


Workshop Preperation

- Local install of WordPress 4.7+
- Code Editor (I am using Atom)
- Node and NPM Installed (node -v and npm -v)
- <https://github.com/zgordon/a-day-of-javascript>

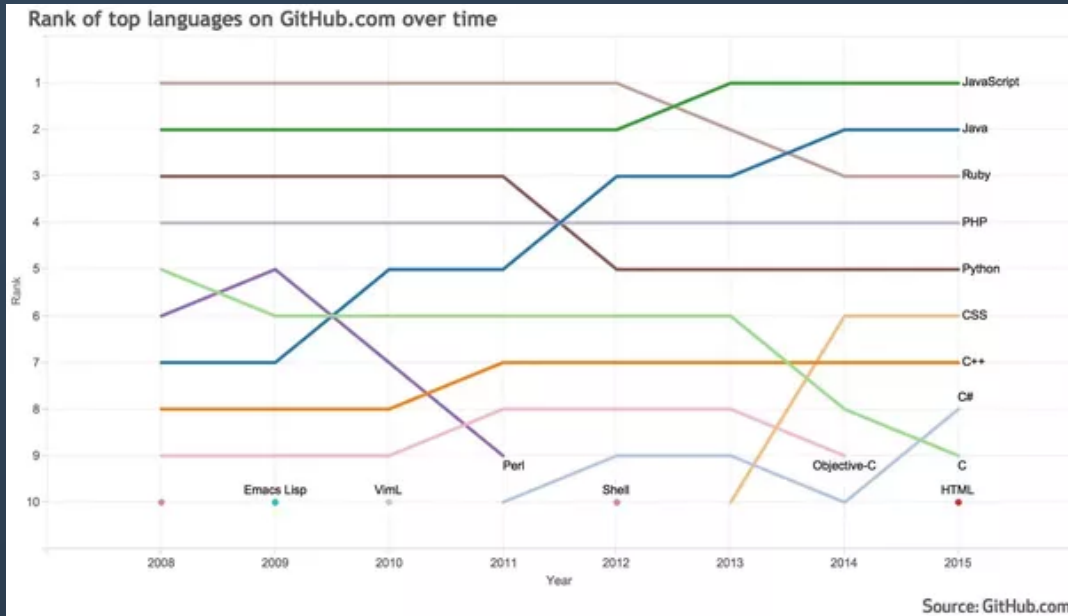
Welcome To

A Day of JavaScript

with Zac Gordon @zgordon

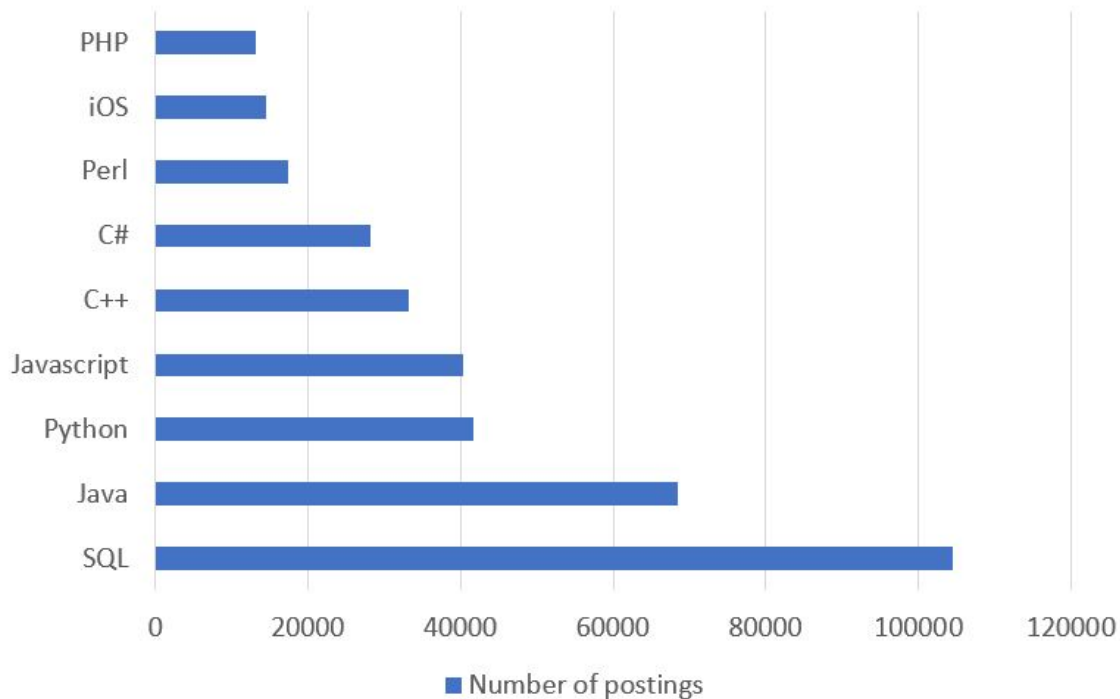
The background is a solid orange color. On the left side, there are several abstract, overlapping shapes in a slightly darker shade of orange. These shapes include a large circle, a smaller circle, and two curved, arc-like segments that suggest a stylized letter 'G' or 'C'.

Oh yes,
JavaScript
Is Important



JavaScript #1

Number of Indeed Job Postings by Programming Language



\$72,500

glassdoor.com

WordPress Developers Live in a Bubble

Most Developers Live in a Bubble

**API + JS =
Bigger Bubble**

What **Do You Need to Know?**

Wait, Do You Know Programming?

1. The DOM

1. The DOM

2. Events

1. The DOM

2. Events

3. JSON & HTTP

- 1. The DOM**
- 2. Events**
- 3. JSON & HTTP**

These 3 Things
Let us do a lot

Use WP API

Use WP API

Build UI/Page w JS

Use WP API

Build UI/Page w JS

Track User Actions

Use WP API

Build UI/Page w JS

Track User Actions

Do Whatever *

Use WP API JSON/HTTP
Build UI/Page w JS DOM
Track User Actions Events
Do Whatever JavaScript*

* *Programming Review*

1. The DOM

2. Events

3. JSON & HTTP

* *Development Tools*

I Love JavaScript
And have for a while

javascriptforwp.com/courses
javascriptforwp.com/blog

Group Up **It's Quiz Time**

Question 1

What is a **variable**?

Question 2

What is an **array**?

Question 3

What is a **function**?

Question 4

What is an **object**?

Question 5

What is a **loop**?

Question 6

**What is a
conditional?**

Question 7

**How do you include
JavaScript in HTML?**

Variable

A container for storing values

```
var username = 'zgordon';  
let username = 'zgordon';  
const siteURL = 'https://site.com';
```

Array

A collection of values

```
let postIds = [1,2,3,5];  
let usernames = [  
    'zgordon',  
    'admin'  
];
```

Functions

Let us write, call and reuse blocks of code

```
function getPosts() {  
    let posts = apiMagic... // Will learn  
    return posts;  
}  
getPosts();
```

Function Parameters

Can pass data into functions

```
function getUser( userId ) {  
    let user= apiMagic... // Will learn  
    return user;  
}  
getUsers( 1 );
```

Objects

Container with **properties** (values) and **methods** (functions).

```
let post = {  
  id: 1,  
  title: 'Hello Object',  
  displayTitle: function() {  
    console.log( this.title );  
  }  
}
```

```
post.id;  
post.title;  
post.displayTitle();
```

Loops

Let us perform an action on a collection of items.

```
var postIds = [ 1, 7, 14, 34, 88, 117 ];  
  
for ( let i = 0, max = postIds.length; i < max; i++ ) {  
  
    console.log( 'Display post #' + postIds[i] );  
  
}
```

For Of Loop

Lets us loop through an array

```
let postIds = [ 1, 2, 34, 55, 77 ];
```

```
for( let id of postIds ) {  
    console.log( id );  
}
```

Conditional Statements

Tests to determine what code to run

```
var loggedIn = true;
```

```
if ( true === loggedIn ) {  
    console.log( 'Show dashboard' );  
} else {  
    console.log( 'Please login' );  
}
```


Including JavaScript

```
<html>  
  <head>  
  </head>  
  <body>  
    <script src="index.js"></script>  
  </body>  
</html>
```

Review - How'd you do?

1. Variable
2. Array
3. Function
4. Object (properties, methods)
5. Loops
6. Conditionals
7. Including JavaScript in HTML

BREAK

The DOM

1. Nodes
2. Document Object Methods
3. Traversal (parents, children, siblings)
4. Creating and Appending

The Document Object Model



**In the DOM,
Everything is a Node**

```
<!DOCTYPE html>
<html>
<head>
  <title>The DOM</title>
</head>
<body>
  <h1>Title</h1>
  <p>Lorem <a title="Learn more"
href="#">to the</a> ipsum</p>
</body>
</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>The DOM</title>
```

```
</head>
```

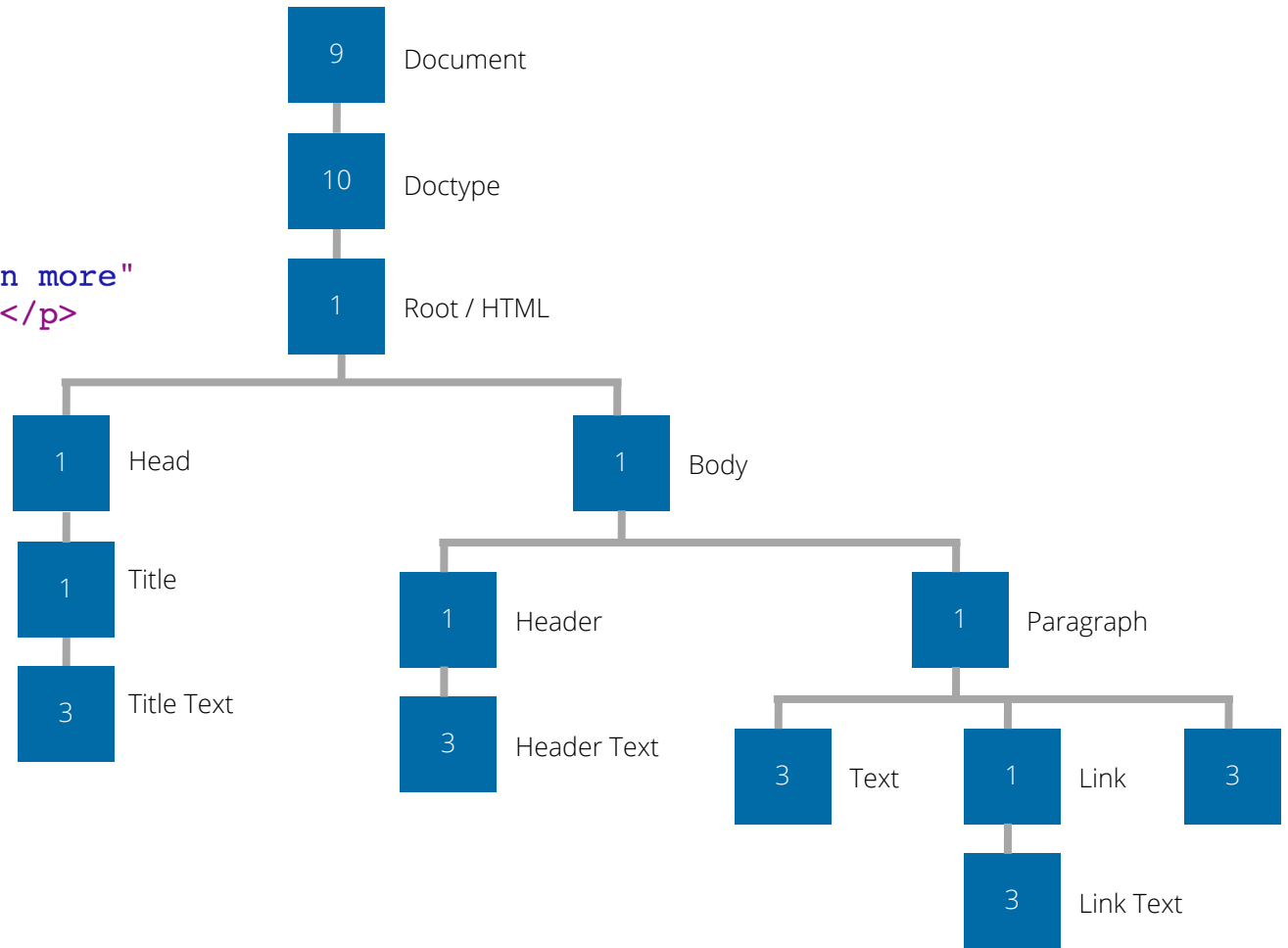
```
<body>
```

```
  <h1>Title</h1>
```

```
  <p>Lorem <a title="Learn more"  
href="#">to the</a> ipsum</p>
```

```
</body>
```

```
</html>
```



Document Object Methods

```
document.getElementById('main')
```

```
document.getElementsByTagName('a')
```

```
document.getElementsByClassName('post')
```

```
document.querySelector('.entry-title a')
```

```
document.querySelectorAll('.entry-title a')
```

DEMO

Getting Node Values

`el.innerText`

`el.innerHTML`

`el.id`

`el.href`

`input.value`

`el.classList.add('new')`

`el.classList.remove('new')`

`el.classList.toggle('new')`

DEMO

DOM Traversal

Any Node

- `.parentNode`
- `.childNodes`
- `.firstChild`
- `.lastChild`
- `.nextSibling`
- `.previousSibling`

Element Nodes

- `.parentElement`
- `.children`
- `.firstElementChild`
- `.lastElementChild`
- `.nextElementSibling`
- `.previousElementSibling`

DEMO

Node Creating, Appending

```
document.createElement('p')  
document.createTextNode('Text')  
document.createDocumentFragment()  
el.appendChild( child )  
parent.insertBefore( childEl, el )  
el.innerHTML = '<p>Text</p>';  
el.cloneNode( true )
```

DEMO

DOM Cheat Sheet

Document Methods

`document.getElementById()`

`document.getElementsByTagName()`

`document.getElementsByClassName()`

`document.querySelector()`

`document.querySelectorAll()`

Working with Attributes

`el.innerText` `el.id`

`el.innerHTML` `el.href`

`input.value` `el.attribute`

`el.classList.add('new')`

Traversal - Any Node Type

`.parentNode` `.lastChild`

`.childNodes` `.nextSibling`

`.firstChild` `.previousSibling`

Traversal - Element Nodes

`.parentElement` `.lastElementChild`

`.children` `.nextElementSibling`

`.firstElementChild` `.previousElementSibling`

Creating & Appending Nodes

`document.createElement('article')`

`el.appendChild(childEl)`

DOM Practice 01.01

Practice selecting following elements and logging to console

- Main content container
- First post title
- First post content
- All post titles

DOM Practice 01.02

1. Select the parent element for the first post title
2. Select the first post, then log out next sibling post
3. Select the main post container and log out it's children

DOM Practice 01.03

1. Create a post with markup using JavaScript
2. Append post to the list of other posts on the page

DOM Practice **

1. Write a function to append post content to the page
2. Create a post object with a title and content
3. Call the function to append posts and pass it the post you created

DOM Practice **

1. Create a form with preset values
2. Get the values from the form and log them in the console

A Word on jQuery

Events

1. Types of Events
2. Hooking Into Events (inline handlers, event listeners)
3. The Event Object

Types of Events in JavaScript

- Mouse events
- Keyboard events
- Form events
- Media events
- Drag and Drop events
- Window events
- Many more....

DEMO

Hooking Into Events

Inline Event Handler in HTML

```
<a href="#" onclick="console.log('Clicked');">Link</a>
```

Event Listener in JS

```
link.addEventListener( 'click', myFunction, false );
```

DEMO

The Event Object

```
link.addEventListener( 'click', sayHi, false );
```

```
function sayHi( e ) { // e is the event object
    console.log( e );
    console.log( e.type );
    console.log( e.target );
}
```

DEMO

Events Cheat Sheet

Inline Event Handler

```
<a href="#" onclick="//JS here;">Link</a>
```

Event Listener

```
el.addEventListener( 'event', function, false)
```

```
el.removeEventListener( 'event', function, false)
```

Event Types

load, offline, online, focus, blur, animationstart, animationend, transitionstart, transitionend, submit, resize, scroll, keypress, mousemove, mouseenter, mouseleave, click, dragstart, dragenter, dragend, dragover, drop, timeupdate, hashchange, more!

Event Object

e = arguments[0] | | events

e.target

e.type

Events Practice 02.01

- Write an inline click event on a link that logs out a message when clicked
- Add an event listener to a link that logs out a message when clicked

Events Practice 02.02

- Select all post title links on the page
- Add an event listener to each link that logs the title of that post when clicked

Events Practice 02.03

- Get the values from the form when submitted
- Append the new post values as a new post on the page
- Clear form values once post is loaded

Events in Different Libraries & Frameworks

JSON & HTTP

1. JSON Overview
2. HTTP Requests with Fetch
3. Pulling in WordPress API Content

JSON

```
{  
  "title": 2,  
  "title": "Hello JSON!",  
  "slug": "hello-json",  
  "content": "Some post content goes here."  
}
```

JSON

```
[  
  {  
    "title": "Hello JSON!",  
    "content": "Some post content goes here."  
  },  
  {  
    "title": "Hello API!",  
    "content": "More post content."  
  },  
]
```

DEMO

JSON Into JavaScript

```
let jsonString = '{ "id": 1, "title": "Hello JSON!" }',  
    post = JSON.parse( jsonString );  
  
console.log( post.title );
```


JavaScript Into JSON

```
let post = {  
    id: 2,  
    title: "Hello JSON!"  
},  
postJSON = JSON.stringify( post );  
  
console.log( postJSON );
```

DEMO

Making HTTP Get Requests with Fetch

```
fetch( 'https://site.com/wp-json/wp/v2/posts' )  
  .then( response => {  
    response.json().then( posts => {  
      console.log( posts );  
    });  
  })  
  .catch( err => {  
    console.log( err );  
  });
```

Making HTTP Post Requests with Fetch

```
fetch( 'https://site.com/wp-json/wp/v2/posts', {
  method: 'POST',
  body: JSON.stringify( post ),
  headers: new Headers({
    'Content-Type' : 'application/json',
    'Authorization' : 'Bearer ' + token
  }),
  credentials: 'include'
})
.then( function( response ) {
  if( 201 === response.status ) {
    console.log( 'Post Update' );
  }
});
```

DEMO

JSON & HTTP Cheat Sheet

JSON Example

```
[{ "title" : "Hello" }, { "title" : "JSON" }]
```

JS JSON Conversion

```
let posts = JSON.parse( postsJSON );  
let postJSON = JSON.stringify( posts );
```

Fetch with Promises

```
fetch( 'https://site.com/posts.json' )  
  .then( response => {  
    response.json().then( posts=> {  
      console.log( posts );  
    });  
  })  
  .catch( err => {  
    console.log( err );  
  });
```

JSON Practice 03.01

- Write a valid JSON file for a post
- Validate that JSON file online

JSON Practice 03.02

- Make a fetch call to your JSON file
- Display the post on the page

JSON Practice 03.03

- Make a fetch call to a WP site for posts (5 max)
- Display the posts on the page

WP API + JS in a Theme

1. Add the apidemo.zip theme to your WP site
2. Functions.php enqueue main.js with dependency of wp-api and jquery
3. Use new wp.api.collections.Posts() and .fetch to get posts from WP Site
4. Use JavaScript (and jQuery) to add the posts to the page

Review

A Day of JavaScript

A Day of JavaScript Review

1. Programming Review: variables, arrays, functions, objects, loops, conditional statements
2. The Document Object Model: