**Language Map for JavaScript**

|  |  |
| --- | --- |
| **Variable Declaration**  *Is this language strongly typed or dynamically typed? Provide at least three examples (with different data types or keywords) of how variables are declared in this language.* | JavaScript is a dynamically typed language.  Ex: let name = “Chris”;  let age = 23;  let fruit = [“apple”, “orange”, “banana”] |
| **Data Types**  *List all of the data types (and ranges) supported by this language.* | JavaScript supports 8 different data types, including:  1. Strings – limited by memory 2. Number- floating point #’s between -2-1074 and 21024 but safely between -253 – 1 to 253 − 1 3. Bigint – limited by memory 4. Boolean – true or false 5. Undefined – just undefined 6. Null – just null 7. Symbol – no numerical range 8. Object(arrays, objects, dates) – available memory |
| **Selection Structures**  *Provide examples of all selection structures supported by this language (if, if else, etc.)* ***Don’t just list them, show code samples of how each would look in a real program.*** | **If statement:**  let x = 10;  if (x > 0) {  console.log("x is a positive number.");  }  **If-else statement:**  let temperature = 75;  if (temperature > 80) {  console.log("It's a hot day");  } else {  console.log("It's not too hot today");  }  **Else-if statement:**  let time = 18;  if (time < 12) {  console.log("Good morning!");  } else if (time < 18) {  console.log("Good afternoon!");  } else {  console.log("Good evening!");  }  **Switch statement:**  let id = 3;  let idNum;  switch (id) {  case 1:  idNum = "manager";  break;  case 2:  idNum = "supervisor";  break;  case 3:  idNum = "floor";  break;  default:  idNum = "invalid";  }  **Ternary operator:**  let age = 21;  let eligibility = (age >= 18) ? "Eligible to vote" : "Not eligible to vote";  console.log(eligibility); |
| **Repetition Structures**  *Provide examples of all repetition structures supported by this language (loops, etc.)* ***Don’t just list them, show code samples of how each would look in a real program.*** | **For loop:**  for (let i = 0; i < 5; i++) {  console.log(`Iteration ${i + 1}`);  }  **While loop:**  let count = 0;  while (count < 3) {  console.log(`Count: ${count}`);  count++;  }  **Do-while loop:**  let num = 1;  do {  console.log(`Number: ${num}`);  num++;  } while (num <= 5);  **For-in loop:**  const person = {  name: "John",  age: 30,  job: "Developer"  };  for (let key in person) {  console.log(`${key}: ${person[key]}`);  } |
| **Arrays**  *If this language supports arrays, provide at least two examples of creating an array with a primitive or String data types (e.g. float, int, String, etc.)* | Array of integers  let numbers = [1, 2, 3, 4, 5];  Array of floating-point numbers  let decimalNumbers = [1.8, 2.4, 3.2, 4.9, 5.5];  Array of strings  let sports = ["baseball", "football", "basketball", "hockey", "soccer"]; |
| **Data Structures**  *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.* | **Arrays:**  Access: O(1)  Search: O(n)  Insertion (at the end): O(1)  Deletion (at the end): O(1)  Insertion (at the beginning or in the middle): O(n)  Deletion (in the middle): O(n)  **Linked Lists:**  Access: O(n)  Search: O(n)  Insertion : O(1)  Deletion: O(1)  **Sets:**  Search: O(1) on average  Insertion: O(1) on average  Deletion: O(1) on average  **Maps (Objects in JavaScript):**  Search: O(1) on average  Insertion: O(1) on average  Deletion: O(1) on average  **Queues:**  Enqueue: O(1)  Dequeue: O(1)  **Stacks:**  Push: O(1)  Pop: O(1)  **Hash Tables:**  Search: O(1)  Insertion: O(1)  Deletion: O(1)  **Binary Trees:**  Search: O(log n)  Insertion: O(log n)  Deletion: O(log n)  **Heaps:**  Insertion: O(log n)  Extract Min/Max: O(log n) |
| **Objects**  *If this language support object-orientation, provide an example of how you would write a simple object with a default constructor and then how you would instantiate it.* | class Person {  constructor(name, age) {  this.name = name;  this.age = age;  }//end constructor  displayInfo() {  console.log(`Name: ${this.name}, Age: ${this.age}`);  }//end method  }//end class  // Instantiating  let person1 = new Person("John Doe", 25);  console.log(person1.name); // Output: John Doe  console.log(person1.age); // Output: 25 |
| **Runtime Environment**  *What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine.*  *Do other languages also compile to this runtime?* | JavaScript is not a compiled language, so it does not compile to a specific machine code. Instead, an interpreter in the browser reads over the JavaScript code, interprets each line, and runs it |
| **Libraries/Frameworks**  *What are the popular libraries or frameworks used by programmers for this language? List at least three (3) and describe what they are used for..* | 1. **React**: used for building dynamic web applications, reusable UI components, and User Interfaces 2. **Angular**: Developing scalable single-page applications, building web applications with a modular structure. 3. **Express.js**: building RESTful APIs, building server-side apps using Node.js |
| **Domains**  *What industries or domains use this programming language? Provide specific examples of companies that use this language and what they use it for****. E.g. Company X uses C# for its line of business applications.*** | 1. **Amazon:** Uses JavaScript for its website, incorporating features like product recommendations and real-time updates. 2. **Microsoft:** uses JavaScript to build Microsoft Edge browsers 3. **Google:** Uses JavaScript for building the user interfaces of its various web applications, including Gmail and Google Maps. |