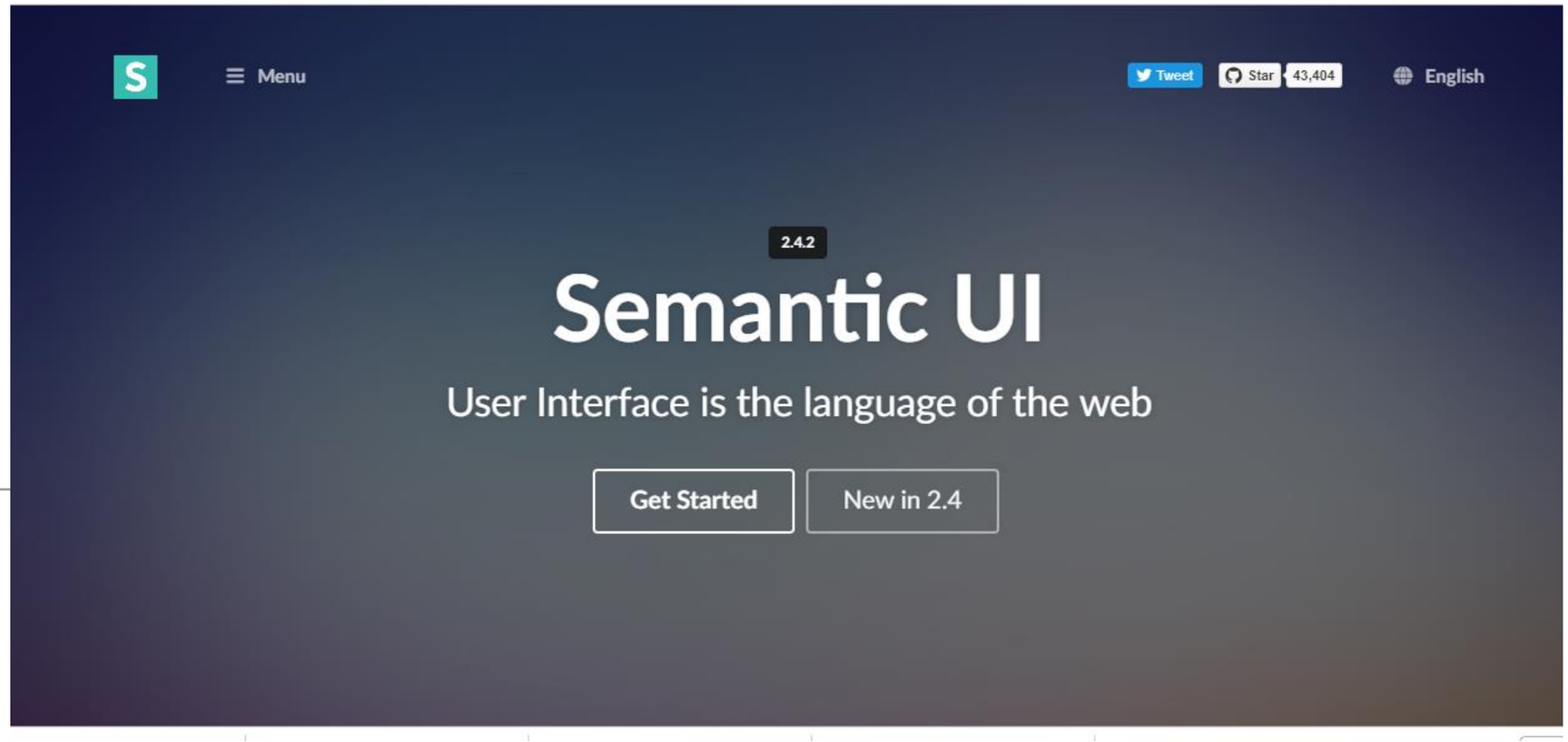
Semantic UI


Menu
Tweet
Star
43,404
English
2.4.2
Semantic UI
User Interface is the language of the web
Get Started
New in 2.4

CSS frameworks

From Wikipedia, the free encyclopedia

CSS frameworks are pre-prepared software frameworks that are meant to allow for easier, more standards-compliant web design using the Cascading Style Sheets language. Most of these frameworks contain at least a grid. More functional frameworks also come with more features and additional JavaScript based functions, but are mostly design oriented and unobtrusive. This differentiates these from functional and full JavaScript frameworks.

Some notable and widely used examples are Bootstrap or Foundation.

CSS frameworks offer different modules and tools:

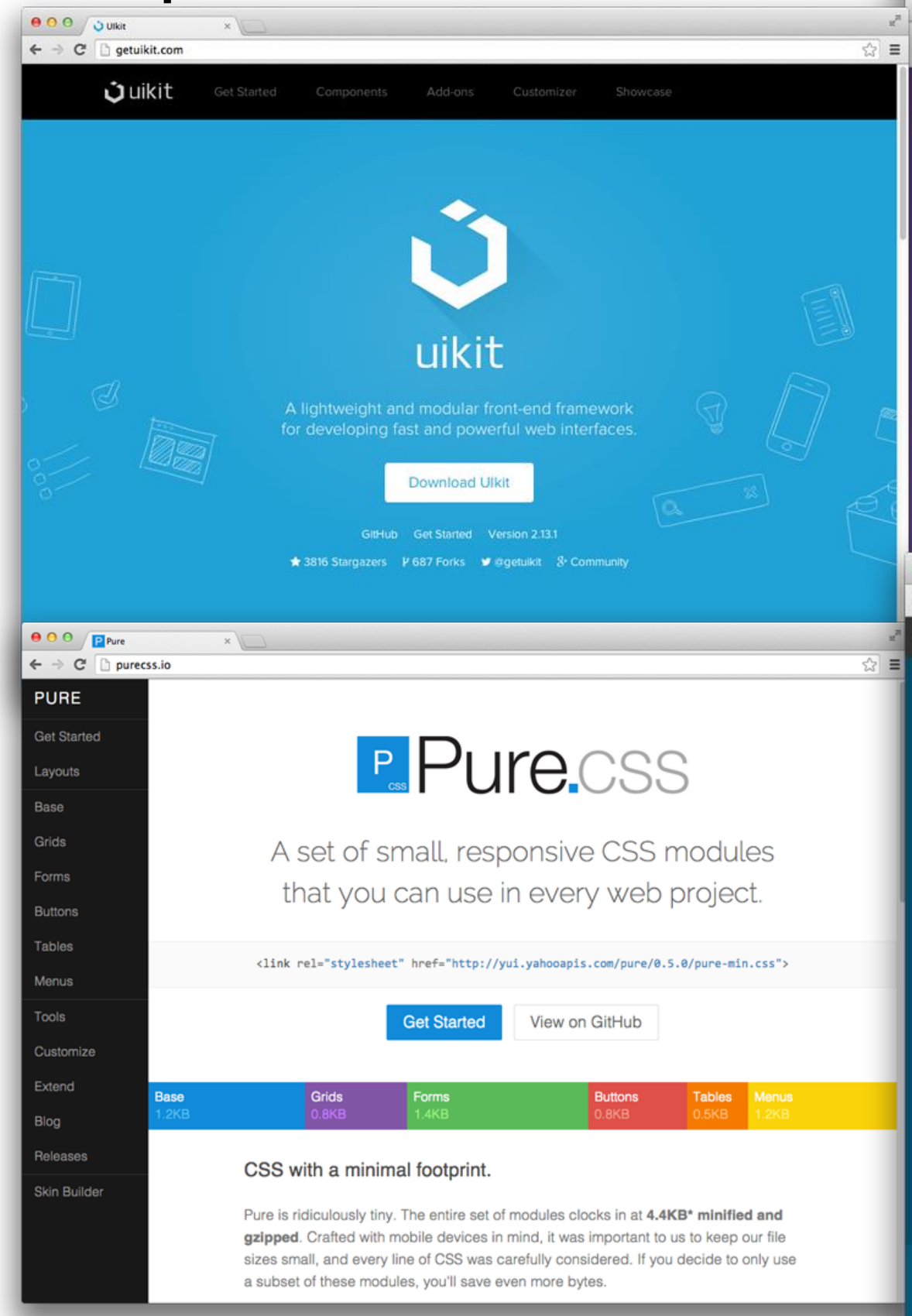
- reset style sheet
- grid especially for responsive web design
- web typography
- set of icons in sprites or icon fonts
- styling for tooltips, buttons, elements of forms
- parts of graphical user interfaces like accordion, tabs, slideshow or modal windows (Lightbox)
- equalizer to create equal height content
- often used css helper classes (*left*, *hide*)

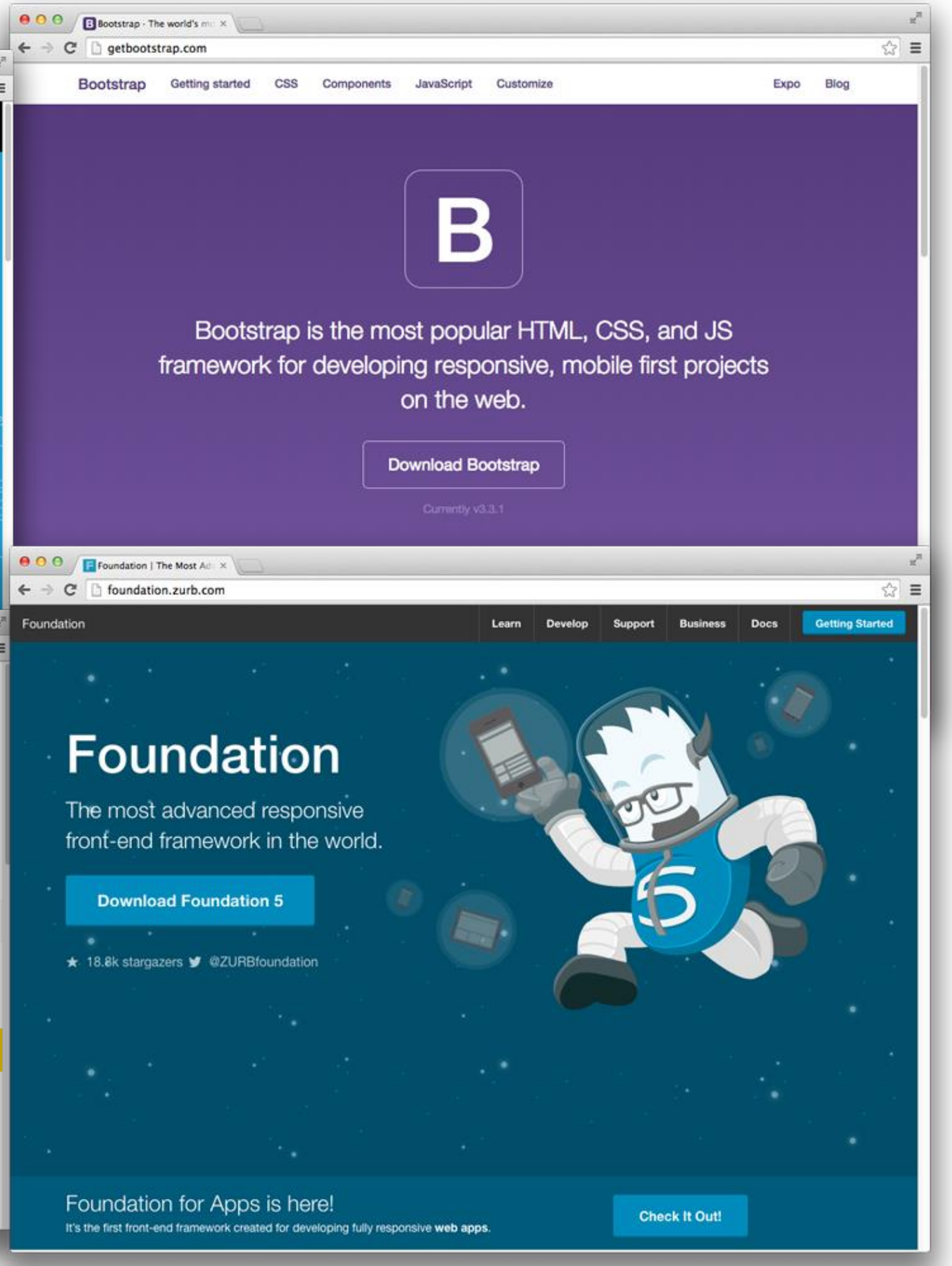
What is a CSS Framework?

“framework is defined as a package made up of a structure of files and folders of standardised code (HTML, CSS, JS etc.) which can be used to support the development of websites, as a basis to start building a site.”

“The aim of frameworks is to provide a common structure so that developers don’t have to redo it from scratch and can reuse the code provided"

Popular Frameworks





Menu

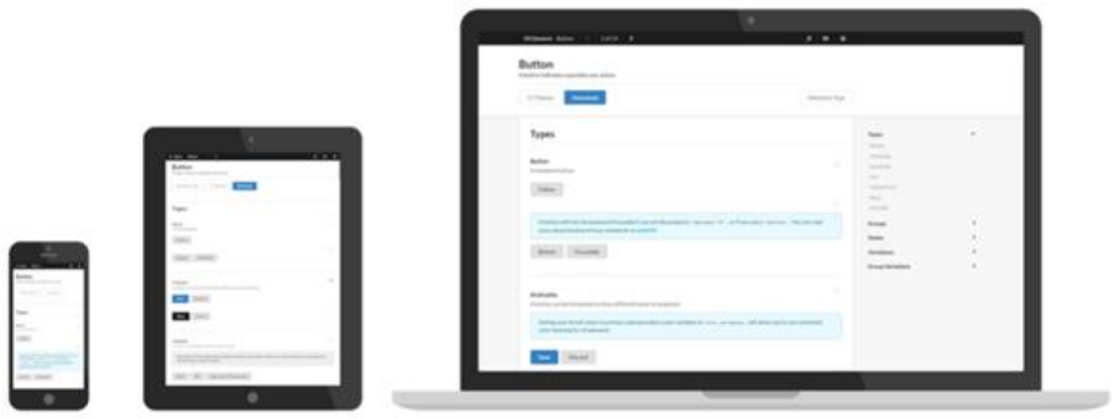
Tweet Star 43,404 English

2.4.2

Semantic UI

User Interface is the language of the web

Get Started | New in 2.4



Design Beautiful Websites Quicker

Semantic UI Starter

- Semantic css defines a large number of classes
- Your elements take on Semantic-UI styles by adopting specific classes
- All classes are preceded by “ui” to mark them as part of the framework

class =“ui container”

class =“ui segment”

class=“ui header”

class=“ui image”

class =“ui grid”

class “ui row”

class=“ui column”

First Steps…

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.css" type="text/css">
  <link rel="stylesheet" href="css/mycss.css" >
  <title>WIT Alternative Site</title>
</head>
<body>
...
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js"></script>
</body>
</html>
```

- We will link to the required Semantic UI files using Cloud Flare.

- Cloud Flare is a Content Delivery Network (CDN).

Container

http://semantic-ui.com/elements/container.html

When To Use

A container is an element designed to contain page elements to a reasonable maximum width based on the size of a user's screen. This is useful to couple with other UI elements like grid or menu to restrict their width to a reasonable size for display.

Container Sizes

Containers are designed to responsively adjust their maximum width based on the size of the screen they are appearing.

	Mobile	Tablet	Small Monitor	Large Monitor
Width	100%	723px	933px	1127px
Gutter Size	1em	Fluid	Fluid	Fluid
Responsive Visibility	`mobile only`	`tablet only`	`small monitor only`	`large monitor only`
Device Width	below 768px	768px - 991px	992px - 1200px	above 1200px

Container

- As stated a container is a fixed width element that wraps your site's content. It remains a constant size and uses margin to center.

- Containers are the simplest way to center page content inside a grid.

```
<section class="ui container">
  Example of a fixed width container which is set using the <code>.ui.container</code> class.
</section>
```

Grid

- Grid systems enable you to create advanced layouts using rows and columns.

- Semantic UI uses a *16 column grid*, where each column is of equal width with spaces in between. These spaces are called *gutters*. The use of grids is by far the most popular method for dividing and utilizing space. Regardless of which framework you choose to develop with a solid understanding of the grid layout will allow you the most flexibility and control in your designs.

Grid - Columns

- If you would like to divide your content into 4 sections across the page it's as simple as dividing the 16 total columns by 4, which equals 4. This means you'll need to specify that each column takes up a width of four in order to have four columns across the page.

```
<section class="ui container">
   <div class="ui grid">
     <div class="four wide column">One</div>
     <div class="four wide column">Two</div>
     <div class="four wide column">Three</div>
     <div class="four wide column">Four</div>
  </div>
</section>
```

One	Two	Three	Four

Grid - Columns

- We could code the grid also as follows:

```
<section class="ui container">
    <div class="ui four column grid">
      <div class="ui column">One</div>
      <div class="ui column">Two</div>
      <div class="ui column">Three</div>
      <div class="ui column">Four</div>
  </div>
</section>
```

Grid - Columns

- Alternatively, we could code the grid also as follows:

```
<section class="ui container">
    <div class="ui equal width grid">
      <div class="ui column">One</div>
      <div class="ui column">Two</div>
      <div class="ui column">Three</div>
      <div class="ui column">Four</div>
  </div>
</section>
```

Grid - Rows

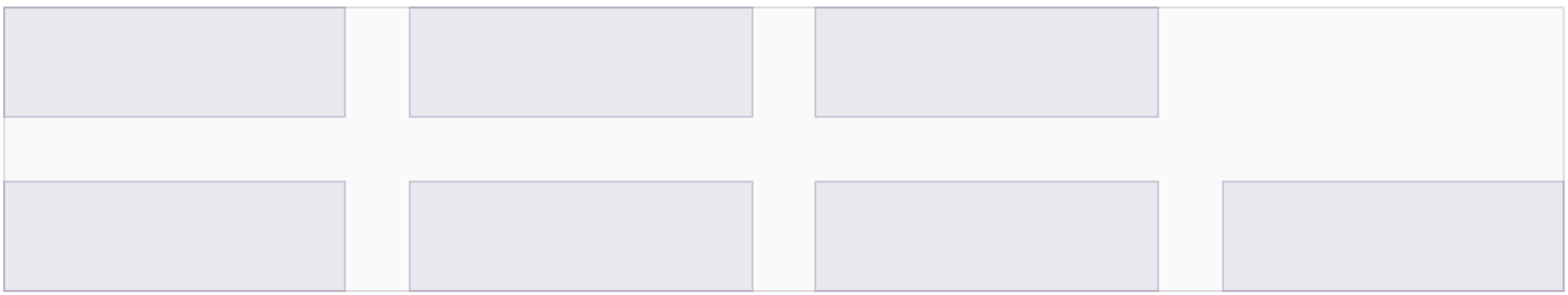
- **Rows** are implicitly rendered if required but new rows may be explicitly specified.

```
<section class="ui container">
   <div class="ui grid">
     <div class="four wide column">One</div>
     <div class="four wide column">Two</div>
     <div class="four wide column">Three</div>
     <div class="four wide column">Four</div>
     <div class="four wide column">Five</div>
     <div class="four wide column">Six</div>
     <div class="four wide column">Seven</div>
     <div class="four wide column">Eight</div>
  </div>
</section>
```

One	Two	Three	Four
Five	Six	Seven	Eight

Grid - Rows

Example



```
<div class="ui four column grid">
  <div class="row">
    <div class="column"></div>
    <div class="column"></div>
    <div class="column"></div>
  </div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
  <div class="column"></div>
</div>
```

Grid

- Grids may be nested:

Column A	Column B	Column C	Column D

- Coloured:

Frontend	Serverside	Databases

- Aligned and Floated:

Left floated right aligned column	Right floated left aligned column
Center aligned row	Center aligned row

Grid

- We need to design for all viewports so what looks readable and usable on a computer may not look the same on small devices such as mobile.
- There are various options available to change the *look* of a website depending on the viewport.

Grid

- A **stackable** grid will automatically stack rows to a single columns on mobile devices.

```
<section class="ui container">
<div class="ui stackable four column grid">
  <div class="column">Column A</div>
  <div class="column">Column B</div>
  <div class="column">Column C</div>
  <div class="column">Column D</div>
</div>
</section>
```

Column A
Column B
Column C
Column D

Grid

- You can **reverse** the order of columns or rows by device.

```
<section class="ui grid container">
<div class="ui mobile reversed tablet reversed equal width row">
 <div class="column">
  First
 </div>
 <div class="column">
  Second
 </div>
 <div class="column">
  Third
 </div>
</div>
</section>
```

First	Second	Third

Third	Second	First

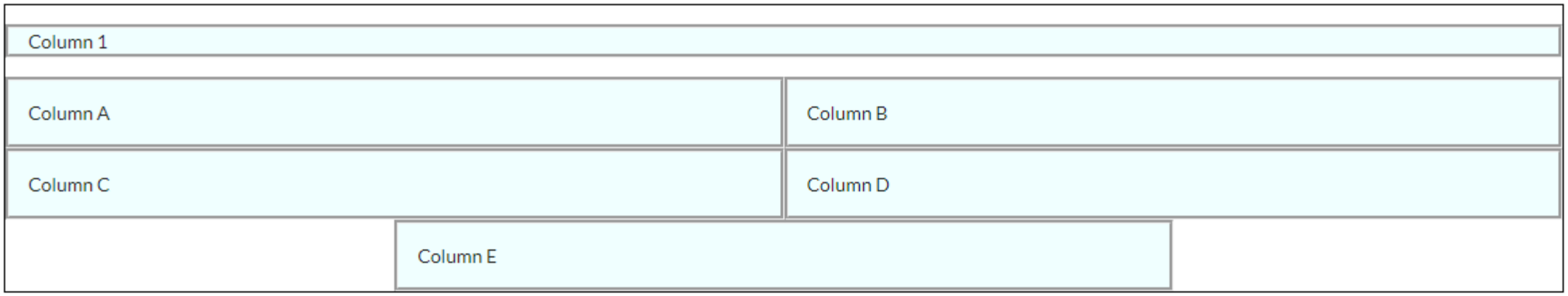
Grid

- You can also manually tweak device presentation by specifying **(x) wide device** or **device only columns** or **rows**.

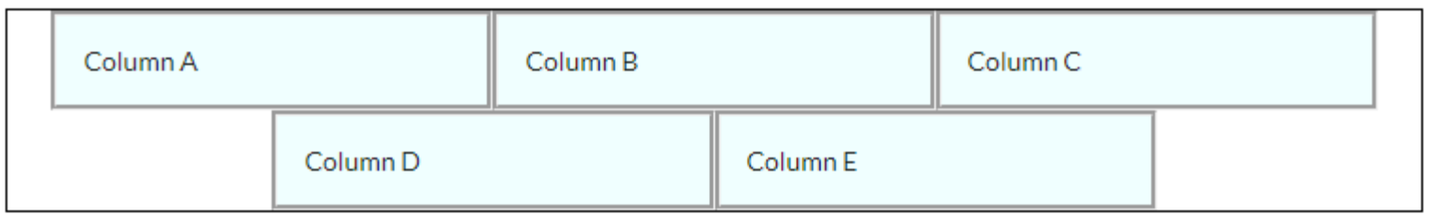
```
<section class="ui centered stackable grid container">
  <div class="computer only row">
    <div class="column">Column 1</div>
  </div>
  <div class="five wide tablet eight wide computer column">Column A</div>
  <div class="five wide tablet eight wide computer column">Column B</div>
  <div class="five wide tablet eight wide computer column">Column C</div>
  <div class="five wide tablet eight wide computer column">Column D</div>
  <div class="five wide tablet eight wide computer column">Column E</div>
</section>
```

Grid

- Computer viewports and larger:



- Tablet viewport:



Segment

- Use to group related content
- Give the enclosing articles/sections the class

class="ui segment":

```
<section class="ui container">
<div class="ui  three column grid segment">
 <div class="ui column">
        <h2>Frontend</h2>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        </p>
 </div>
...
</div>
</section>
```

Frontend

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.

Serverside

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.

Databases

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.

Segment

- Variations of the segment include:
  - Raised
  - Piled
  - Stacked

Programming

Learn a broad range of programming and problem solving skills, including exciting new platforms, software tools and languages. Use these skills to build apps for mobile, cloud and device based IoT applications. Evolve a porfolio of facinating aplications.

Data Science

At the heart of many IoT applications is data: measurements, events alarms and other information that must be relayed, stored and ultimately turned into knowledge. Learn the fundamentals of modern approaches to data in this strand.

Devices

The 'Things' we connect to are often physical devices. These can range from simple temperature sensors to sophisticated control systems like traffic lights or cameras. C
world is the subject of this strand.

Networks

This strand will explore modern networks and cloud technolc
categories of computer systems from simple controlers to sir
workstations.

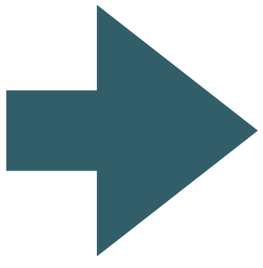
Project

Building exciting IoT projects in every semester of the progra
from the other strands and enable you to build a comprehens
and services.

Mathematics

Introduce foundation concepts for many of the more applied
mathematical techniques in a modern context and apply core

Grid Example



Programming

Learn a broad range of programming and problem solving skills, including exciting new platforms, software tools and languages. Use these skills to build apps for mobile, cloud and device based IoT applications. Evolve a porfolio of facinating aplications.

Data Science

At the heart of many IoT applications is data: measurements, events alarms and other information that must be relayed, stored and ultimately turned into knowledge. Learn the fundamentals of modern approaches to data in this strand.

Devices

The 'Things' we connect to are often physical devices. These can range from simple temperature sensors to sophisticated control systems like traffic lights or cameras. Connecting to and interacting with the physical world is the subject of this strand.

Networks

This strand will explore modern networks and cloud technology. Be able to configure, network and manage all categories of computer systems from simple controlers to single board board computers, mobiles and full workstations.

Project

Building exciting IoT projects in every semester of the programme. Your projects will combine skills acquired from the other strands and enable you to build a comprehensive an compelling portfolio of IoT applications and services.

Mathematics

Introduce foundation concepts for many of the more applied concepts in the other Strands. Learn mathematical techniques in a modern context and apply core principles in new an interesting ways.

Programming

Learn a broad range of programming and problem solving skills, including exciting new platforms, software tools and languages. Use these skills to build apps for mobile, cloud and device based IoT applications. Evolve a porfolio of facinating aplications.

Data Science

At the heart of many IoT applications is data: measurements, events alarms and other information that must be relayed, stored and ultimately turned into knowledge. Learn the fundamentals of modern approaches to data in this strand.

Devices

The 'Things' we connect to are often physical devices. These can range from simple temperature sensors to sophisticated co systems like traffic lights or cameras. Connecting to and interacting with the physical world is the subject of this s

Networks

This strand will explore modern networks and cloud technology. Be able to configure, network and manage all categories of computer systems from simple controlers to single board board computers, mobiles and full workstations.

Project

Building exciting IoT projects in every semester of the programme. Your projects will combine skills acquired from the other strands and enable you to build a comprehensive an compelling portfolio of IoT applications and services.

Mathematics

Introduce foundation conce for many of the more applie concepts in the other Stran Learn mathematical techniq a modern context and apply principles in new an interest ways.

- Two Rows
  - Row 1 - three columns
  - Row 2 - three columns

```
<section class="ui grid segment">

  <section class="ui three column row">
    <article class="column">
       ... code for the first column
    </article>
    <article class="column">
       ... code for the seconod column
    </article>
    <article class="column">
       ... code for the third column
    </article>
  </section>

  <section class="ui three column row">
    <article class="column">
       ... code for the first column
    </article>
    <article class="column">
       ... code for the second column
    </article>
    <article class="column">
       ... code for the third column
    </article>
  </section>

</section>
```

Image Element

- The **ui image** class can be added to the <img> tag to add more options for images.

```
<div class = "column">
  <img src="images/bird1.jpg" class="ui middle aligned tiny image" alt="Sample image">
  <span>Small</span>
</div>
<div class = "column">
 <img src="images/bird2.jpg" img class="ui middle aligned rounded image" alt="Sample image">
 <span>Rounded</span>
</div>
<div class = "column">
 <img src="images/bird2.jpg" class="ui middle aligned circular image" alt="Sample image">
 <span>Circular</span>
</div>
```


Small

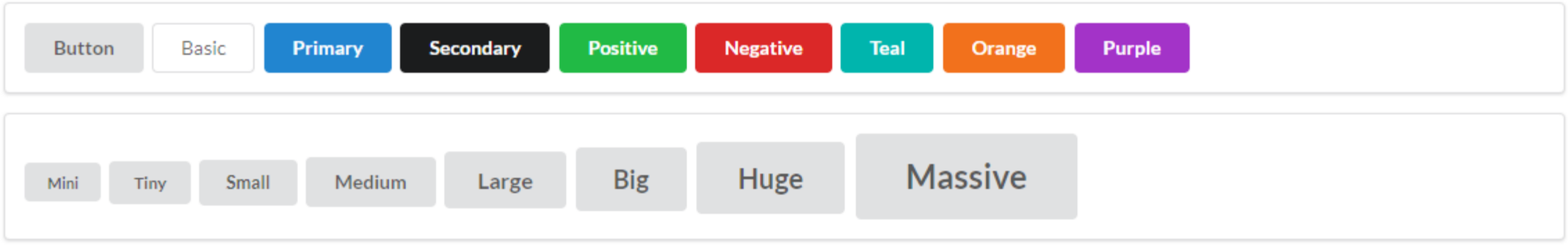

Rounded


Circular

Button Element

- Likewise, the **ui button** class can be added to the <button> tag to add more options for buttons.

- Options include:
  - Size
  - Type
  - Colour
  - Inverted



Elements

- Information about Semantic UI elements can be found at:

  https://semantic-ui.com