**JavaScript Learning Path**

**1. Basics**

* **Variables:** var, let, const
* **Data Types:** String, Number, Boolean, Object, Array, Null, Undefined, Symbol
* **Operators:**
  + Arithmetic: +, -, \*, /, %
  + Assignment: =, +=, -=
  + Comparison: ==, ===, !=, >, <
  + Logical: &&, ||, !
  + Ternary: condition ? expr1 : expr2

**2. Control Structures**

* **Conditionals:** if, else, else if, switch
* **Loops:** for, while, do...while, for...of, for...in

**3. Functions**

* **Declaration:** function name() { ... }
* **Expression:** const myFunc = function() { ... };
* **Arrow Functions:** const add = (a, b) => a + b;
* **IIFE:** (function() { ... })();
* **Parameters:** Default function greet(name = "User") { ... }, Rest (...args), Spread [...arr]

**4. Arrays**

* **Creation:** let arr = [1, 2, 3];
* **Methods:**
  + Modify: push(), pop(), shift(), unshift()
  + Transform: map(), filter(), reduce(), forEach()
  + Search: find(), findIndex()
  + Combine: slice(), splice(), concat()

**5. Objects**

* **Creation:** let obj = { key: "value" };
* **Methods:** Object.keys(), Object.values(), Object.assign()
* **Destructuring:** const { key } = obj;, [a, b] = [1, 2];

**6. ES6+ Features**

* **Template Literals:** `Hello, ${name}`
* **Shorthand:** const user = { name, age };
* **Destructuring:** { key } = obj;
* **Spread/Rest:** [...arr], (...args) => { ... }
* **Modules:** import/export

**7. Async JS**

* **Callbacks:** function(callback) { callback(data); }
* **Promises:** .then(), .catch()
* **Async/Await:** const data = await fetch(url);

**8. DOM Manipulation**

* **Select:** document.querySelector()
* **Modify:** element.textContent, element.style
* **Events:** element.addEventListener('click', () => { ... });

**9. Error Handling**

* **Try/Catch:** try { ... } catch (e) { ... }
* **Throw:** throw new Error("Error!");

**10. Advanced Topics**

* **Generators:** function\* gen() { yield 1; }
* **Sets/Maps:** let set = new Set();, let map = new Map();

**11. APIs**

* **Fetch:** fetch(url).then(res => res.json());
* **Axios:** axios.get(url).then(res => { ... });

**12. Classes**

* **Declaration:** class Person { constructor(name) { this.name = name; } }
* **Methods:** greet() { return this.name; }

**13. Closures**

* **Example:** function outer() { let x = 10; return function inner() { return x; }; }

**14. Higher-Order Functions**

* **Array Methods:** map(), filter(), reduce()

**15. Design Patterns**

* **Singleton:** const instance = { ... };
* **Module:** (function() { ... })();
* **Observer:** Subscribe/notify system.