# A Comprehensive Guide to Intermediate JavaScript

Mastering Your "goding" Skills

## Behavior of Functions and Variables

- Variable Scope
- Variable Hoisting
- ES2015 Scope & Hoisting let & const
- Introduction to Callbacks
- Closures
- Value vs. Reference
- Value vs. Reference with ES2015

```
modifier_ob.
  mirror object to mirror
mirror_mod.mirror_object
 peration == "MIRROR_X":
irror_mod.use_x = True
"Irror_mod.use_y = False
lrror_mod.use_z = False
 _operation == "MIRROR_Y"
lrror_mod.use_x = False
 lrror_mod.use_y = True
 lrror_mod.use_z = False
 _operation == "MIRROR_Z"
  lrror_mod.use_x = False
  lrror_mod.use_y = False
  rror_mod.use_z = True
  melection at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.active
   "Selected" + str(modified
   rror ob.select = 0
  bpy.context.selected_obje
  lata.objects[one.name].sel
  int("please select exactle
  --- OPERATOR CLASSES ----
      mirror to the selected
    ject.mirror_mirror_x"
  ext.active_object is not
```

#### Essential Modern JavaScript - ES2015+ In Depth

- Template Literals
- Default Function Parameters
- Arrow Functions () => {}
- New Object Functionality
- Array & Object Destructuring



### Object Oriented Programming: The Core

Introduction to 'this', 'new' & OOP

The Complete Rules to `new`

Apply, Call, Bind, & arguments

The Utility of Rest & Spread



**Object Oriented Programming: `this`** and Inheritance

The Complete Rules to 'this'

Arrow Functions and 'this'

**Object Prototypes** 

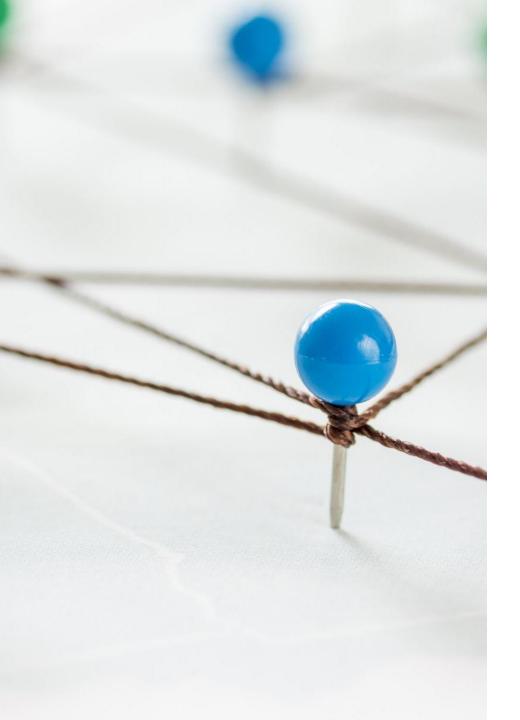
Implementing Inheritance



Introduction to Functional Programming

Immediately
Invoked Function
Expressions

The IIFE Interview Question



#### Functional Programming: Arrays and their Powerful Methods

- Map
- Filter
- Map & Filter Practice
- forEach
- Reduce
- forEach & Reduce Practice



#### Functional Programming: Objects and their Powerful Methods

Object	Object.assign
Object	Object.keys
Object	Object.values & Object.entries
Object	Object.defineProperty
Object	Object.preventExtensions, seal, and freeze

```
modifier_ob
  mirror object to mirror
mirror_mod.mirror_object
 peration == "MIRROR_X":
Lrror_mod.use_x = True
lrror_mod.use_y = False
 lrror_mod.use_z = False
 _operation == "MIRROR_Y"
 lrror_mod.use_x = False
 "Irror_mod.use_y = True"
 __mod.use_z = False
  _operation == "MIRROR_Z"
  lrror_mod.use_x = False
  _rror_mod.use_y = False
  rror_mod.use_z = True
  election at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
   "Selected" + str(modified
    rror ob.select = 0
  bpy.context.selected_obj
   lata.objects[one.name].sel
  int("please select exactle
  --- OPERATOR CLASSES ----
      mirror to the selected
     ect.mirror mirror x
  ext.active_object is not
```

### A Dive into Asynchronous JavaScript

- Introduction to Asynchronous Programming in JavaScript
- Callbacks, setTimeout, & the Event Loop
- Common Use Cases of Asynchronous JavaScript Programming
- Promises
- Error Handling with Promises
- Async/Await
- ES5, Promises and async/await Conclusion

### Advanced Modern JavaScript - Cutting Edge ES2015+

Section Overview

Classes

Sets

Maps

Symbols

Iterables & Iterators

Generators



Projects

