Day 1 Assignments

Question 1: Explore and explain various methods in console functions. Explain them.

Answer:

1. **console.log():** It is a method used to write a message to the console.
   Example: console.log("Hello!!");
2. **console.error():** It is a method used to write an error to the console.
   Example: console.error("This is for testing");
3. **console.clear():** It is a method used to clear the console.
   Example: console.clear();
4. **console.assert():** It is a method used to write a message to the console only if the first argument is false.
   Example:
   ```
   //There is no ID name called Demo
   console.assert(document.getElementById("Demo"),"There is no ID named 'Demo'");
   ```
5. **console.warn():** It is a method that writes a warning to the console.
   Example: console.warn("WARNING!!!!");
6. **console.trace():** It is a method displays a trace that show how the code ended up at a certain point.
   Example: console.trace();
7. **console.info():** It is a method that writes a message to console.
   Example: console.info("Hello!!!");
8. **console.table():** It a method that writes a table to a console screen. To use this method there are few parameters required and it should be an object or an array that contains the data.
   Example: console.table(["Fruits","Vegitables"]);
9. **console.time():** It is a method that starts timer in console view and it allows you to time certain operations in your code for testing purposes
   Example: console.time():
10. **console.timeEnd():** It is a method that stops a timer in console window.
    Example: console.timeEnd();
11. **console.count():** It I a method that writes the number of times console.count() is been called.
    Example: console.count();

Question 2: Write the difference between var, let and const with code examples.

Answer:

a. **var:** The scope of a variable defined with the keyword "var" is limited to the "function" within which it is defined. If it is defined outside any function, the scope of the variable is global.
var is "function scoped".
Example:

```
<html>
<head></head>
<body>
<script>
{
  var a=10;
  console.log(a);
} //block 1{
  a++;
```

```
  console.log(a);
} //block 2
/* Since we are using "var a=10", scope of "a" is limited to the function within which it is defined. In this case it is within the global function scope */
</script>
</body>
</html>
```

b. **let:** The scope of a variable defined with the keyword “let” or “const” is limited to the “block” defined by curly braces i.e. {} .

“let” and “const” are“block scoped”.

Example:

```
<html>
<head></head>
<body>
<script>
{
 let a=10;
 console.log(a);
} //block 1{
  a++;
  console.log(a);
} //block 2/* Since we are using "let a=10", scope of "a" is limited to block 1 and "a" is not recognized in block 2 */
</script>
</body>
</html>
```

c. **const:** The scope of a variable defined with the keyword “const” is limited to the block defined by curly braces. However if a variable is defined with keyword const, it cannot be reassigned.

“const” cannot be re-assigned to a new value. However it CAN be mutated.

Example:

```
<html>
<head></head>
<body>
<script>
{
 const PI=3.14;
 console.log(PI);
} //block 1
{
  console.log(PI);
} //block 2
/* Since we are using "const PI=3.14", scope of "PI" is limited to block 1 and "PI" is not recognized in block 2 */
</script>
</body>
</html>
```

Question 3: Write a brief intro on available data types in Javascript.

Answer:

1. **Number:** The number type represents both integer and floating-point numbers. There are many operations for numbers, e.g. multiplication *, division /, addition +, subtraction -, and so on. It is limited by $\pm 2^{53}$.
   Example:
   let n=298;
   n=2.98;
2. **Bigint:** In JavaScript, the "number" type cannot represent integer values larger than (2^{53}-1) (that's 9007199254740991), or less than -(-2^{53}-1) for negatives. It's a technical limitation caused by their internal representation. For most purposes that's quite enough, but sometimes we need really big numbers, e.g. for cryptography or microsecond-precision timestamps.
   Example:
   const Bigint=87452704850857309857n
   // The n in the end means it's Bigint
3. **String: It** is a type of datatype that stores a single letter or word or even a sentence. To store string data, we use single quote or a double quote. In java script there is no difference in single and double quote.
   Example:
   let str='a';
   let str="Hello!!!";
   let str="Hello World!!!";
4. **Boolean:** It is a logical type data type. It has only two values i.e., TRUE and FALSE.
   Example:
   let a=10>5;
   Alert(a);
5. **Null**: The special null value does not belong to any of the types described above. It forms a separate type of its own which contains only the null value.
   Example:
   let aa=null;
   // it stores null value to aa
6. **Undefined:** The special value undefined also stands apart. It makes a type of its own, just like null. The meaning of undefined is "value is not assigned".
   Example:
   let aa=undefined;
   alert(aa); //Shows undefined