JavaScript for WordPress

Zac Gordon @zgordon

https://javascriptforwp.com

Getting Setup

- **Local install of WordPress:**

  DesktopServer - serverpress.com

  FakerPress - wordpress.org/plugins/fakerpress

- **Code Editor:**

  Atom*, Visual Studio Code*, PHPStorm / WebStorm

- **Example Files and Slides:**

  github.com/zgordon/wcmpls-2017

Workshop Overview

1. JavaScript Basics
2. The WordPress REST API
3. JavaScript & WP API in Themes

JS 101

Variable

A container for storing values (in memory)

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

Variable

A container for storing values (in memory)

```
let username = 'zgordon',
    twitter = '@zgordon',
    website = ‘zacgordon.com';
```

Array

A collection of values

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

Array

A collection of values (zero indexed)

```
let postIds = [ 1, 2, 3, 5 ];

postIds[ 0 ] // Equals 1
postIds[ postIds.length - 1 ] // Last item
```

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( id = 0 ) {
    let user = apiMagic( id );
    return user;
}
getUser( 1 );
```

Objects

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

```
let post = {
  title: 'Hello World!',
  render: function() {
    console.log( this.title );
  }
}
```

```
post.title;
post.render();
```

Loops

Let us perform an action on a collection of items.

```
let postIds = [ 1, 7, 14, 34, 88, 117 ];


for ( let i = 0, max = postIds.length; i < max; i++ ) {
  console.log( 'Post #' + postIds[ i ] );
}
```

For Of Loop

Lets us loop through an array

```
var postIds = [ 1, 7, 14 ];


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

Mapping

Mapping is a functional alternative to looping

```
var postIds = [ 1, 7, 14 ];


postIds.map( id => {
   console.log( id );
});
```

Conditional Statements

Tests to determine what code to run

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

Include JS in HTML

```
<html>
  <head>
  </head>
  <body>
        <script>
            console.log( 'JS Running!' );
        </script>
  </body>
</html>
```

Include JS in HTML

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

JS 101
Pop Quiz!

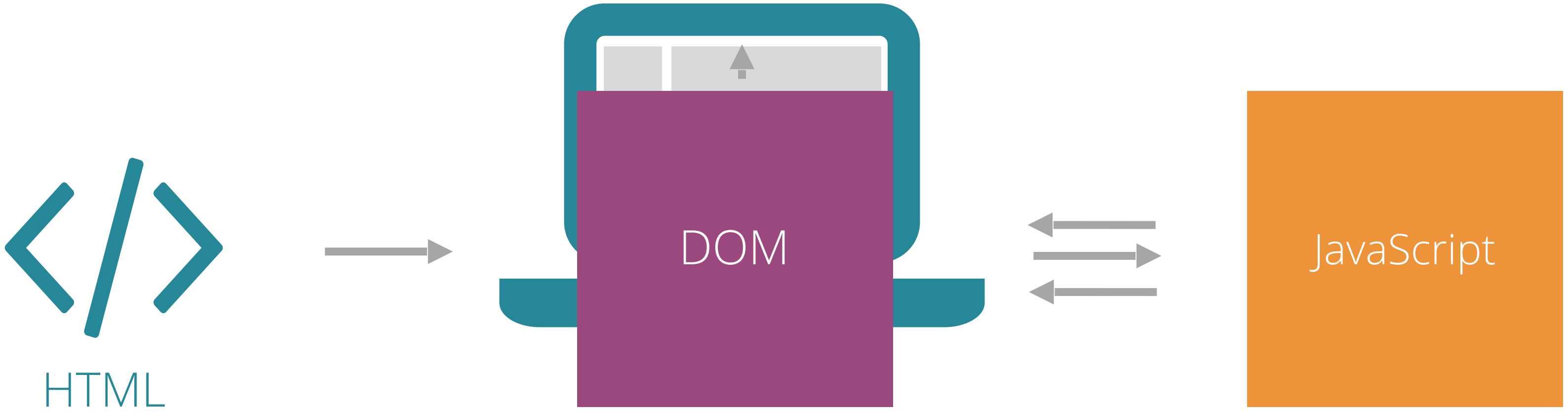
JS 101 Pop Quiz!!!

1. What is a variable?
2. What is an array?
3. What is a function?
4. What is an Object?
5. What is a Loop?
6. What is a Conditional Statement?
7. How do you include JavaScript in HTML?

The DOM & Nodes

The DOM is an API

For HTML (XML & SVG)


HTML
DOM
JavaScript

Build documents, **navigate** their structure, and **add, modify, or delete** elements and content.

Everything is a Node

In the DOM

Common Node Types

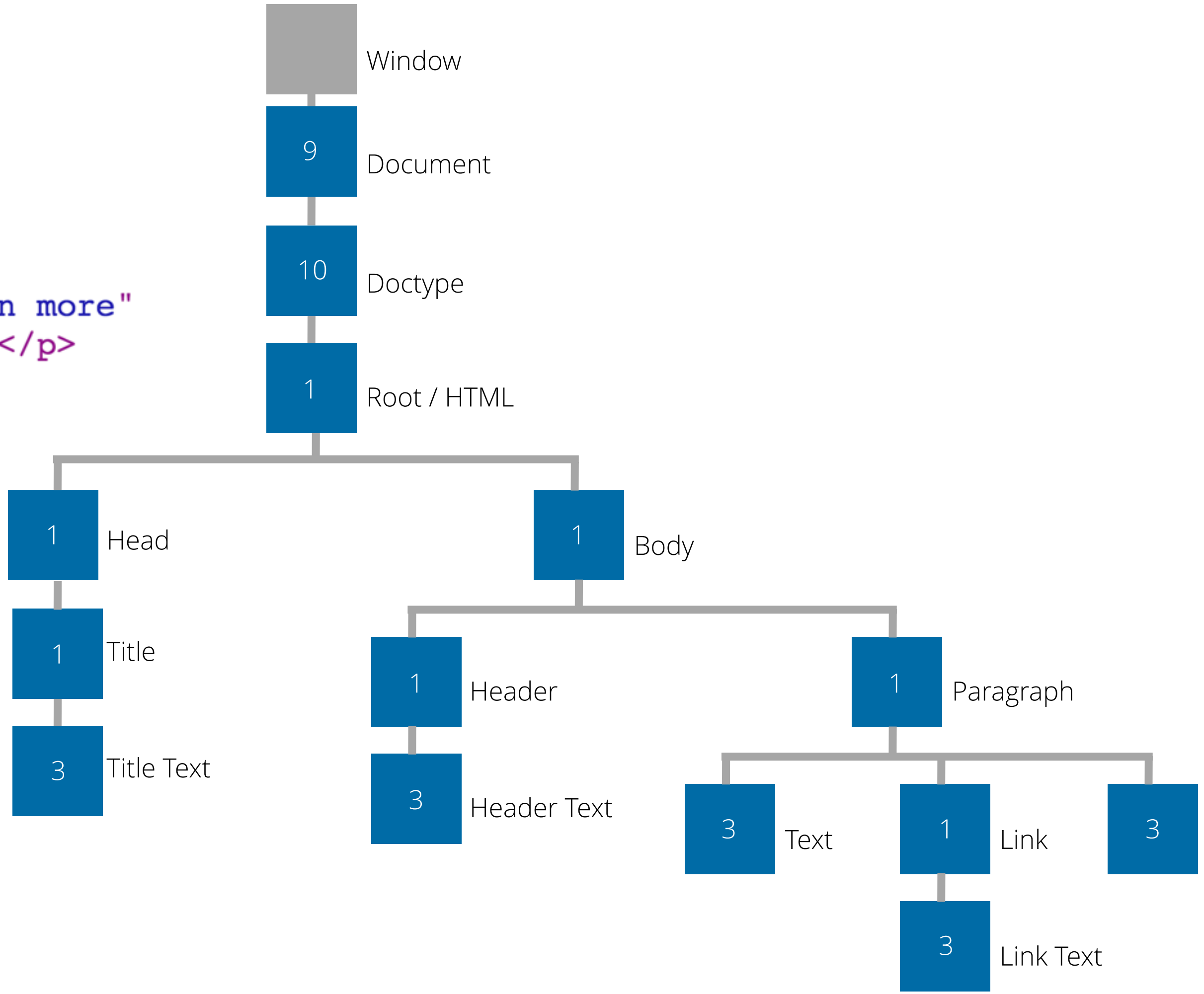
- Document [9]
- DocumentType [10]
- Element [1]
- Text [3]
- Comments [8]
- DocumentFragments [11]

Common Node Types

- Document [9]
- DocumentType [10]
- **Element [1]**
- **Text [3]**
- Comments [8]
- *DocumentFragments [11]*

```
<!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>
```

```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>
```



Getting DOM Nodes w JS

```
document.getElementById( 'main' )
document.getElementsByTagName( 'a' )
document.getElementsByClassName( 'post' )
document.querySelector( '.entry-title a' )
document.querySelectorAll( '.entry-title a' )
```

Getting Node Values

```
const el = document.getElementById( 'main' );
```

```
el.innerText          el.href
el.innerHTML          el.dataset.custom
el.id                 input.value
```

Setting Node Values

```
const el = document.getElementById( 'main' );

el.innerText = 'New';
el.innerHTML = '<p>New</p>';
input.value = 'zgordon';
```

PRACTICE 1.1

Getting & Setting Nodes w JS

Creating & Appending Nodes

Creating Nodes

```
document.createElement( 'p' )
document.createTextNode( 'Text' )
```

Appending Nodes

```
parent.appendChild( newEl );
parent.insertBefore( newEl, el );
```

Appending Nodes Example

```
let pEl = document.querySelector( 'p' ),
    aEl = document.createElement( 'a' ),
    aText = document.createTextNode( 'Link' );

aEl.appendChild( aText );
aEl.href = 'https://javascriptforwp.com';
pEl.appendChild( aEl );
```

“Hacker Style” Appending Nodes

```
let pEl = document.querySelector( 'p' ),
    markup = '';

markup += '<a href="https://site.com">' );
markup += 'Link Text';
markup += '</a>';

pEl.innerHTML = markup;
```

Practice 1.2 + 1.3

Creating & Appending Nodes

Events w JavaScript

Types of Events

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

https://javascriptforwp.com/intro-to-events/ ~40mins

Hooking into Events w JavaScript

1. **Inline** in HTML	<a onclick="alert('hi')">Alert</a>
2. **Global** 1 Off	a.onclick = sayHi
3. **Listeners** Best	a.addEventListener('click’, sayHi, false)

Event Listeners

```
const linkEl = document.querySelector( 'a' );

function displayLinkInfo( event ) {

  event.preventDefault();
  console.log( event );

}

linkEl.addEventListener( 'click', displayLinkInfo, false );
```

The Event Object

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

function sayHi( event ) {
  console.log( event );
  console.log( event.type );
  console.log( event.target );
}
```

Removing Event Listeners

```
el.addEventListener( 'click', sayHi, false );
el.removeEventListener( 'click', sayHi, false );
```

Practice 1.4

Event Listeners Practice

The WordPress REST API

```
/wp-json/wp/v2/posts
/wp-json/wp/v2/revisions
/wp-json/wp/v2/categories
/wp-json/wp/v2/tags
/wp-json/wp/v2/pages
/wp-json/wp/v2/comments
/wp-json/wp/v2/taxonomies
/wp-json/wp/v2/media
/wp-json/wp/v2/users
/wp-json/wp/v2/types
/wp-json/wp/v2/statuses
/wp-json/wp/v2/settings
```

```
/wp-json/custom/v1/something
/wp-json/another/v3/else
```

The WordPress REST API

To access all of the posts for your site:

https://yoursite.com/wp/v2/posts

To access a **post with an ID** of “1” for your site:

https://yoursite.com/wp/v2/posts/1

The WordPress REST API

To access all of the pages for your site:

https://yoursite.com/wp-json/wp/v2/pages

To access a **page with an ID** of “1” for your site:

https://yoursite.com/wp-json/wp/v2/pages/1

You can pass **arguments** to API URLs to **customize what you get** back.

The WordPress REST API

To access all of the pages for your site:

https://yoursite.com/wp/v2/pages

To access a **page with an ID** of “1” for your site:

https://yoursite.com/wp/v2/pages/1

The WordPress REST API

https://yoursite.com/wp/v2/posts

? per_page=10

&_embed

&search=hello%20world

&orderby=title

&order=asc

https://developer.wordpress.org/rest-api/reference/posts/#arguments

PRACTICE - WP REST API

1. Access API endpoints in the browser
2. Access API endpoints w arguments
3. Access custom API endpoints in the browser

The WordPress REST API

The WP REST API

Allows us to get, save, edit, and remove content from WordPress using JavaScript.

The WordPress REST API

```
/wp-json/wp/v2/posts          /wp-json/wp/v2/taxonomies
/wp-json/wp/v2/revisions      /wp-json/wp/v2/media
/wp-json/wp/v2/categories     /wp-json/wp/v2/users
/wp-json/wp/v2/tags           /wp-json/wp/v2/types
/wp-json/wp/v2/pages          /wp-json/wp/v2/statuses
/wp-json/wp/v2/comments       /wp-json/wp/v2/settings

/wp-json/custom/v1/something
/wp-json/another/v3/else
```

The WordPress REST API

To access all of the posts for your site:

https://yoursite.com/wp/v2/posts

To access a **post with an ID** of “1” for your site:

https://yoursite.com/wp/v2/posts/1

The WordPress REST API

To access all of the pages for your site:

https://yoursite.com/wp-json/wp/v2/pages

To access a **page with an ID** of “5” for your site:

https://yoursite.com/wp-json/wp/v2/pages/5

You can pass **arguments** to API URLs to **customize what you get** back.

The WordPress REST API

https://yoursite.com/wp/v2/posts

```
        ? per_page=10
        &_embed
        &search=hello%20world
        &orderby=title
        &order=asc
```

https://developer.wordpress.org/rest-api/reference/posts/#arguments

Practice 2.1

WP API in the Browser

WP API Calls with AJAX

ajax

Allows for us to make calls to the WP REST API and get back content in a format JavaScript can understand. (Can also save, edit and delete)

Ways to Make AJAX Calls

- XMLHttpRequests()
- fetch()
- Axios
- jQuery
- Backbone API Client

AJAX with Axios

```
axios({
    method: 'get',
    url: 'https://example.dev/wp-json/wp/v2/posts',
    params: {
        per_page: 3
    }
})
    .then( function( response ) {
        for( post of response.data ) {
            loadPost( post );
        }
    });
```

Practice 2.2 + 2.3

WP API with Axios Ajax

The WP REST API Inside of Themes

Enqueuing JavaScript

The process of loading JavaScript into a theme or the admin area via PHP functions.

wp_enqueue_script();

wp_localize_script();

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  'unique-handle-name',
  get_stylesheet_directory_uri() . '/path/to/file.js',
  [ 'dependency-handles' ],
  'version',
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘dependency-handles’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

```
wp_localize_script(
 ‘js-handle',
 ‘name_for_js’,
 [
   'site_url' => esc_url( home_url() ),
   'site_name'=> get_bloginfo( 'name' )
 ]
);
```

```
wp_localize_script(
 ‘js-handle',
 ‘name_for_js’,
 [
   'site_url' => esc_url( home_url() ),
   'site_name'=> get_bloginfo( 'name' )
 ]
);
```

```
wp_localize_script(
  ‘js-handle',
  ‘name_for_js’,
  [
    'site_url' => esc_url( home_url() ),
    'site_name'=> get_bloginfo( 'name' )
  ]
);
```

```
wp_localize_script(
  ‘js-handle',
  ‘name_for_js’,
  [
    'site_url' => esc_url( home_url() ),
    'site_name'=> get_bloginfo( 'name' )
  ]
);
```

```
wp_localize_script(
  ‘js-handle',
  ‘name_for_js’,
  [
    'site_url' => esc_url( home_url() ),
    'site_name'=> get_bloginfo( 'name' )
  ]
);  // Can use name_for_js.site_url in our JS
```

functions.php:

```
function my_enqueue_scripts() {
	wp_enqueue_script();
	wp_localize_script();
}
add_action( 'wp_enqueue_scripts', ‘my_enqueue_scripts' );
```

Practice 3.1

Enqueuing JavaScript in a Theme

jQuery + WordPress

WordPress comes with

Several **JS Libraries**

Available to Us

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘jquery’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

The **$** in jQuery

is **NOT available**

by **default**

```
jQuery(‘.this-works’).show();
```

```
(function( $ ) {

    $('.this-works').show();

})( jQuery );
```

Practice 3.2

Using jQuery for AJAX

The Backbone WP API Client

Backbone API Client

Allows us to easily use the WP REST API in a theme or plugin without writing and AJAX calls or worrying about Authentication.

No knowledge of Backbone necessary.

```
wp_enqueue_script(
  ‘unique-handle-name‘,
  get_stylesheet_directory_uri() . ‘/path/to/file.js’,
  [ ‘jquery’ ],
  ‘version’,
  loadInFooter // boolean true/false
);
```

The **$** in jQuery

is **NOT available**

by **default**

```
jQuery(‘.this-works’).show();
```

```
let posts = new wp.api.collections.Posts();
posts.fetch({
    data: {
        per_page: 3
    }
})
.done( () => {
   posts.each( post => {
        loadPost( post.attributes );
    });
```

```
let post = new wp.api.models.Post( { id: id } );
post.fetch()
    .done( () => {
        loadPost( post.attributes );
    });
```

Practice 3.3

Using the Backbone API Client

javascriptforwp.com/category/free-videos

Learn JavaScript Deeply with Zac Gordon

@zgordon