

App Dev

Stefano Balietti

Center for European Social Science Research at Mannheim University (MZES)
Alfred-Weber Institute of Economics at Heidelberg University

@balietti | stefanobalietti.com | @nodegameorg | nodegame.org



Building Digital Skills: 5-14 May 2021, University of Luzern



Express



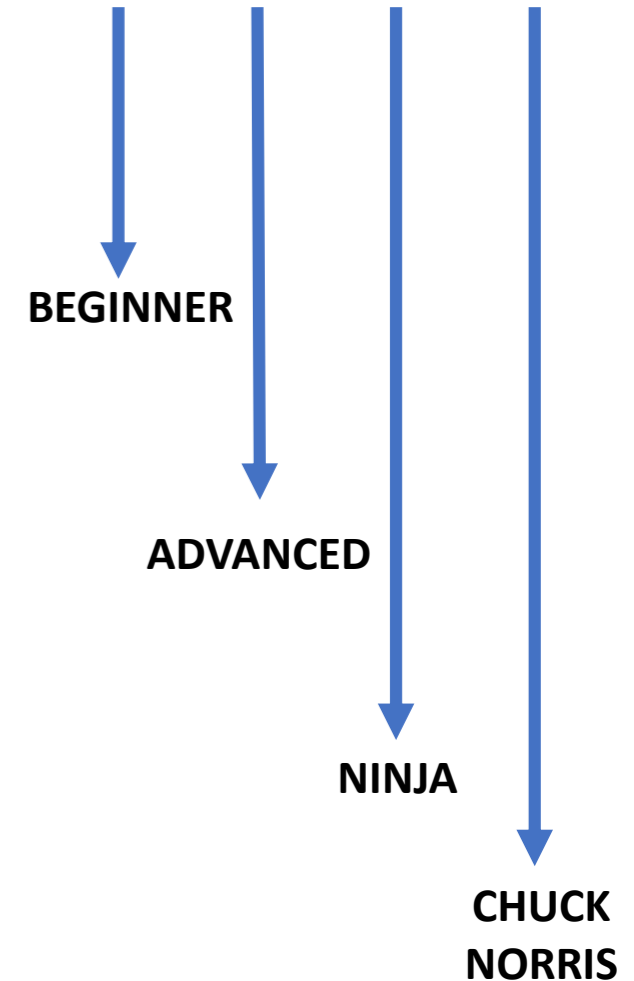
What Is This Course About?

- Leverage knowledge acquired in course "**Intro to Programming for App Development**"
- The course expects some **knowledge of computer programming** and a basic understanding of **front-end web development ...**
- *but we will recap the core concepts.*
- The goal is to gain a fine understanding of the **full web stack** to create **web-based apps** that can run on **multiple devices**
- The **learning speed and materials will be adapted** to the level of the participants

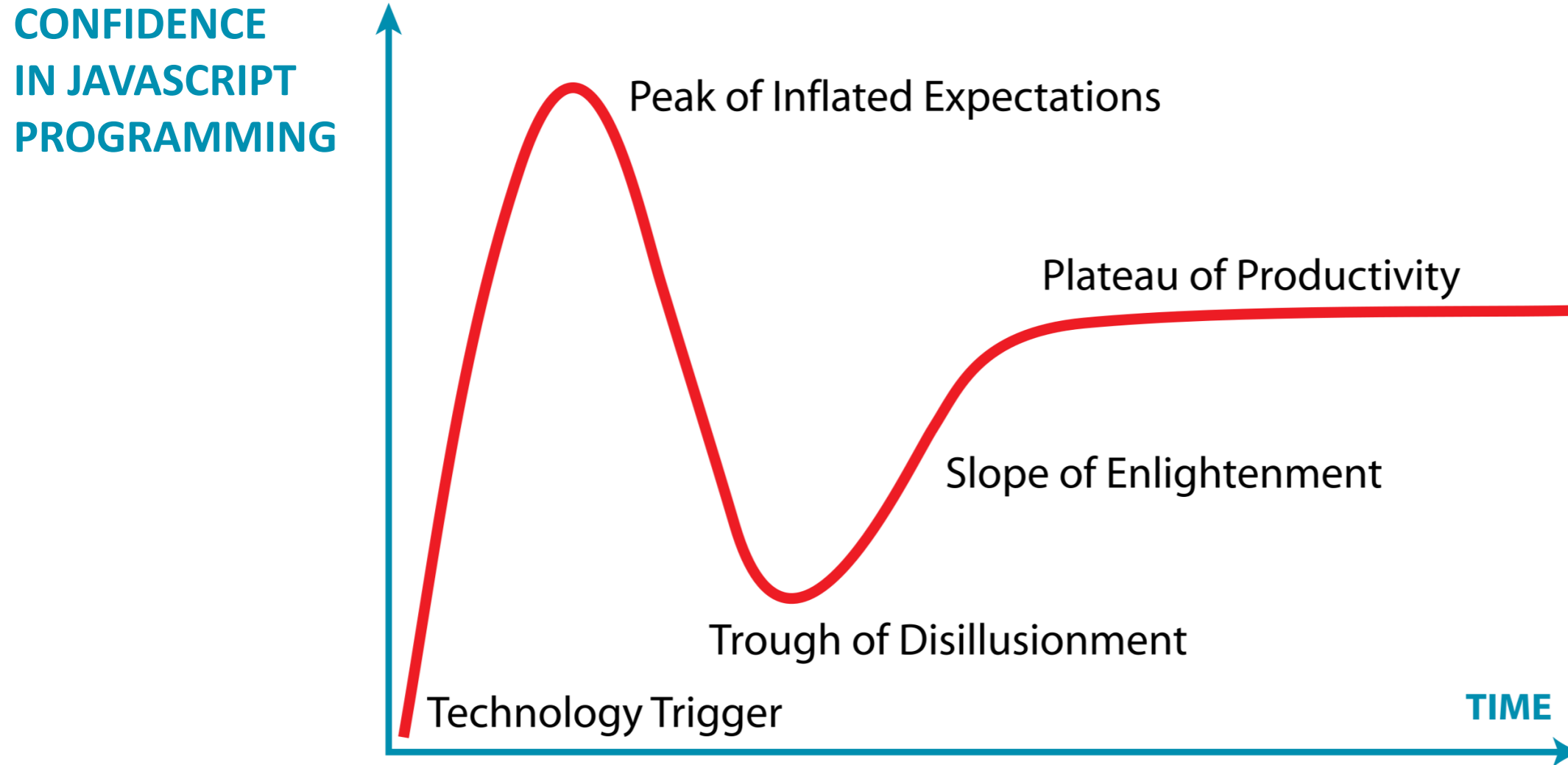
Tentative Schedule

Date	Module	Goals
Nov 11	Recap, Async code, Frontend Frameworks	Environment check, recap of intro course (JS, Browser, CSS), NPM and Node, Asynchronous Code (callbacks, promises, async/await, listeners, fetch, axios, REST APIs), jQuery
Nov 12	More Frontend frameworks, Intro to Express, and Nginx	Bootstrap v5, Flex, and Grid, Express, Nginx Running a local HTTP server
Nov 18	Securing Express, Taming Bots, Hosting, Let's Encrypt, MongoDB, nodeGame	Debugging Backend code, Running MongoDB in the cloud and locally, Choosing an host, Encryption, nodeGame, Honeypots, Captchas
Nov 19	Chrome Extensions, PWA, Mobile Development	Mobile First, Overview of Single-page frameworks, Chrome Extensions, PWA, Ionic, Debugging Mobile

What is a realistic learning goal?



Learning Curve



Adapted from: https://en.wikipedia.org/wiki/Hype_cycle

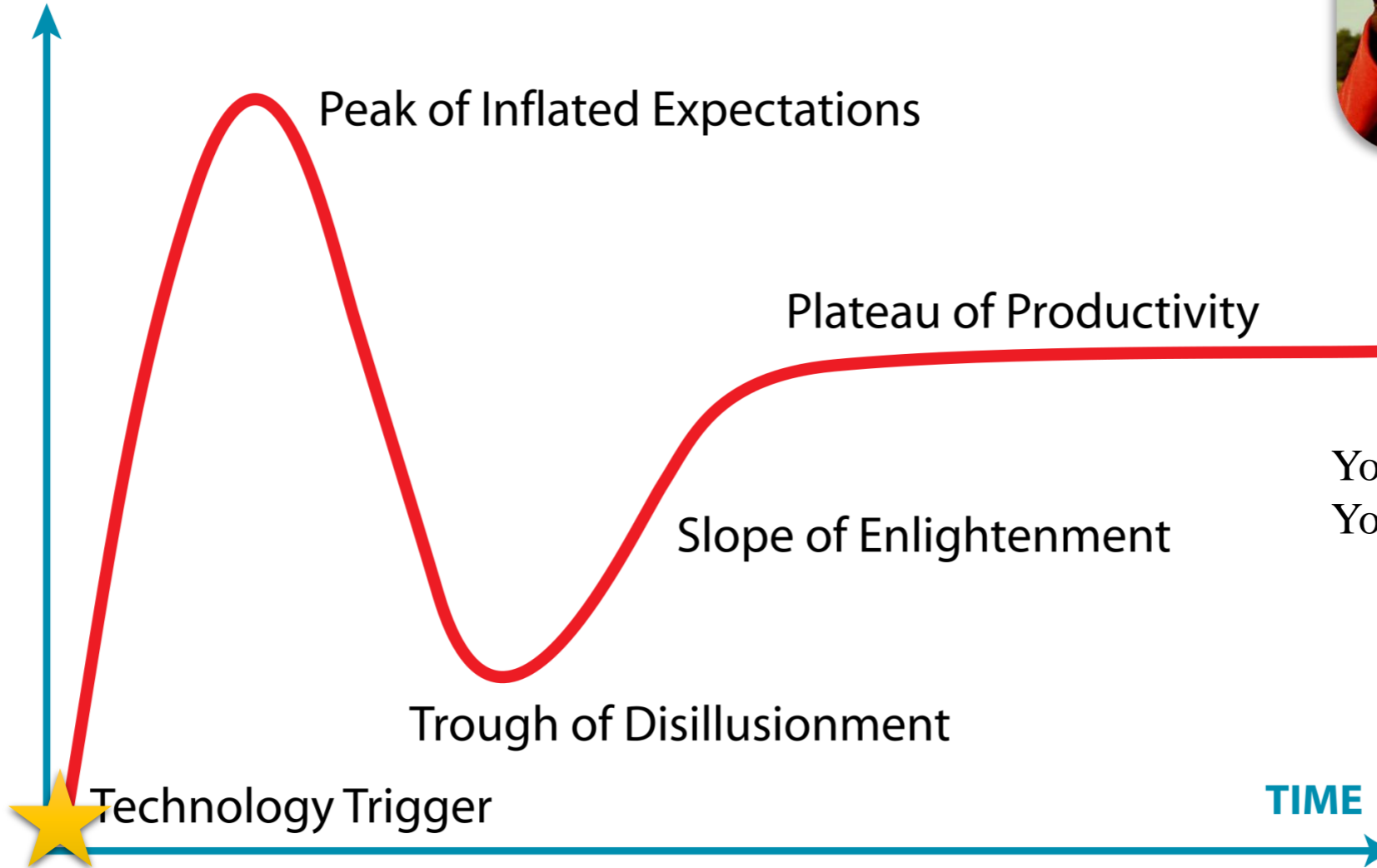
Learning Curve



Dr. Jimmy Cliff is a Jamaican ska & reggae musician, singer and actor.

You can get it if you really want
You can get it if you really want
But you must try,
try and try, try and try
You'll succeed at last

CONFIDENCE
IN JAVASCRIPT
PROGRAMMING

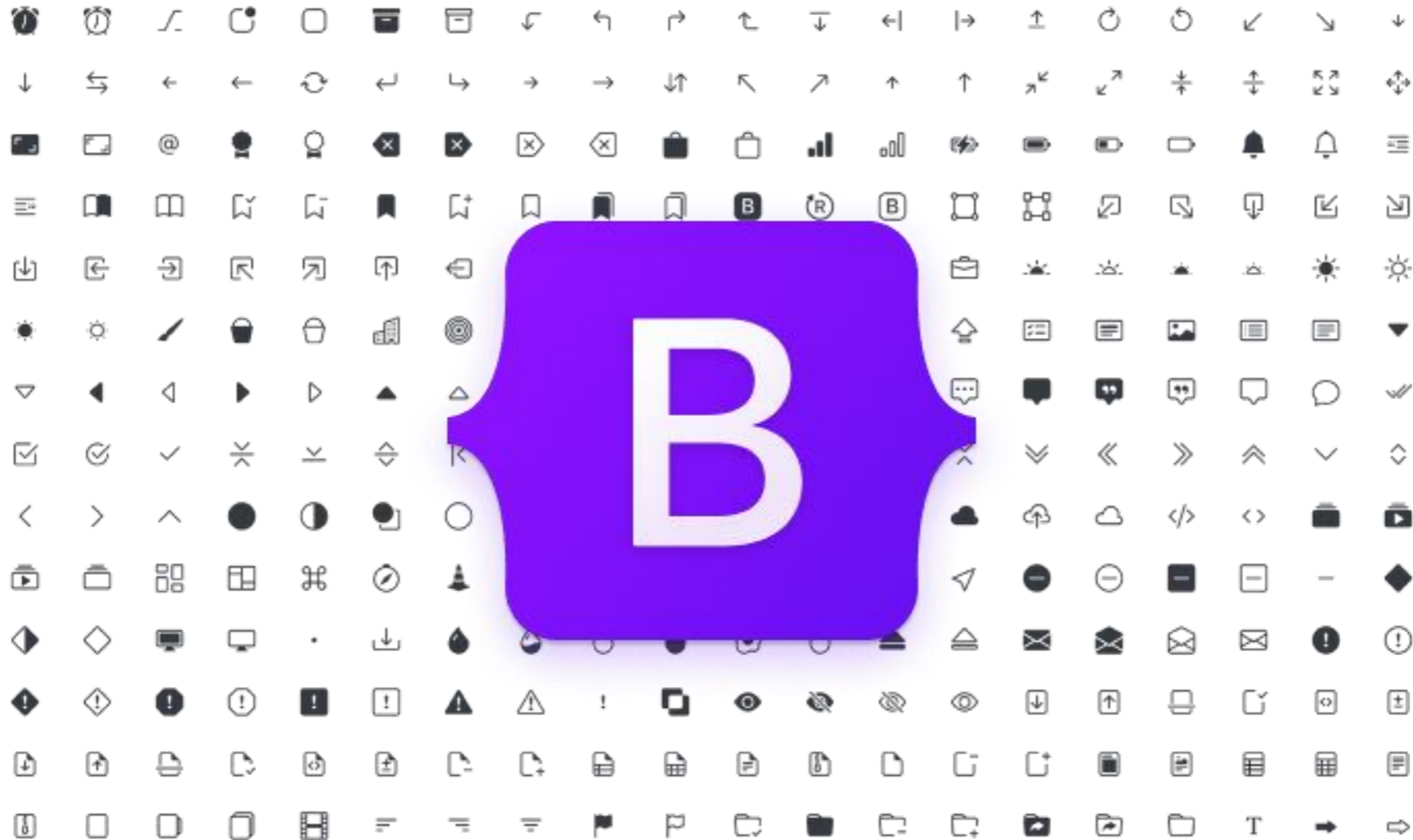


Adapted from: https://en.wikipedia.org/wiki/Hype_cycle

Module 4: References

- <https://jquery.com/>
- <https://github.com/jquery/jquery>
- <https://www.w3schools.com/jquery/>
- <https://polyfill.io/>

Module 5: Bootstrap



[Image source](#)

Module 5: Bootstrap

Learning Goals

- Improve the **old school** web app to cheer us up in difficult times.
- Include bootstrap CSS and JS in your app
- Understand grid layouts
- Understand and use Bootstrap components

Module 5: Bootstrap

- **Free** and **open-source mobile-first** front-end web development framework.
- With over **150k stars** is in the top-ten of most starred GitHub projects
- Bootstrap was created at Twitter in mid-2010 by @mdo and @fat
- Prior to being an open-sourced framework, Bootstrap was known as **Twitter Blueprint**
- Originally released on August 2011 it has had over twenty releases

Module 5: Bootstrap

- Current stable version is **v5**: replaces jQuery with vanilla JS
- Previous **v4**: uses jQuery for JS animations
- Great **documentation!**
- Gallery of **examples**
- **Themes** available (commercial)

Module 5: Bootstrap

- It contains CSS- and (optionally) JavaScript-based design templates for:
 - Typography
 - Spacing
 - Forms
 - Buttons
 - Navigation
 - Modal messages
 - ...

The Basic Idea

Adding attributes (mainly class and role) to HTML triggers special behavior and rendering

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

Link

```
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-light">Light</button>  
<button type="button" class="btn btn-dark">Dark</button>  
  
<button type="button" class="btn btn-link">Link</button>
```

Copy

The Basic Idea

Adding attributes (mainly class and role) to HTML triggers special behavior and rendering



```
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-light">Light</button>  
<button type="button" class="btn btn-dark">Dark</button>  
  
<button type="button" class="btn btn-link">Link</button>
```

Copy

The Basic Idea

Adding attributes (mainly class and role) to HTML triggers special behavior and rendering

```
<ul class="nav">  
  <li class="nav-item">  
    <a class="nav-link active" aria-current="page" href="#">Active</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link" href="#">Link</a>  
  </li>  
  <li class="nav-item">  
    <a class="nav-link disabled">Disabled</a>  
  </li>  
</ul>
```

Active Link Link Disabled

- [Active](#)
- [Link](#)
- [Link](#)
- Disabled

The Basic Idea

Adding attributes (mainly class and role) to HTML triggers special behavior and rendering

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link active" aria-current="page" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled">Disabled</a>
  </li>
</ul>
```

Active Link Link Disabled

- [Active](#)
- [Link](#)
- [Link](#)
- Disabled

A responsive site should adapt to the container



[Image source](#)

Grid System

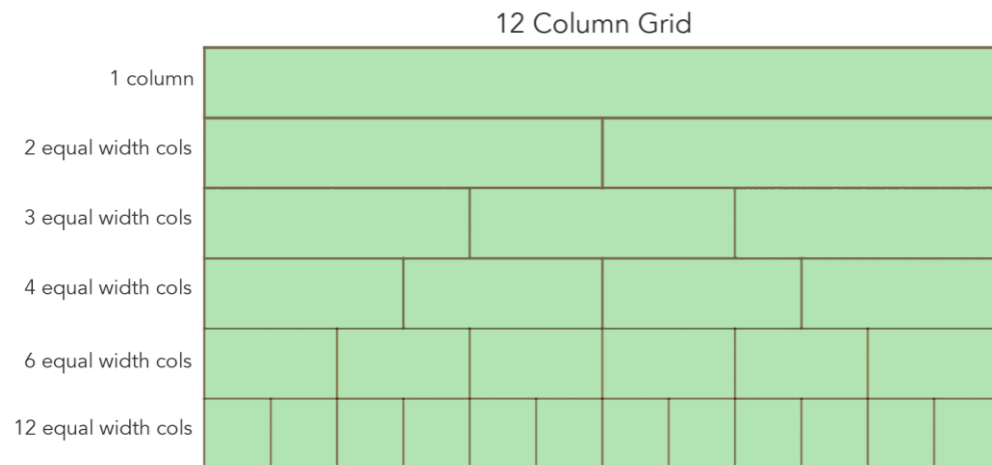


Grid System

- CSS Grid Layout (aka “Grid” or “CSS Grid”), is a two-dimensional grid-based layout
- Major improvement over [what we have used before](#) to position elements on page

Grid System

- CSS Grid Layout (aka “Grid” or “CSS Grid”), is a two-dimensional grid-based layout
- Major improvement over [what we have used before](#) to position elements on page
- Main idea:
 - divide the page in rows (all rows are inside a div **container**)
 - set the (fixed) number of columns per row (**in Bootstrap 12**)
 - in each row decide how many how large each column is
 - set breakpoints to change number of columns for every screen size



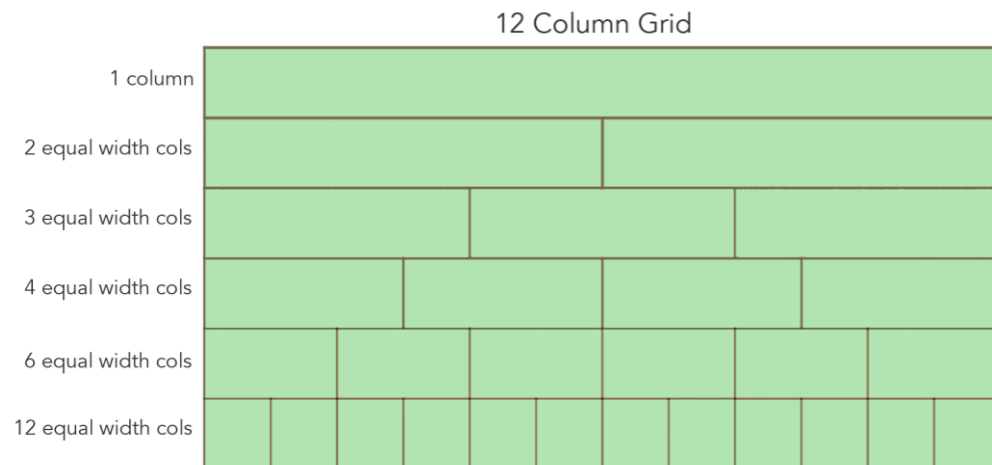
[Img source](#)



[Img source](#)

Grid System

- CSS Grid Layout (aka “Grid” or “CSS Grid”), is a two-dimensional grid-based layout
- Major improvement over [what we have used before](#) to position elements on page
- Main idea:
 - divide the page in rows (all rows are inside a div **container**)
 - set the (fixed) number of columns per row (**in Bootstrap 12**)
 - in each row decide how many how large each column is
 - set breakpoints to change number of columns for every screen size



[Img source](#)



[Img source](#)

Elements of the first bootstrap Grid

```
<div class="container">  
  <div class="row">  
    <div class="col">  
      Column  
    </div>  
    <div class="col">  
      Column  
    </div>  
    <div class="col">  
      Column  
    </div>  
  </div>  
</div>
```

A parent DIV with class "container"

One div of class "row"

A number of div of class "col"

OUTPUT

Column	Column	Column
--------	--------	--------

Elements of the first bootstrap Grid

```
<div class="container">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```

A parent DIV with class "container"

One div of class "row"

A number of div of class "col"

OUTPUT

Column	Column	Column
--------	--------	--------

Changing class "col" into class "col-x" specifies how many cells within a row of 12 cells that column will occupy (x is a number 1-12)

GRID A

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

GRID B

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-6">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-5">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

OUTPUT 1

1 of 2	2 of 2	
1 of 3	2 of 3	3 of 3

OUTPUT 2

1 of 3	2 of 3 (wider)	3 of 3
1 of 3	2 of 3 (wider)	3 of 3

Which grid is producing which output?
Why?

Bootstrap Grid Breakpoints

- Consider the following classes and break points associated to a given screen sizes

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container <i>max-width</i>	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	<i>.col-</i>	<i>.col-sm-</i>	<i>.col-md-</i>	<i>.col-lg-</i>	<i>.col-xl-</i>	<i>.col-xxl-</i>

If screen/container is larger than this,
then
apply the following rule to size the column

Bootstrap Grid Breakpoints

- Consider the following classes and break points associated to a given screen sizes

	xs	sm	md	lg	xl	xxl
	<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px
Container <i>max-width</i>	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	<i>.col-</i>	<i>.col-sm-</i>	<i>.col-md-</i>	<i>.col-lg-</i>	<i>.col-xl-</i>	<i>.col-xxl-</i>

If screen/container is larger than this,
then
apply the following rule to size the column

Rule is for instance:
col-sm-4: take for cells if larger than 576px

Bootstrap Grid Breakpoints

- Consider the following classes and break points associated to a given screen sizes

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container <i>max-width</i>	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	<i>.col-</i>	<i>.col-sm-</i>	<i>.col-md-</i>	<i>.col-lg-</i>	<i>.col-xl-</i>	<i>.col-xxl-</i>

If screen/container is larger than this,
then
apply the following rule to size the column

By combining multiple directives together
we gain a tight control on how to display
and size cells of the grid

Bootstrap Grid Breakpoints: Mix and Match

By combining multiple directives together we gain a tight control on how to display and size cells of the grid

```
<div class="container">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

Module 5: Resources

- <https://getbootstrap.com/>
- <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
- <https://github.com/twbs/bootstrap>
- <https://www.ipraxa.com/blog/bootstrap-5-whats-new-whats-changed/>
- <https://gridbyexample.com/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://alistapart.com/article/the-story-of-css-grid-from-its-creators/>
- <http://grid-layout.com/history.html>