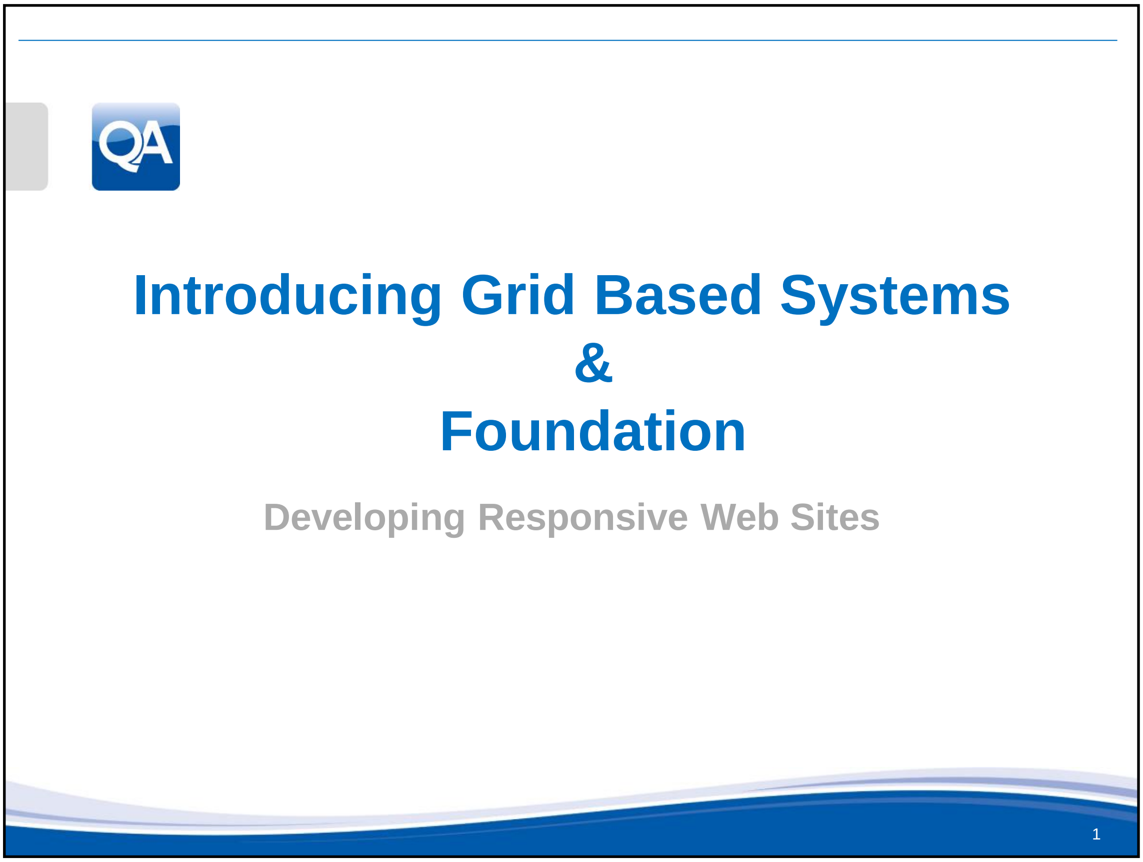
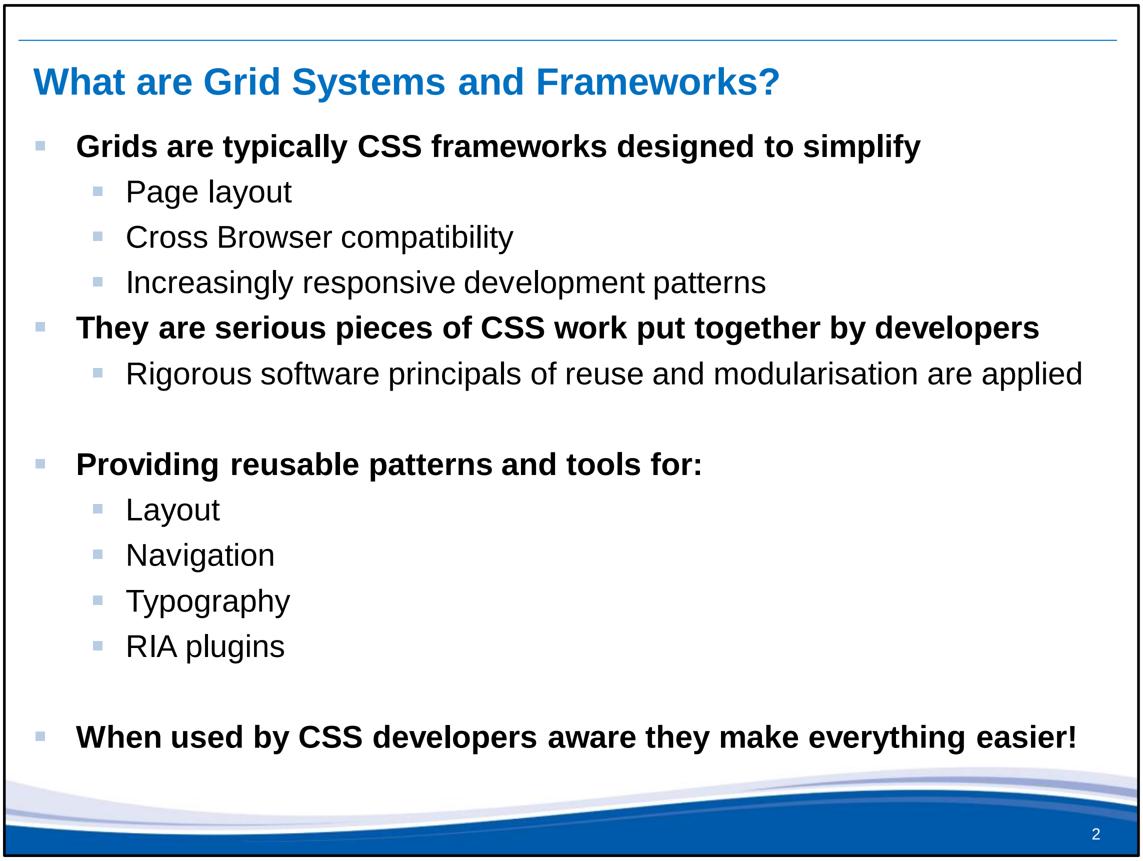


Introducing Grid Based Systems & Foundation

Developing Responsive Web Sites



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!**

The idea of a Grid is not dissimilar to drawing on a piece of graph paper. As a concept in design it became a common part of design and development in the Art Deco movement and give rise to the beloved Helvetica font.

Within web development they are CSS frameworks providing standardised rules for build and shortcuts for site construction using pre-prepared layout classes.

Grids save a lot of time in development, they implement many of the web’s best practices for desktop and mobile development. By learning to build a grid the basis of page construction is done for you leaving you to focus on the functionality and/or appearance of the page.

The CSS file that provides the grid provides a good reuse policy similar to the way jQuery has been used in page scripting.

Examples of Grid Systems

Each grid system is built on a similar principal of UI design using columns as the basis of layout with most grids having between 16 (960) and 12 (Foundation) being common.

960 is a non responsive grid system – best suited to desktop only or server sorted requests.

960 - http://960.gs/

With the rise of responsive development the basis of the 960 framework is used by unsemantic

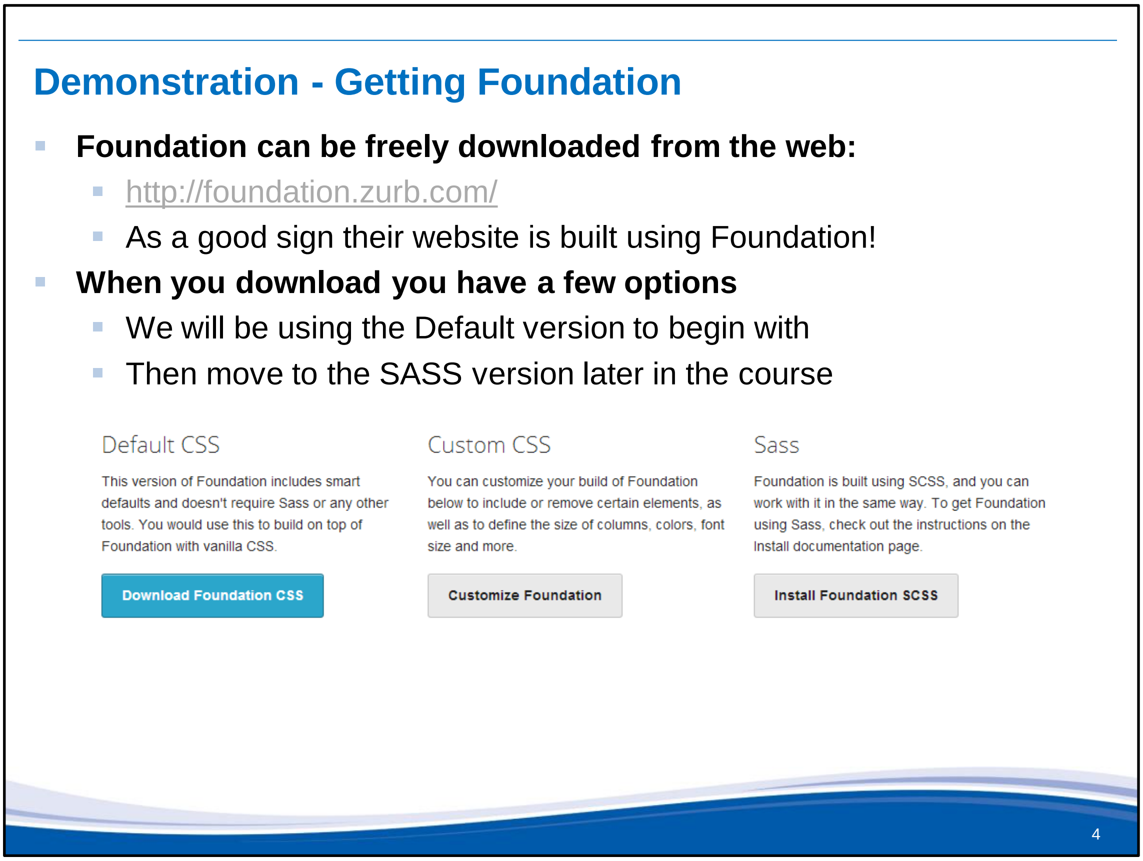
http://unsemantic.com/

Bootstrap is designed by Twitter and made available as a mobile first responsive template and a common site on the web. To push the grid to its limit developers need to master LESS

Foundation 4 is a responsive mobile first framework with a number of useful addons. For the remainder of the course we will work with the Foundation Framework.

All CSS frameworks follow a similar principal, they should be lean and reusable and ensure consistent behaviour across all common browsers (Even IE6).

All Grids use some form of columns and box elements are aligned to the grid element and spaced from the next content area by padding or margin. In responsive browsers this is done using percentages or EMs



Demonstration - Getting Foundation

- **Foundation can be freely downloaded from the web:**
  - http://foundation.zurb.com/
  - As a good sign their website is built using Foundation!
- **When you download you have a few options**
  - We will be using the Default version to begin with
  - Then move to the SASS version later in the course

Default CSS

This version of Foundation includes smart defaults and doesn't require Sass or any other tools. You would use this to build on top of Foundation with vanilla CSS.

Download Foundation CSS

Custom CSS

You can customize your build of Foundation below to include or remove certain elements, as well as to define the size of columns, colors, font size and more.

Customize Foundation

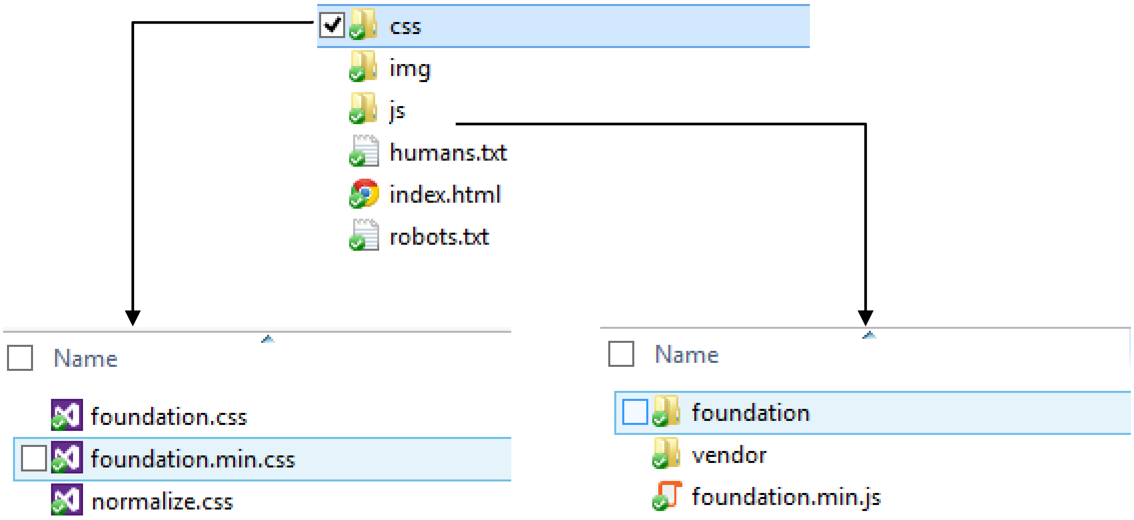
Sass

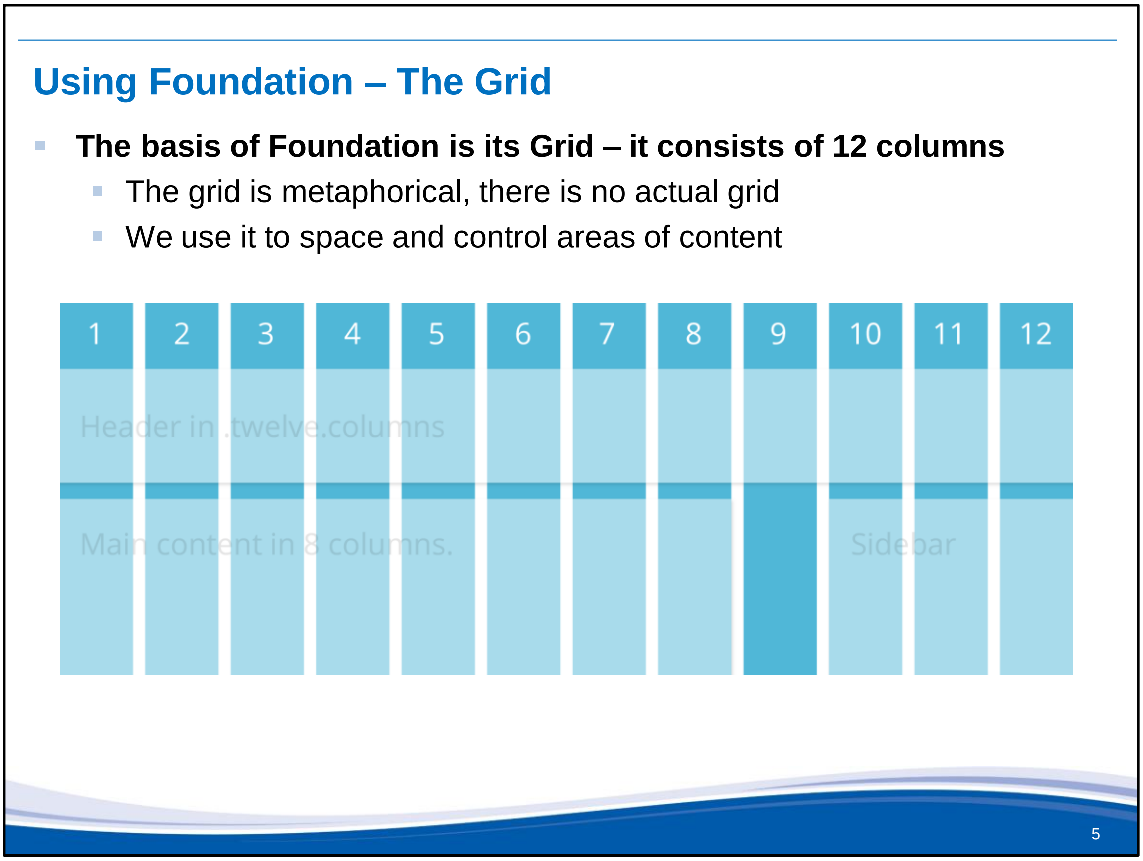
Foundation is built using SCSS, and you can work with it in the same way. To get Foundation using Sass, check out the instructions on the Install documentation page.

Install Foundation SCSS

4

Foundation is a lightweight framework its dependencies consist of linking the **foundation.css** (or min.css) file and if you plan to use the advanced features (We will explore in the next chapter) the f**oundation.min.js** and **zepto.js**





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

5

“The grid is the most vivid manifestation of the will to order in graphic design. Units are the basic building block of a grid. They’re all uniform. Columns are the grouping of units that create the visual structure of the page. They are not necessary uniform.”(sic) – Grids are good

Foundation uses the CSS3 border-box and is mobile-first meaning that content is staked by default.

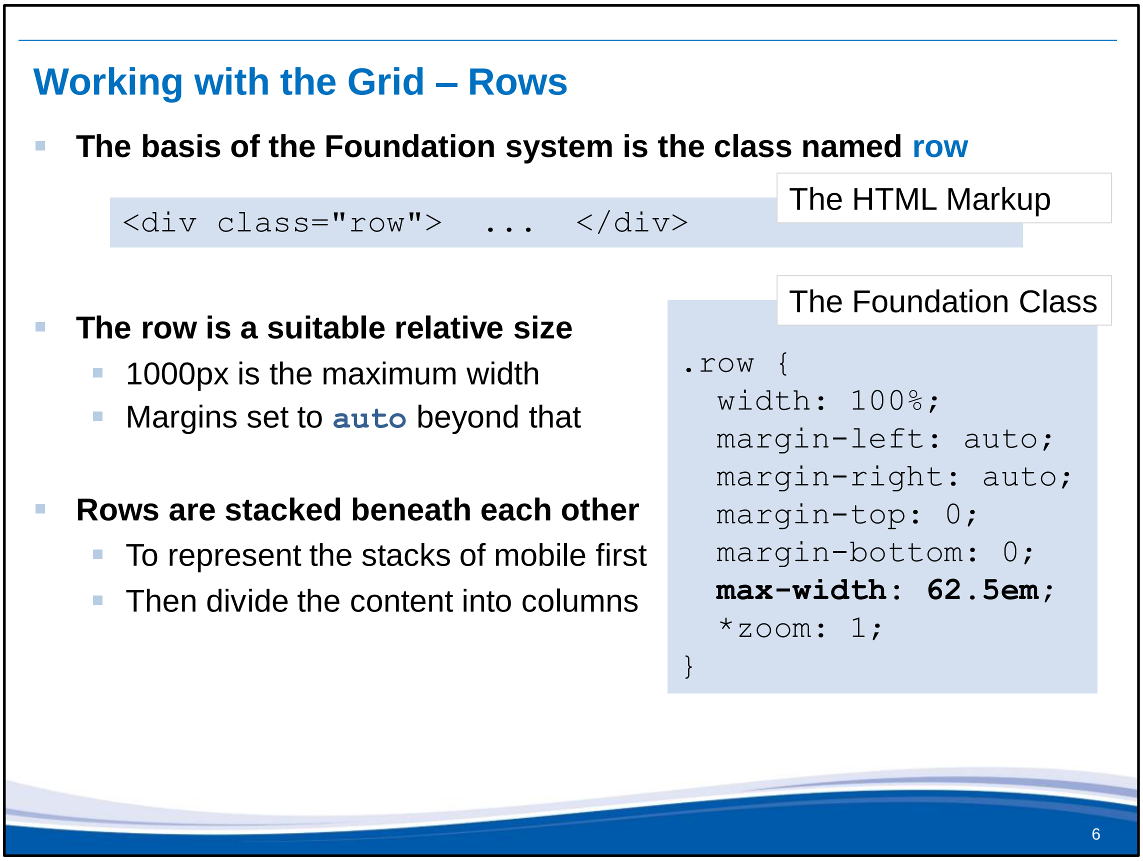
The core of Foundation can be summed up in a few points:

A 12-column, percentage-based grid with an arbitrary maximum width. The grid can be nested and used for quite complex layouts, and it works all the way back to IE 7. The grid reshuffles itself for smaller devices.

Image styles that disregard pixels.
Images in Foundation are scaled by the grid to different widths.

UI and layout elements.
Foundation includes common pieces such as typography and forms, as well as tabs, pagination, N-up grids and more.

Mobile visibility classes.
Rapidly prototyping is partly about having built-in functionality to tailor the experience. Foundation lets you very quickly hide and show elements on desktops, tablets and phones.



Working with the Grid – Rows

- **The basis of the Foundation system is the class named row**

The HTML Markup

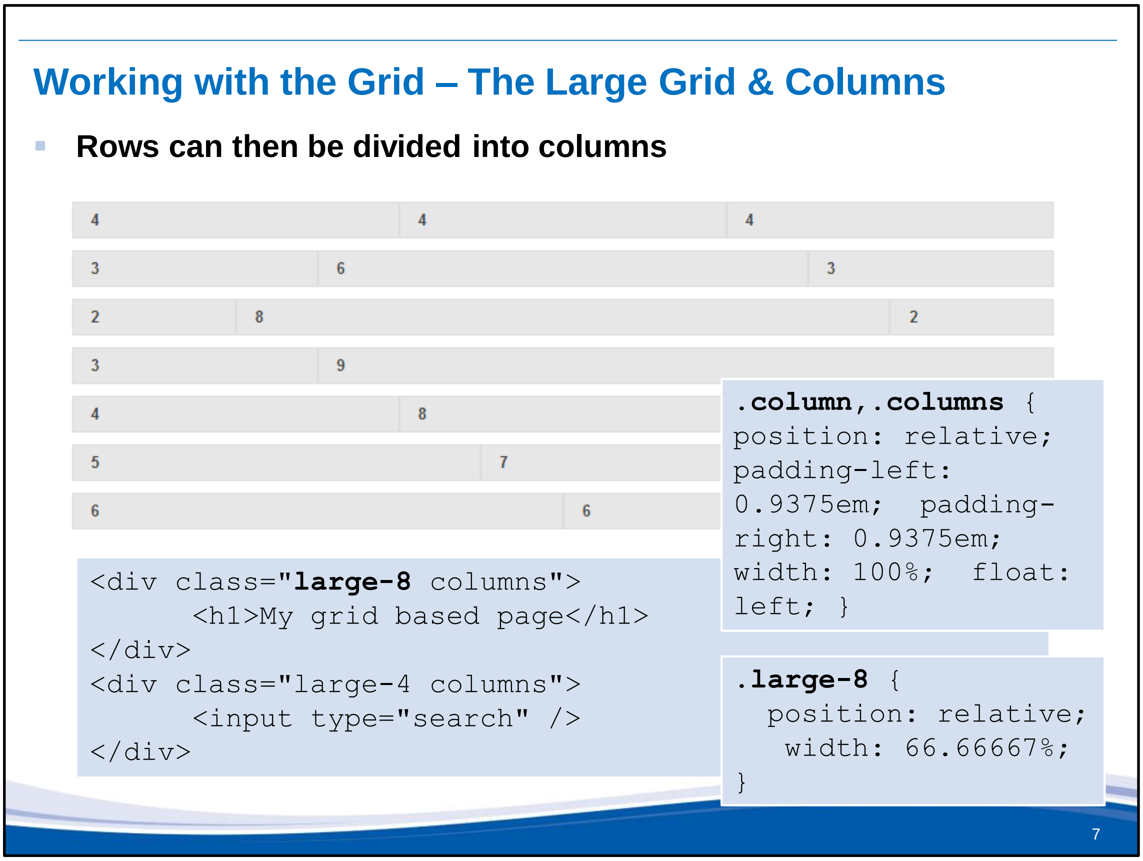
```
<div class="row">  ...  </div>
```

- **The row is a suitable relative size**
  - 1000px is the maximum width
  - Margins set to `auto` beyond that
- **Rows are stacked beneath each other**
  - To represent the stacks of mobile first
  - Then divide the content into columns

The Foundation Class

```
.row {
  width: 100%;
  margin-left: auto;
  margin-right: auto;
  margin-top: 0;
  margin-bottom: 0;
  max-width: 62.5em;
  *zoom: 1;
}
```

One of the critical pieces of device-agnostic design is having a fluid layout that conforms to the size (and orientation) of the device. Foundation’s grid is completely fluid, with percentage-based widths and margins, and it works back to IE7.



Working with the Grid – The Large Grid & Columns

- **Rows can then be divided into columns**

4 4 4
3 6 3
2 8 2
3 9
4 8
5 7
6 6

```
<div class="large-8 columns">
      <h1>My grid based page</h1>
</div>
<div class="large-4 columns">
      <input type="search" />
</div>
```

```
.column,.columns {
position: relative;
padding-left:
0.9375em;  padding-
right: 0.9375em;
width: 100%;  float:
left; }
```

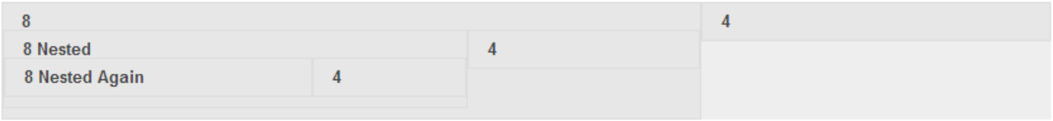
```
.large-8 {
  position: relative;
   width: 66.66667%;
}
```

The row is a relative unit in the space of the viewers device based on EM measurement units. We then subdivide the role using the inbuilt grid classes large-1 through large-12. These classes use a proportionate amount of the row using percentage measurement units. For example large – one uses 8.3% and large – 12 uses 100%.

You commonly need to use the columns class in combination with the large grid classes. The columns class specifies proportional gap between our elements and floats the block to the left. Without this being used the large separated blocks will sit beneath each other.

Working with the Grid - Nesting

- **Once we have created the grid this block can be further subdivided**
  - By reuse of the grid and columns classes.



8

4

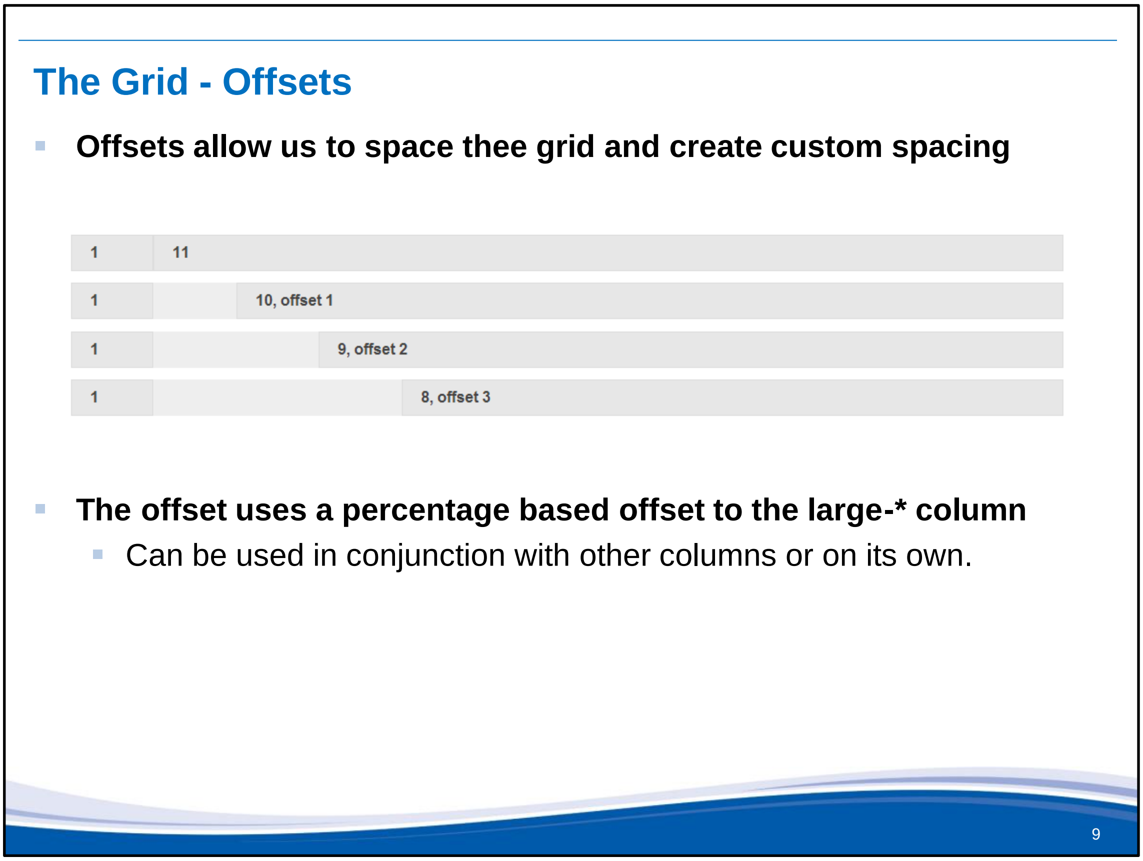
8 Nested

4

8 Nested Again

4

- **The foundation grid can be infinitely subdivided**
  - Each time you are taking a percentage and subdividing it
    - In relation to the width of the outer row EM unit.



The Grid - Offsets

- **Offsets allow us to space thee grid and create custom spacing**

1	11	
1		10, offset 1
1		9, offset 2
1		8, offset 3

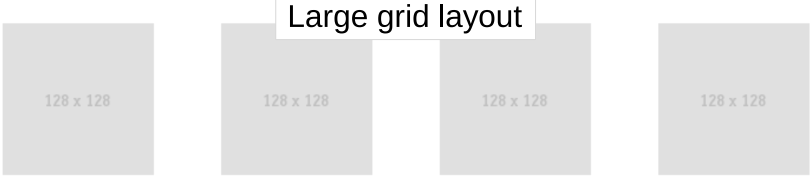
- **The offset uses a percentage based offset to the large-* column**
  - Can be used in conjunction with other columns or on its own.

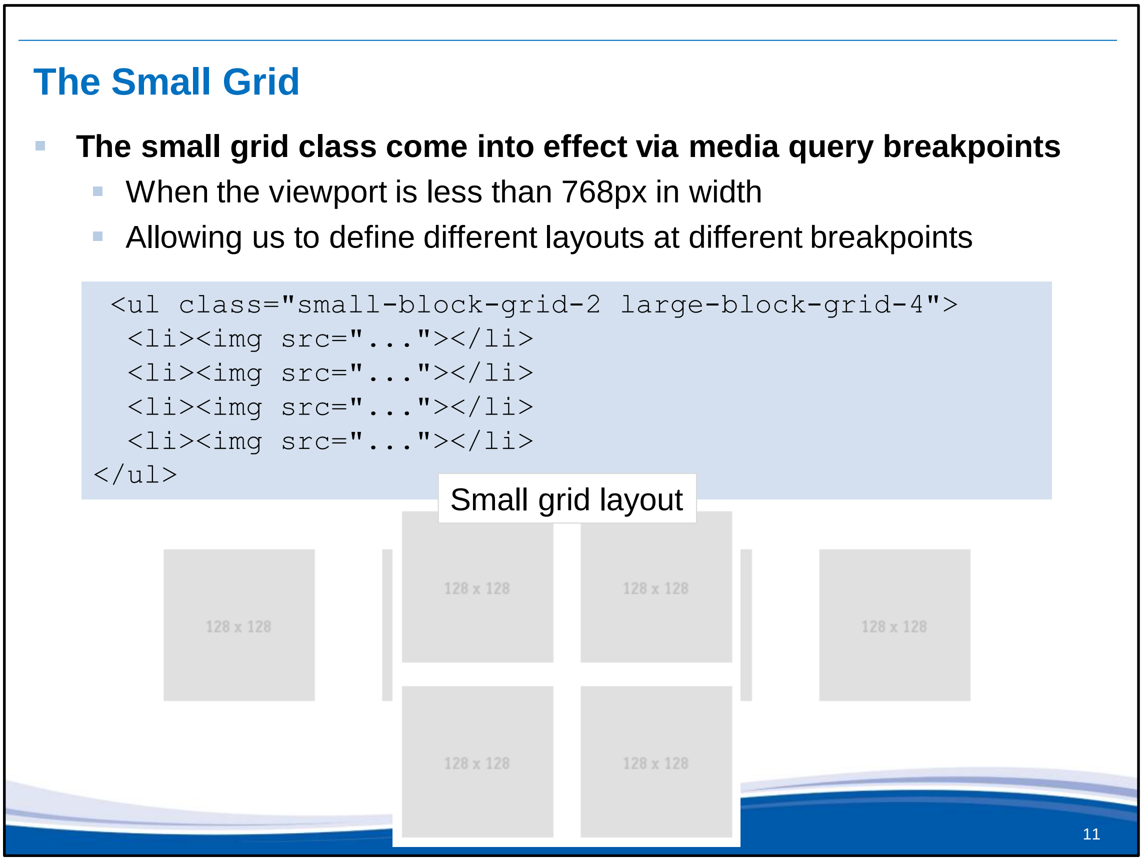
```
<div class="row" style="background-color:indianred">
        <div class="large-2 columns" style="background-color:cadetblue">content</div>
        <div class="large-offset-3" style="background-color:cadetblue">content</div>
 </div>
```

The Grid – Block Grids

- **The block grid allows us to group a defined number of items**
  - Often used with images but can be used with blocks

- **Block grids use ul.small-block-grid-# or ul.large-block-grid-#**
  - These are ideal for blocked-in content generated by an application
  - They do not require rows or number of elements to display correctly


Large grid layout

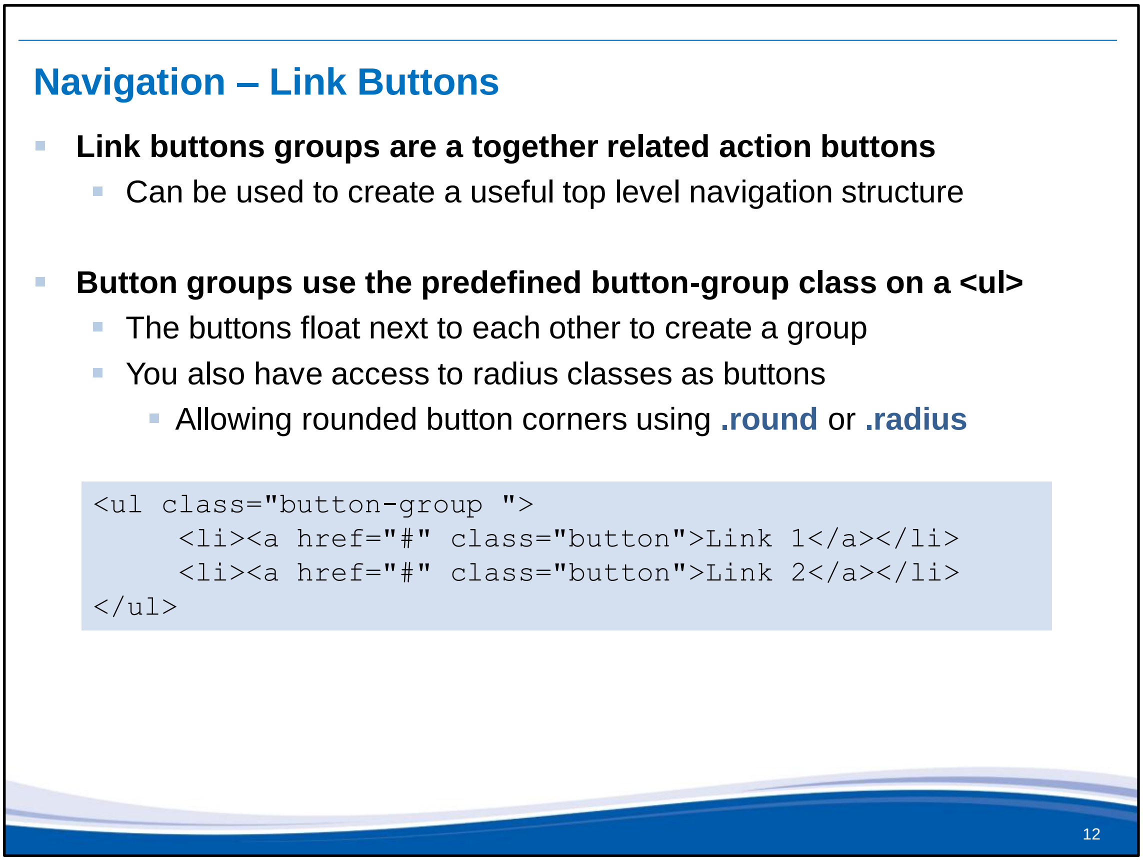


The Small Grid

- **The small grid class come into effect via media query breakpoints**
  - When the viewport is less than 768px in width
  - Allowing us to define different layouts at different breakpoints

```
 <ul class="small-block-grid-2 large-block-grid-4">
  <li><img src="..."></li>
  <li><img src="..."></li>
  <li><img src="..."></li>
  <li><img src="..."></li>
</ul>
```

Small grid layout

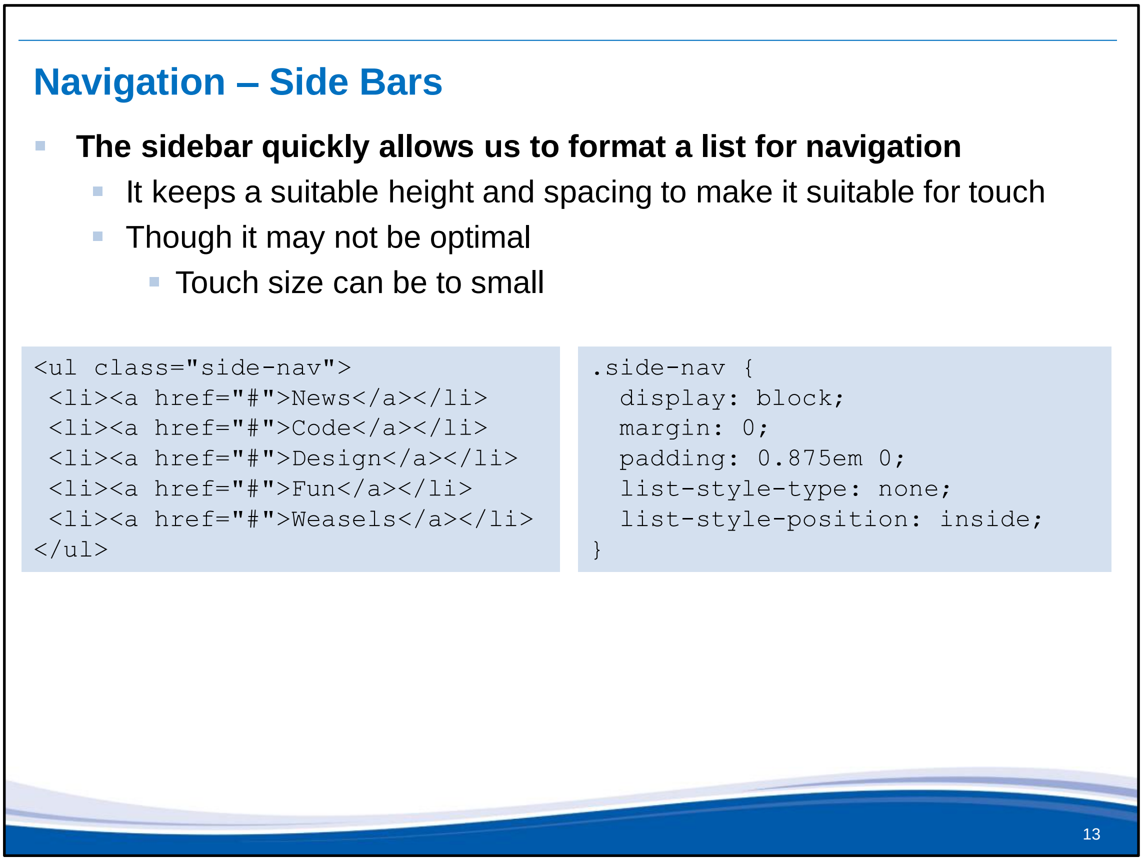


Navigation – Link Buttons

- **Link buttons groups are a together related action buttons**
  - Can be used to create a useful top level navigation structure

- **Button groups use the predefined button-group class on a <ul>**
  - The buttons float next to each other to create a group
  - You also have access to radius classes as buttons
    - Allowing rounded button corners using **.round** or **.radius**

```
<ul class="button-group ">
     <li><a href="#" class="button">Link 1</a></li>
     <li><a href="#" class="button">Link 2</a></li>
</ul>
```

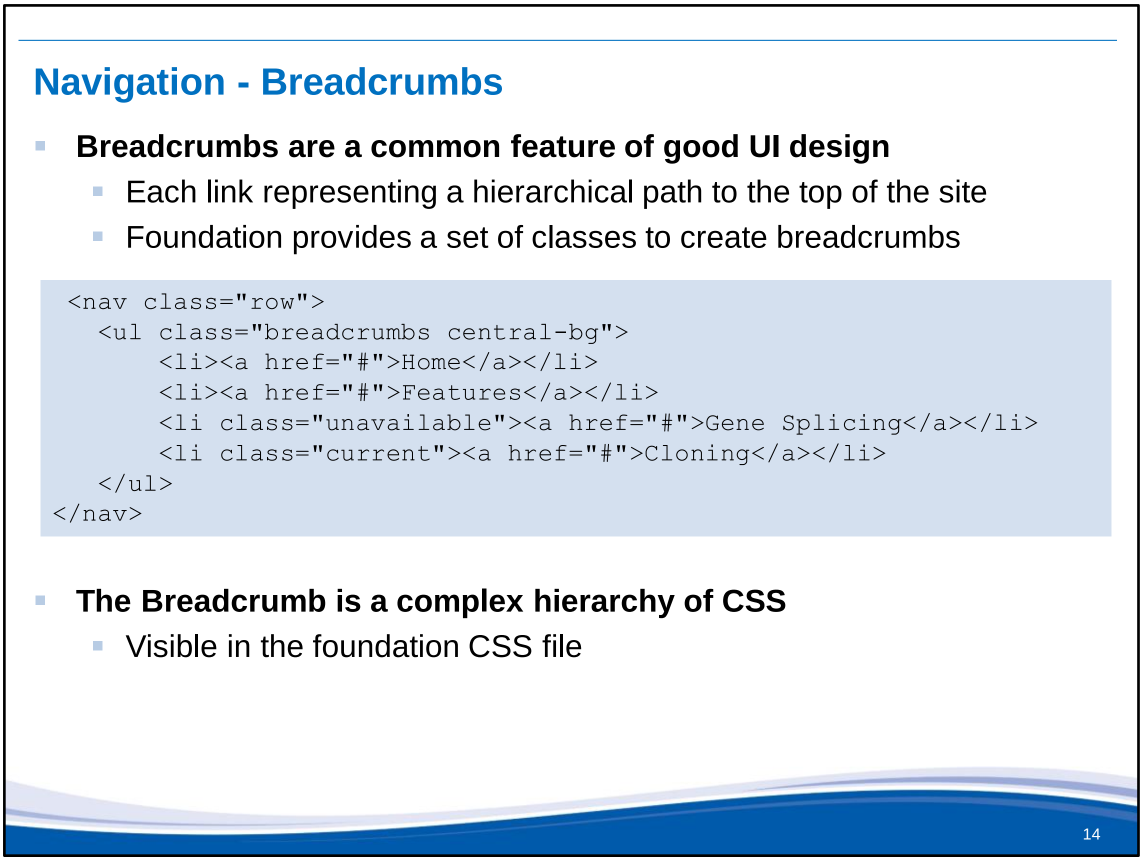


Navigation – Side Bars

- **The sidebar quickly allows us to format a list for navigation**
  - It keeps a suitable height and spacing to make it suitable for touch
  - Though it may not be optimal
    - Touch size can be to small

```
<ul class="side-nav">
 <li><a href="#">News</a></li>
 <li><a href="#">Code</a></li>
 <li><a href="#">Design</a></li>
 <li><a href="#">Fun</a></li>
 <li><a href="#">Weasels</a></li>
</ul>
```

```
.side-nav {
  display: block;
  margin: 0;
  padding: 0.875em 0;
  list-style-type: none;
  list-style-position: inside;
}
```



Navigation - Breadcrumbs

- **Breadcrumbs are a common feature of good UI design**
  - Each link representing a hierarchical path to the top of the site
  - Foundation provides a set of classes to create breadcrumbs

```
 <nav class="row">
   <ul class="breadcrumbs central-bg">
       <li><a href="#">Home</a></li>
       <li><a href="#">Features</a></li>
       <li class="unavailable"><a href="#">Gene Splicing</a></li>
       <li class="current"><a href="#">Cloning</a></li>
   </ul>
</nav>
```

- **The Breadcrumb is a complex hierarchy of CSS**
  - Visible in the foundation CSS file