1. **Truthy and Falsy Value:** Truthy and falsy are Boolean contexts. Truthy value is a type of value that is by default true in nature. And falsy value is the type of value that is by default false in nature. A value is truthy unless it’s defined as falsy.
2. **Undefined and Null:** Undefined means in JavaScript that a variable is being declared with no value. In other words, without value, declaring a variable is so-called undefined.

Null means zero. That is the absence of value and a variable. Neither variable nor value declared the JavaScript returns null.

1. **“==” and “===”:** The “==” and “===” operator is the comparison operator. The main difference between “==” and “===” are “==” compares only two variables without **checking the datatype** of the variable. Whereas the “===” operator compares two variables in **strict modes**, such as comparing the variable with **checking the datatypes**.
2. **Closure:** A closure is a combination of functions. In JavaScript, an inner function always takes some reference of the outer function. In general, a closure makes access to an outer function from an inner function. When the function is created a closure is also being created in JavaScript. A parent function always sends the value or variable to its child function. But The parent can’t access the child’s element.

A Closure is a function having access to the parent scope, even after the parent function is closed.

1. **Encapsulation:** The wrapping up of data and function into a single unit (called class) is known as encapsulation. The data is not accessible to the outside and those functions which are wrapped in the class can access it.
2. **Event loop Stack:** The Event loop has only one task which is it just monitors the callback queue and the callback stack. Generally, JavaScript programs are executed by the Google browser v8 engine or in node js. The JavaScript engine moves the code in a callback stack. Naturally, the stack is a LIFO system. All the functions and code go through to the stack. And then serially the code is being executed. But when an asynchronous function occurs like a set timeout. The JavaScript engine pops out the function and sends it to the web API for the browser or C++ API for node-js. After the pop-up, the stack works normally as before. Besides this, the web API also executed the function given from the callback stack. And after execution, the response is transferred to the callback queue. And all the responses get and executed by the web API from the callback stack are transferred to the callback queue. The callback queue stores the execution code and when the call stack is free it sends the code to the callback stack again. The monitoring between the callback queue and callback stack is done by the event loop.
3. **This:** This keyword refers to the object that it belongs to. In a method this refers to the owner object, in a function this refers to the global object.
4. **Event Bubble and Event Delegate:** Event bubbling is a method in which an event is propagated to one element inside another element and both elements is being registered and handled. The event is first captured to the inside element and then propagated to the outside element. We can terminate the event bubble by using the stopPropagation method.
5. **DOM:** DOM stands for Document Object Model. DOM represents the content or document visually to the web browser. It’s a programming interface for web documents. The Dom is not a part of the JavaScript language but is instead a Web API used to build the websites.
6. **Window and Document:** A window is a global object and contains all the variables, functions, classes and history, and location. The document is also a part of the window and as considered the part of the Window

1. **New :** The new operator lets developers create an instance of a user-defined object type or of one of the built-in object types that have a constructor function.