

Web Dev Crash Course

Powered By:  GDG Erbil



Welcome to the workshop

What's to expect?

- Get the big picture.
- Know about what happens in the world.
- See if Web Development interests you.
- The first step of the 1000 mile journey.

What is Web Development?

ABC of Web Development

- Web (or World Wide Web) is created by Tim Berners-Lee
- Web is set of documents and resources identified by a URI
- ... and referenced by a Hyperlink
- Web is not equal to Internet

ABC of Web Development

- HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) is the underlying protocol of Web.
- A protocol means a set of rules between 2 parties.
- HTML (**H**yper**T**ext **M**arkup **L**anguage) is the language of web pages.
- A Markup Language means it only describes a structure of something (i.e. Web Page).
- HTML5 has more focus on Web Semantics.
- An example about Semantics? Google “Kitkat Calories”.

Web Development 10 years ago...

- HTML
- CSS
- JS

Web Development nowadays

- HTML, CSS, JS
- SASS, Stylus, LESS (CSS Preprocessors)
- React, Vue, Angular (Front-end frameworks)
- WebPack, Browserify, Parcel (Bundlers)
- Gulp, Grunt (Task Runners)
- NPM, Node
- ES6, ES2015
- Web Assembly
- ... etc

Let's `git` started!

Demo: Initialize a project with a Repository

Get started with HTML: The <markup language>

Demo: <Tags>

Demo: Structuring the Todo List

Give it some style with ...

...unsurprisingly with CSS

CSS stands for *Cascading Style Sheet*

The building block of CSS are
Rules

Rules are: selector and block
properties

CSS Rules



```
type#id.class, #another + .sibling {  
  prop-name: value;  
}
```

Demo: CSS Rules










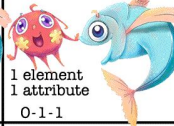
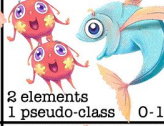
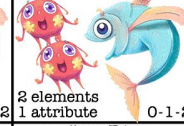

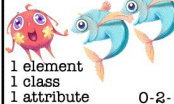


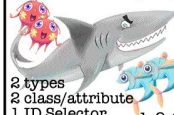



A Style Sheet that **cascade!**

It cascades from *most specific rule to the least specific*, from *bottom to top*

CSS Specificity

CSS SPECIFISHITY

WITH PLANKTON, FISH AND SHARKS

<p>*</p>  <p>universal selector 0-0-0</p>	<p>div</p>  <p>1 element 0-0-1</p>	<p>li > ul</p>  <p>2 elements 0-0-2</p>	<p>body div ...ul li p a</p>  <p>12 elements 0-0-12</p>
<p>.myClass</p>  <p>1 class 0-1-0</p>	<p>*.myClass</p>  <p>1 universal selector 1 class 0-1-0</p>	<p>[type=checkbox]</p>  <p>1 attribute selector 0-1-0</p>	<p>:only-of-type</p>  <p>1 pseudo-class 0-1-0</p>
<p>li.myClass</p>  <p>1 element 1 class 0-1-1</p>	<p>li[attr]</p>  <p>1 element 1 attribute 0-1-1</p>	<p>li:nth-of-type(3n)~li</p>  <p>2 elements 1 pseudo-class 0-1-2</p>	<p>form input[type=email]</p>  <p>2 elements 1 attribute 0-1-2</p>
<p>li.class:nth-of-type(3n)</p>  <p>1 element 1 class 1 pseudo-class 0-2-1</p>	<p>input[type]:not(.class)</p>  <p>1 element 1 class 1 attribute 0-2-1</p>	<p>olnth-child(4n)chk[type]...</p>  <p>10 class/attribute/pseudo-classes 0-10-0</p>	<p>#myDiv</p>  <p>1 ID Selector 1-0-0</p>
<p>#myDiv li.class a[href]</p>  <p>2 types 2 class/attribute 1 ID Selector 1-2-2</p>	<p>#divitis #myDiv a</p>  <p>2 ID Selectors 1 type selector 2-0-1</p>	<p>style=""</p>  <p>1 inline style 1-0-0-0</p>	<p>!important</p>  <p>1 important 1-0-0-0</p>

ESTELLE WEYL * @ESTELLEWV * WWW.STANDARDISTA.COM * 2014

X-0-0: The number of ID selectors

0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes

0-0-Z: The number of type selectors and pseudo-elements

*, +, >, ~ : Universal selector and combinators do not increase specificity

:not(x): Negation selector has no value. Argument increases specificity



Demo: Specificity

Demo: Give style to the Todo List

Demo: 2 Column layout

Why reinvent the wheel?!

Let's use bootstrap!

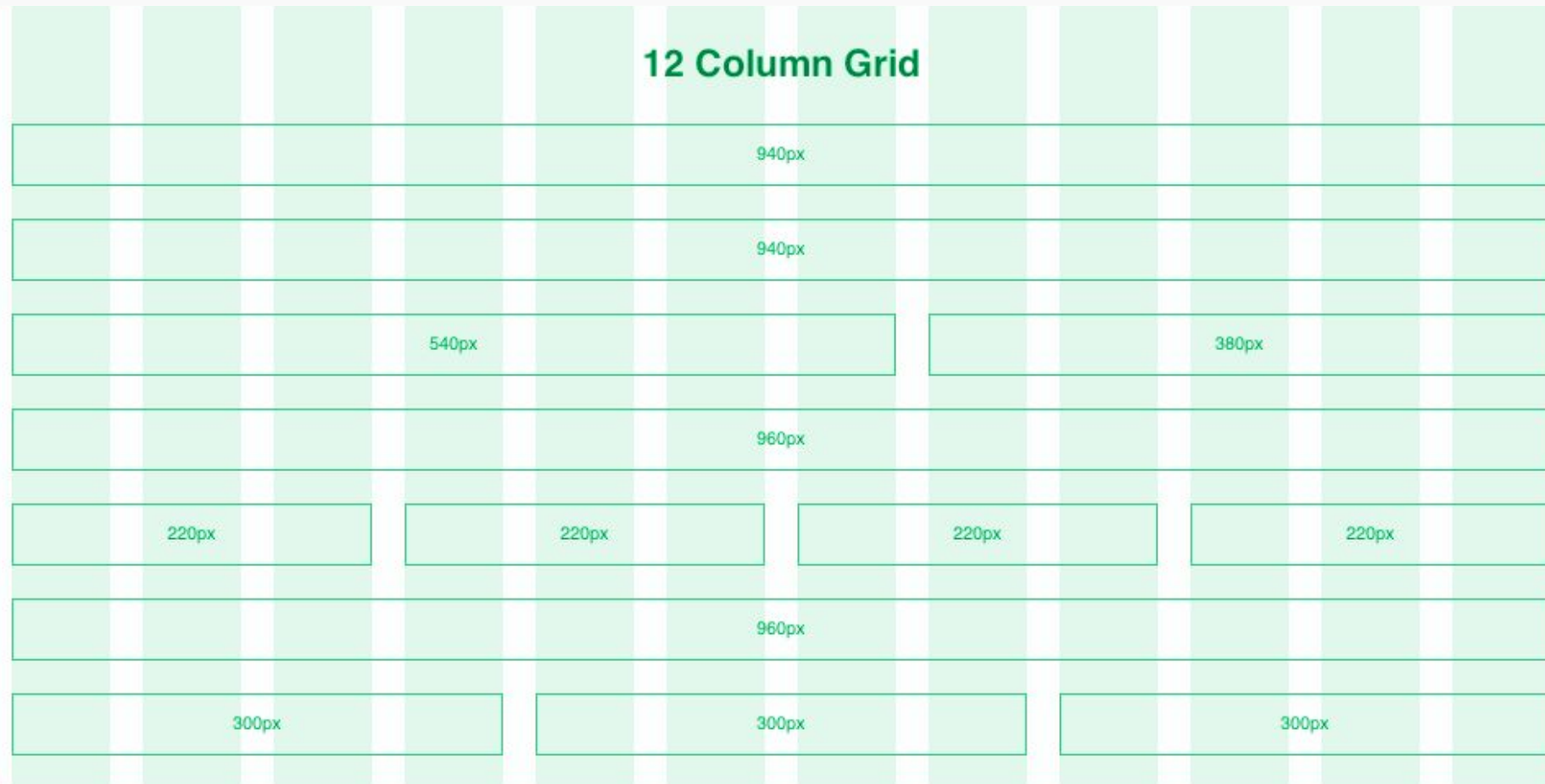
What is bootstrap?

What is bootstrap?

A framework for responsive design

Demo: Installing Bootstrap via npm

Bootstrap has 12-column grid system



Demo: Using bootstrap to give
colors and 2 columns layout

Let's make it actually work...

Intro to JavaScript

JavaScript is a programming language...

It's not related to Java and it's not
“scripting language”

It can do everything!

It can do everything!

“Any application that can be written in JavaScript, will eventually be written in JavaScript” — Jeff Atwood’s Law

Show me some code!

Variables and Constants

- You may declare variables with `let`
- And constants with `const`
- The type of the variable will be set on runtime (Dynamic typing)
- And also all types can be mixed up (Weakly Typing)



```
let x = 10; // A number
let s = 'String' // A string
let a = [] // An array

const b = {} // An object

const r = x + s + a + b; // Works!!!
```

Demo: Variables

Flow control & Iterations

- If statements are used to control the program flow
- Loops to repeat block of code multiple times
- For...of and for...in are introduced in ES6 to loop through iterables (i.e. Arrays)



```
if (truthy) {  
  // Do this  
} else if (truthy) {  
  // Do that  
} else {  
  // Or else  
}  
  
for (let i = 0; i < MAX; i++) {  
  // Do statements  
}  
  
while (truthy) {  
  // Statements  
}  
  
do {  
  // Statements  
} while (truthy);
```



```
const arr = [1,2,3,4];  
  
for (const x in arr) {  
  // Do statements  
}  
  
for (const x of arr) {  
  // Do statements  
}
```

Demo: Flow controls

Objects

- Objects are reference type values
- It's collection of `key: value` pairs separated with comma.
- It's also in format called JSON
- You may nest them inside each other.



```
const obj = {  
  name: 'Foo',  
  age: 20,  
  address: {  
    street: 'Foobar st',  
    city: 'Baz'  
  }  
}
```

`obj.name` // DOT notation

`obj['name']` // Bracket notation

Demo: Objects

Arrays

- They are actually *array-like* objects
- The key is index, and the value is the value in that key.
- Has method just like any object, like `push()`, `pop()`, `unshift()`, `splice()`, `slice()`, `concat()`, `map()`, `filter()`, `reduce()`, `forEach()`



```
const arr = [1, 2, 3];  
  
arr.push(4); // arr = [1,2,3,4]  
  
arr.map(function(x) {  
  return x * x;  
}) // [1,4,9,16]  
// OR  
arr.map(x => x * x) // [1,4,9,16]  
  
arr.filter(x => x % 2 == 0) // [2,4]  
  
arr.reduce( (prev, current) => prev + current) // 10
```

Demo: Arrays

Functions

- Block of code with name
- Are first-class citizen
- It has its own scope
- You may call it with any number of arguments
- If you don't pass arguments the parameter value will be `undefined`
- It bind `this` to the object that call it, if non it's `undefined` (in strict mode)
- Bound using `apply()`, `call()`, `bind()`
- Arrow function has no arguments and it's context can't be bound to any object



```
const func = function(param1, param2, param3...) {  
  // your logic  
  return // you may return value or undefined  
}
```



```
const func = (param1, param2, param3...) => {  
  // your logic  
  return // you may return value or undefined  
}  
// or  
const func = (param1, param2, param3...) => 0
```

Demo: Functions

Classes

- They are PascalCased (convention)
- Class is just syntactic sugar over functions.
- Functions has `prototype` property
- It's the original copy of the created objects.
- The objects created with `new` has `__proto__` property that points to `prototype` of the function
- Arrow functions cannot be used as constructors or methods

```
class Person
{
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  getName() {
    return this.name;
  }
}

// OR
const Person = function Person(name, age) {
  this.name = name;
  this.age = age;
}

Person.prototype.getName = function() {
  return this.name;
}

const p = new Person('Jon Doe', 20);
```

Demo: Classes

A 10 minutes break...

DOM

<Document>-{Object} Model

It's all about document object

You may createElement() ,
append() ,
querySelector() ,
remove() , **or listen to them**

Demo: Make Todo work

Introduction to React

React is a UI library

It works with Virtual DOM

It does only one thing, and it does
it well!

Building blocks of React are
Components

Components are simply a
function!

Demo: `React.createElement()`

Why not use Markup? (aka JSX)

What do we mean markup?

Demo: JSX

Demo: Build our todo with React!

Thank you!

Any questions?

Resources

- <https://pluralsight.com/courses/what-is-programming>
- <https://scrimba.com/>
- <https://specifishity.com/>
- <https://getbootstrap.com/>
- <https://scrimba.com/g/glearnreact>
- <https://scrimba.com/g/ges6>
- <https://egghead.io/courses/the-beginner-s-guide-to-react>

Conclusion