1. Javascript is an interpreted programming language that operates in the realm of high-level languages. It allows a user to have a more complex and deeper interaction with web pages, which can also create dynamic interactivity as well. An interpreted language is executed line by line at runtime. There is no need for a compiler, as an interpreter “interprets” the program at runtime. A compiler is needed to translate the higher-level code into machine-language instructions for a Compiled language. Interpreted coding languages make development quicker but may result in slower execution compared to compiled languages.

2. When declaring variables, ‘var’ Declares a variable that can be redeclared and reassigned. ‘let’ declares a variable that is limited in scope depending on where it is used. ‘let’ variables can be reassigned but not redeclared. Finally, const declares a constant that cannot be reassigned a different value post initialization.

3. Higher order functions in javascript are functions that take other functions as an argument, or, they can return a function as well. This allows the usage of paradigms of programming like reduce, map, and filter. Anonymous functions are functions without names. Usually used as arguments to other functions.

4. Objects in javascript are key value pairs. They can contain any datatypes, including functions and other objects. They can be mutated. However, Unlike traditional OOP programming like such found in c++, the classes in javascript are not like the traditional classes we know are so common in c++. Objects in javascript are dynamic, in the sense that one can add and remove properties at runtime. Classes also differ inbetween OOP and javascript due to dynamic vs static typing, as the variables can hold values of any type, and can change during runtime. This is not the case for an OOP language such as C++. This definitely is a big difference especially when using classes and objects.