**Objects**

* String is used to access the different elements of objects.
* String is called property or key and the element which property points to is called values.
* For creating general objects (i.e., when classes are not involved), use literals and not constructors. For example, do this: const object = {};
* var object = {

name: “Ayushi”,

age: 22

};

* JavaScript knows that the keys will always be strings, which is why you can leave out the quotes.
* When a variable is declare with a new keyword, the variable is created as an object.
* Assigning object to any variable does not copy that object but becomes that object. If changes made to variable then it will reflect in object.

**Accessing values in an object**

* Object[“name”];
* Other way to access object is objectname.property
* To display all keys of an object use **Object.keys(objectname)**
* Double quotes is required around key to access values.
* For in loops through the properties of an object.

**Adding values to object**

* Empty array and empty objects is same.
* Adding items to an object is same as adding values to an array.
* For example: var object1 = {};

object1[“Key”] = value;

* Other way is using dot notation.
* For example: var object = {};

object.name = “Ayushi”;

**Combining array and objects**

* Creating array of an objects: var arrayofobjects = [

{object1name: “name”},

{object2name: “name”}]

* To access an object use the same way as use to access an array. **arrayofobjects[0]**
* Using dot notation, arrayname[index].key

**Methods**

* Object also have methods.
* Methods are actions that can be performed on objects.
* Method is a function stored as properties.

**this keyword in javascript**

* This refers to the object.
* In a function, this refers to the global object.
* This keyword refers to the different objects depending on how it is used.

**Delete properties**

* Use delete keyword to delete the properties.
* It deletes both property as well as value.
* For example: delete object.property
* Also delete[“property name”]

**Nested Objects**

* Object inside an object.
* To access property of inner object use **outerobjectname.innerobjectname.property**  or outerobjectname[“innerobjectname”][“property”].

**Display only values**

* Object.values(objectname) displays values of an object in an array.
* JSON.stringify(objectname) converts all values to the string.

**Iterables**

* For – of loop is used to iterate over the elements.
* When it is used on string, it will return each character of string.
* In array, returns all elements of an array.