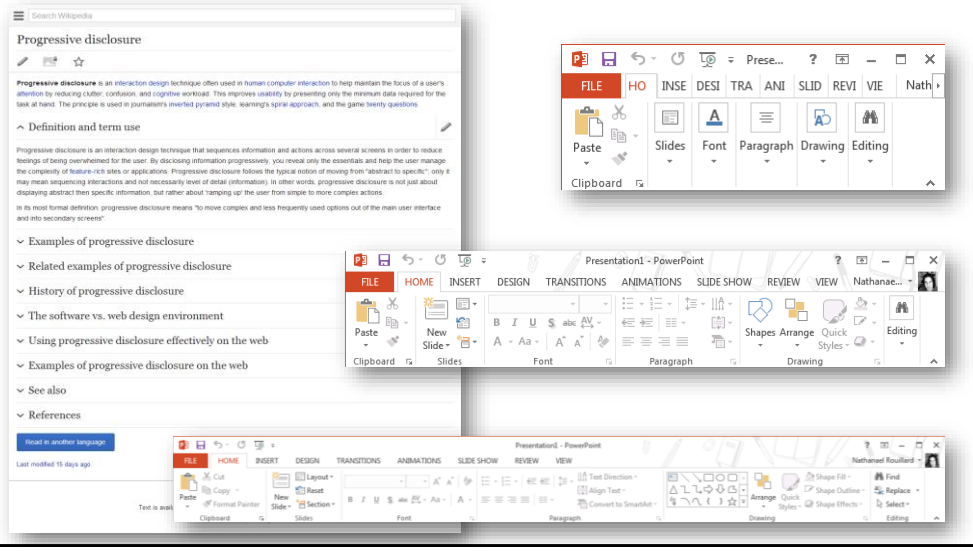
You will either want to use a single unit wide grid, or (at most) a two unit wide grid

This may change as different kinds of mobile display are developed



28

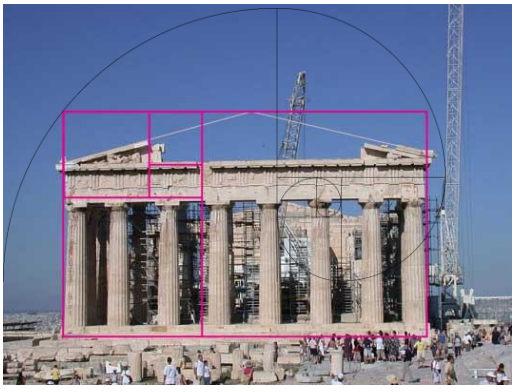
Progressive Disclosure



29

The Golden Ratio

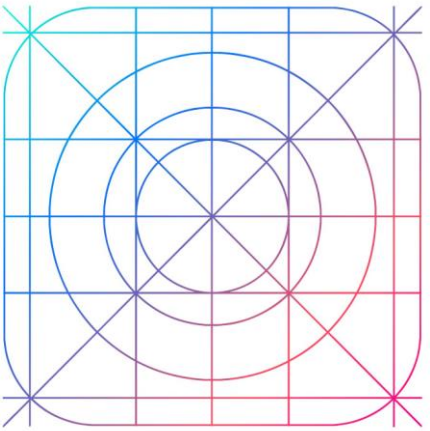
- Based on the Fibonacci Sequence, the Golden Ratio describes the relationship between two proportions
  - Fibonacci numbers, like many elements found in nature, follow a 1:1.61 ratio - this is what we refer to as the Golden Ratio
  - It appears in nature quite commonly
  - It feels pleasing to the eye when we use this same ratio in our UI

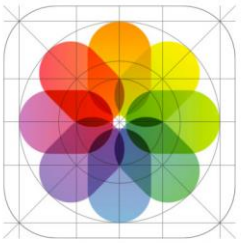


30

Golden Ratio in Design

- Apple use a Golden Ratio Grid System in their product design

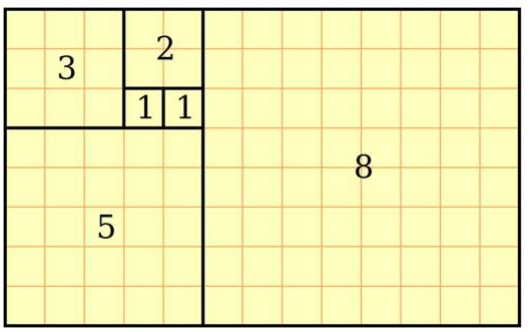




31

Exercise – Laying out a page on a Grid

- Using the page components your instructor has given you, layout a page that tries to follow the golden ratio as a grid



32

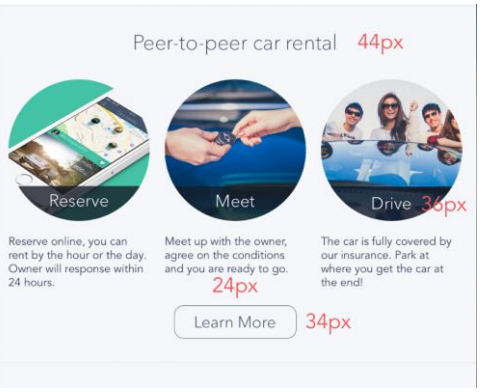
Designing a Fluid Grid

- Split into groups of two or three, and decide on a website concept. Create a Content Hierarchy for that website
- Assume abstract units, and design elements in terms of grid units
- Decide on a number of columns for your desktop website
- Design the layout, and demonstrate how it reflows through different screen sizes
- Does it use Progressive Disclosure?

33

Define Standards

- We need dependable experience across devices – so we need to set standards. The standards below work on all major platforms:
  - The minimum sizes for typography (*24px*+, optimal for reading: *32px*)
  - Buttons (*44px* to *88px*)
  - Navigation bar (*72px* to *98px*)



34

Progressive Enhancement

- Progressive Enhancement starts with the basics and adds embellishments on top
- Establish the basic level of user experience that all browsers and devices will be able to provide
- Provides more advanced functionality that will automatically be available to browsers that can use it
- Improves Accessibility and SEO

35

Progressive Enhancement vs. Graceful Degradation

- Graceful Degradation prioritises presentation
  - Permits less widely used browsers to receive and do less
    - Fallback focused – removes functionality for older browsers
    - No single point of failure
    - Fault containment to prevent propagation of the failure
    - Availability of reversion modes
- Progressive enhancement puts content at the centre
  - Allows most browsers to receive and do more
    - Basic content should be accessible to all web browsers
    - Basic functionality should be accessible to all web browsers
    - Sparse, semantic markup contains all content
    - Enhanced layout is provided by externally linked CSS
    - Enhanced behaviour is provided by externally linked JavaScript
    - End-user web browser preferences are respected

36

Principles of Progressive Enhancement

- Basic content should be accessible to all web browsers
- Basic functionality should be accessible to all web browsers
- All content should be contained within sparse, semantic mark-up
- Enhanced layout is provided by externally linked CSS
- Enhanced behaviour is provided by unobtrusive and externally linked JavaScript
- The user's web browser preferences are to be respected

37

Progressive Enhancement or Graceful Degradation

- When a new browser or extension becomes widely adopted, you can enhance your product further through progressive enhancement without having to re-code the original solution
- Progressive enhancement is the more sophisticated and stable way of keeping the product useful to every user but requires more up front development time and effort
- Graceful degradation is more easily used as a patch for existing products, but takes more effort to maintain

It is possible to mix both methods on one site (and even the same page)

38

When to use Graceful Degradation

- There are a few cases, where graceful degradation is the right choice
  - When maintaining an existing site that uses Graceful Degradation
  - Retrofitting an old product when you do not have the time, access, or insight to replace everything
  - Shortest possible time to initial delivery is of the utmost importance
  - Your product is inherently dependent on scripting for even its most basic functions

39

Discussion - Graceful Degradation & Progressive Enhancement

You have a form field that contains an email address. You want to make sure that there is an email address included before the form is submitted. This can be achieved with JavaScript, but what other approaches and problems should be taken into consideration?

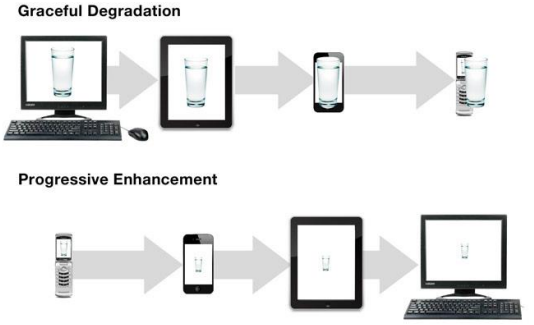
You are building a product using progressive enhancement and you want to display a map inside the product. What base functionality do you start from?

You have an interface that consists of two dropdown form controls. Setting an option in the first, will change the available options in the second. What could be a fall back for this control? What issues could there be with it?

40

Exercise – Rethinking content to be Mobile First

- Using the page snippets from earlier to lay out a page as a phone or a tablet (your instructor will tell you which)
  - Consider how you would restructure the page and content to fit on a screen



41

Text & Proportions

- To some extent, bigger is better when working with mobile
  - But more importantly, it's the proportions that harmonise the content
  - If your body typography is set at *24px*
  - Make sure that the rest of the site is consistent with it
    - There are no hard rules to this, but the *line-height* should be from *1.2x to 1.4x* the font size
      - Many grid systems use the Golden Ratio to achieve this
    - Titles are usually larger by *1.5x-2x*, but thinner as well—otherwise, they steal too much attention
  - Line length should be between 45 and 90 characters

42

What is a Wireframe

- A blueprint of schematic that represents the skeletal framework of a website or application
- It is created as part of a design process to help arrange elements in a design
- It depicts a layout or arrangement of content, interface elements, and navigation systems and how they interact with each other
- Most will lack any visual similarity to the look and feel of the end product

43

What is a Wireframe

- Wireframes can be pen and pencil drawings
- They can be produced by a wide range of software applications
- They are not beholden to any particular platform or medium

44

The Focus of a Wireframe

- Type of information displayed
- Range of functions available
- Relative priorities
- Rules of information display
- Scenario display

45

Types of Wireframe

- Low fidelity
  - Rough sketches or mockups that are quick to produce with low levels of detail
- High fidelity
  - Higher level of detail, suitable for use in project documentation

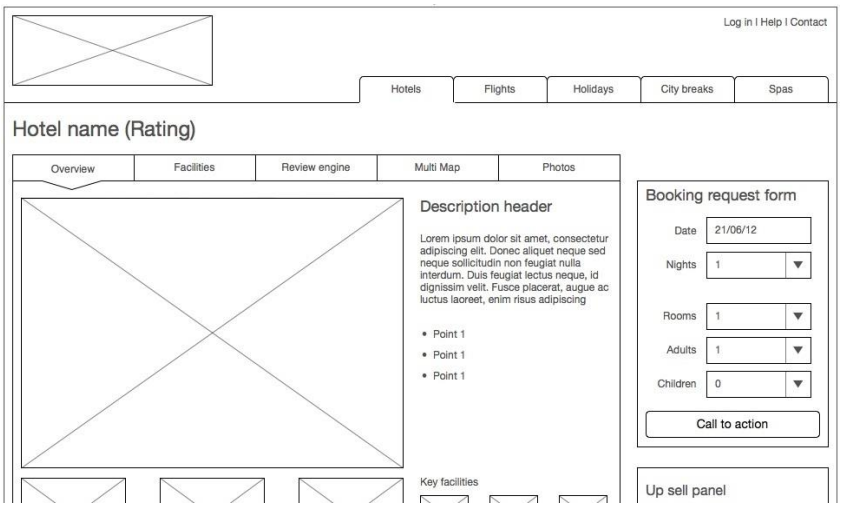
46

Elements of a Wireframe

- The structure of an application can be broken down into three components
  - Information design
  - Navigation design
  - Interface design
- Wireframing is what depicts the relationship between these components

47

Wireframe Symbols



48

The Language of Wireframing

- Widgets
  - Any control or GUI element that displays information that is changeable by the user
- Text
- Graphic Elements
  - Any artwork that is displayed

49

The Language of Wireframing

- Alignment grids
  - A set of invisible lines that content is “snapped to”
  - It is a way of reducing visual clutter, creating clean and evenly spaced layouts, and maintaining coherency across a product or product line
- Icons
  - A pictogram Graphic Element that is displayed to help the user navigate the system, or understand the data presented
  - They are designed to be easily comprehensible and as such are stripped of extraneous detail

50

Exercise – Wireframing a Mobile First Site

- We will redefine the intent of the QA website into a wireframe
  - Prepare a single low fidelity wireframe
  - Then design a higher fidelity wireframe
    - Create phone, tablet and desktop sites

51

Exercise

- Develop a mobile first web page using CSS

52

Media Types & Queries

WEB DEVELOPMENT FUNDAMENTALS

Introduction

- Understanding @media blocks
  - Setting media types
  - Setting parameters
- Working with breakpoints
  - Understanding the concept of breakpoints
  - Implementing common breakpoints
  - Controlling content via breakpoints
- Using media blocks for other types
  - Creating media for print
  - Using print specific instructions

54

What are Media Queries?

- Media queries are the basis of client-side responsive design
  - Allowing us to provide different CSS blocks for
    - Media types e.g. screen and print
    - Parameters e.g. orientation or screen size

55

CSS Media queries – First Look

- Extends the media attribute of CSS2
- CSS3 Media queries allows us to apply CSS selectively
  - Filtering by browser criteria
  - Without JavaScript

```
@media all and (min-width: 640px) {
   #media-queries-1 { background-color: #0f0; }
}
@media screen and (min-width: 600px) and (max-width: 900px) {
  .class {background: #333;}
}
@media screen and (max-width: 480px) {
  .class {background: #000;}
}
```

56

Breakpoints (1)

- Break points are browser widths target by a media query
  - Each query changes the layout
  - Once the browser is within the declared range
- Every responsive site will have a minimum of two break points
  - One for tablets and one for a mobile phone
- 4" Phone

```
@media only screen and (min-width:320px)and(max-width:480px) {

}
```

- 10" Tablet

```
@media only screen and (min-width:768px) and (max-width:1024px) {

}
```

57

Breakpoints (2)

- Additional breakpoints and refinements should be considered
- Mobile Portrait

```
@media only screen and (max-width : 320px) {
}
```

- Mobile Landscape

```
@media only screen and (min-width : 321px) {
}
```

- Tablet Portrait

```
@media only screen and (min-width : 768px) and (max-width
: 1024px) and (orientation : portrait) {

}
```

- Tablet Landscape

```
@media only screen and (min-width : 768px) and (max-width
: 1024px) and (orientation : landscape) {

}
```

58

Modifying Content using breakpoints

- Media Query allows us to specify the content of media
  - Consider the following CSS:

```
@media screen and (min-width: 960px){
  #wrap {width: 960px; margin: 0 auto;}
}
@media screen and (max-width: 320px) {
  #wrap {width: 100%;}
}
```

  - Now consider the following HTML
  - We want this image to adapt to its device viewport

```
<div id="wrap">
  <img src="image.jpg" alt="Image"/>
</div>
```

  - We could fix this with the following CSS:

```
@media screen and (max-width: 320px) {
  #wrap {width: 100%;}
    #wrap img {width: 100%;}
}
```

59

Alternative Media Types - Print

- Media types allow us to target CSS for specific display types
  - Braille, projector and print for example
- Print stylesheets are a small additional amount of work
  - that deliver a significant UI benefit

```
@media only print {

}
```

- Media specific blocks are usually parameterless

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

- They can also be targeted using HTML

60

Print Specific Techniques

- Printer friendly colours are a must! Explicitly change them
  - Turn the background white and text black
- Remove redundant areas for print layout e.g. navigation bars
  - Use display: none;
- When printing to paper, we can use the cm, point and mm units
- Ensure that content is not broken across pages when printed

```
h2, h3 {
   page-break-after: avoid;
}

article {
   page-break-before: always;
}
```

61

Review

- Media queries are the basis of multi device design
  - Create breakpoints based upon device proportions
  - Building for the layout not the device
- Media types also provide a useful tool for non screen elements
  - Building for Print etc.

62

Exercise

- Create a mobile-first navigation structure
- Add breakpoints to optimise our site across different sized devices

63