## **JavaScript Syllabus**

|  | UNITS  Outlined below are the units for this 8-week course. Keep in mind that we do not expect you to come to the course knowing the different terms, languages, and technical words that are included in the units. |
| --- | --- |
| UNIT 1 | **Basics**     * Dynamic * Numbers   + Only one type - no integer vs float   + Number limits   + Number operators   + Precision   + NaN   + Infinity/-Infinity * Strings   + How they are written   + Breaking up code lines within stings   + Escaping   + Concatenation   + Template literals   + Unicode   + Numeric strings   + Adding numbers and strings * Boolean   + Comparison Operators   + Logical Operators * Arrays   + How written   + Zero-based   **Section 2: (4 Hours)** - *Data Types & Operators*   * Objects   + Name:value pairs   + How written * Typeof Operator * Empty values * Null and Undefined   + Difference between * Operators   + Binary   + Unary * Type conversion   + How expressions are evaluated (left to right) * Value Comparison   + == vs === * Practice Examples |
| UNIT 2 | **Programming Structure**   * Definition of expression * Bindings   + Var   + Const   + Let * Binding names * Functions   + Function definitions   + invoking/ calling * Methods   + This keyword   + Accessing methods   + invoking/calling/applying     - Event     - Called     - Self-invoked     - Purpose of ()   Accessing function without ()   * + Arguments/Parameters   + Return values   **Section 4: (4 Hours)** - *Programming Structure*   * Control Flow * Conditional Execution   + if/else   + For loops   + Types     - For     - for/in     - for/of     - while/do-while       * Differences     - break/continue   + Switch Statements     - Case     - Default     - Break     - Matching multiple cases     - Strict comparison   + Indentation and camelCase |
| UNIT 3 | **Object Oriented Programming**    Function Syntax   * Defining Functions   + Parameters     - VS arguments * Bindings and Scopes   + Global   + Local     - Function     - Block   + Nested scope * Functions as values * Function declaration   + Hoisting     - Var, let/const     - Initialization * Arrow Functions * Call stack * Optional arguments * Closure * Recursion * Controlling function size * Functions and side effects   Objects and Arrays   * Data sets/Arrays   + Creating     - Using [ ] for new array     - Multiple lines     - Keyword ‘new’     - Accessing elements of an array     - Changing array elements     - Accessing full array     - Looping through arrays * Properties   + name/value pairs * Objects   + Differences between arrays and objects     - Names indexes vs numbered indexes   + Object literals   + Accessing object members * Mutability   + Identity   + Array Methods     - Length     - pop/push     - Substring     - shift/unshift     - indexOf/lastIndexOf     - forEach     - toString     - Join     - delete     - concat     - Sort       * Compare function     - Reverse   + String methods     - Slice/substring/substr       * differences     - indexOf/search       * differences     - Trim     - split/join     - Length     - toUpperCase/toLowerCase     - Repeat     - Replace     - Concat     - padStart/padEnd     - charAt/charCodeAt   + Number methods     - toString     - toExponential     - toFixed     - toPrecision     - valueOf     - Number method     - parseInt     - parseFloat   + Rest parameters   + Math object     - Namespace - definition     - Math functions       * min/max       * Sqrt       * Random         + Not really random       * floor/ceil       * cos/sin       * round       * pow       * Abs     - Math constants       * PI |
| UNIT 4 | **Regular Expression/Asynchronous Programming**  Regular expressions   * Purpose * Creating * Testing for matches * Sets of characters   + Modifiers   + Metacharacters   + Quantifiers * Repeating parts of a pattern * Grouping subexpressions * Matches and groups * Date class * Word and string boundaries * Choice patterns * How matching works * Backtracking * Replace method * Greed * Search method * lastIndex property   Asynchronous programming   * Definition * Callbacks * Promises * Event loop   + setTimeout |
| UNIT 5 | **Client-Side Programming**  JSON   * Definition * Structure * serialize/deserialize * Methods   + Stringify   + Parse * Data Object   + Creating     - Different ways to create   + How dates are stored   + Date methods     - toString/toUTCString/toDateString/toISOString     - getFullYear/getMonth/getDate/getHours/getMinutes/getSeconds/getMilliseconds/getTime/getDay     - Date.now     - UTC Date methods     - Set Date methods       * setDate       * setFullYear       * setHours       * setMilliseconds       * setMinutes       * setMonth       * setSeconds       * setTime   + ISO dates   + Time Zones   + Long Dates   + Parsing Dates   DOM   * Trees * Traversal * Getting elements * Changing the document * Creating nodes * Attributes * Layout * Styling * querySelectors   Event handling   * Events and DOM nodes * Event objects * Propagation * Default actions * Types of events   + Key events   + Pointer events   + Scroll events   + Focus events   + Load event * Event handler attributes   + Onclick   + Onchange   + Onmouseover   + Onmouseout   + Onkeydown   + onload * Events and the event loop * Timers * Debouncing * Encapsulation * Prototypes * Classes   + constructor   + Using * Class notation * Overriding prototype properties * Maps * Polymorphism * Iterators (next) * Getters and setters * Static * Inheritance * instanceOf   Bugs & Eros   * Console.log * Dev tools * Strict Mode * Types * try/catch/finally/throw * Validation * Testing * Debugging * Error propagation * exceptions |
| UNIT 6 | **Server-Side Programming**  **Using MVC to Structure Our Application**  Node JS   * Introduction   + What is Node JS?   + Advantages of Node JS   + Traditional Web Server Model   + Node.js Process Model   Setup Dev Environment   * Install Node.js on Windows * Installing in mac os * Working in REPL * Node JS Console   Modules   * Functions * Buffer Module * Module Types * Core Modules * Local Modules * Module.Exports   NPM   * What is NPM * Installing Packages * Locally Adding dependency in package.json * Installing packages globally * Updating packages   Creating Web Server   * Creating web server * Handling http request * Sending request   Events   * EventEmitter class * Returning event emitter * Inhering events   Express JS   * Configuring routes * Working with express * Express application structures - routes, views, and static files   Serving Static Resources   * Serving static files * Working with middle ware |
| Unit 7 | **Debugging Node JS Application**   * Core Node JS debugger * Node Inspector * Debugging with Visual Studio   + Set breakpoints to pause execution   + Debug variables   + Debug functions   **Unit Testing with Mocha**  **Test Driven Development**   * Automate and Organize Tests * Write Expressive Tests * Learn TDD with Mocha |
| UNIT 8 | **Database /RESTful Services**  Database Connectivity   * installing MySQL Workbench * Create sample Database * Connection string * Configuring * Working with select command   RESTful Web Services  Postman   * REST API creation and consumption using JSON format. * Connection * Collections * Updating records * Deleting records |
| UNIT 9 | **Template Engines**     * Why template engine * What is jade   + Simple tags   + Adding attributes to tags   + Blocks of text   + Powerful Features     - Loops     - Javascript     - Interpolation     - Mixins   + Putting it all together * What is vash   + Configuration   + Using vash with express.js   + Template options   + Helpers   + API * Example |
| Unit 10 | **Final Project**  **Students will get more than a week to complete their final project but that last week will be used primarily for this. It will also be a time to ask questions pertaining to final project code or any errors etc.** |

|  | Resources |
| --- | --- |
|  | Students are not required to purchase a textbook for this course because there are many online resources available either publicly on the web or as e-books. |
|  | NEXT STEPS  Are you ready to change your career path and learn an incredibly valued skill set? [Apply](https://interfaceschool.com/course/foundations/) today. There’s no risk in applying! After your application has been received, you’ll be sent an online, non-technical assessment to gauge your problem-solving skills. |