Builtin Methods (Numbers)

```
    let a = 10.7
    alert(Math.floor(a)) //rounds down
    alert(Math.ceil(a)) //rounds up
    alert(Math.round(a)) //rounds to nearest integer
    let b = 12.345
    alert(b.toFixed(1)) // returns string, try toFixed(5)
    let str = 10
    alert(isNaN(str))
    let v1 = "10“
    alert(parseInt(v1) + 2)
console.log(Number(v1) + 2)
    alert(Math.random())  // random between 0 and 1
    alert(Math.max(2,4,7,9,1))
    alert(Math.min(2,4,7,9,1))
    alert(Math.pow(2,3)) // 2 to the power 3
```

Builtin Methods (String)

```
let str = "Hello";
 alert(str[0])
 alert(str.charAt(0))
alert(str.length)
 for (let c of "Welcome") {
   alert(c);
 }
alert(str.toUpperCase())
alert(str.toLowerCase())
alert(str.indexOf('l’))
alert(str.lastIndexOf('l’))
alert(str.includes('l’))
alert(str.startsWith('H’))
alert(str.endsWith('o’))
alert(str.slice(1,4))  /try (-4,-1)
alert(str.substring(1,4)) // try substr - start, length
alert('a' > 'Z')  // comparing string
```

slice

Date Methods (get)

```
let d = new Date();
document.write(d.getDate()); // 1 to 31
document.write(d.getFullYear());
document.write(d.getMonth());
document.write(d.getDay()); //weekday
document.write(d.getHours());
document.write(d.getMinutes());
document.write(d.getSeconds());
document.write(d.getMilliseconds());
document.write(d.getTime()); // milliseconds since 1/1/1970
document.write(Date.now()); // milliseconds since 1/1/1970
```

Creating Dates in JavaScript

```
let d = new Date()
let d = new Date(2021,10,27,20,12,10) //try yyyy-mm-dd, mm/dd/yyyy
let d = new Date("November 10, 2015 11:13:00");
let d = new Date(1000) // milliseconds starts January 01, 1970
alert(d);
```

Date Methods (set)

```
let d = new Date()
d.setFullYear(2000)
d.setMonth(0)
d.setDate(1)
d.setHours(1)
d.setMinutes(1)
d.setMinutes(1)
d.setMilliseconds(1)  //0 – 999
d.setTime(1) // starting 1/1/1970
document.write(d)
```

Events and EventListener

```
 <input type="button" id="btn" value="Click" />
    <!-- <input type="button" id="btn" value="Click" onclick="alert('Hello World')" > -->
    <script>
      btn.addEventListener("click", function () {
        alert("Hello World - EventListener");
      });
    </script>
```

Javascript Hoisting

Hoisting is moving declarations to the top.

It is a JavaScript

……………………………………………………………..

```
x = 15;
elem = document.getElementById("mydiv");
elem.innerHTML = x;
```

var x; 's default behavior.

Accessing element’s value/innerHTML

```
<p id="p2">This is a paragraph</p>
  <input type="text" id="t1" placeholder="Enter a value">    <br>
  <input type="button" onclick="myfunc()" value="Submit">
  <script>
   function myfunc(){
    alert(document.getElementById('t1').value)
    alert(t1.value)
    alert(document.querySelector('#t1').value)
    alert(p2.innerHTML)
   }
  </script>
```

getElements (HTML Collection)

```
<body>
   <div class="c" name="d">Hello</div>
   <div class="c" name="d">Good Morning</div>
   <div class="c" name="d">Good Evening</div>
   <script>
      // let tags = document.getElementsByTagName('div’)
      let tags = document.getElementsByName('d’)
      // let tags = document.getElementsByClassName('c’)
      for (let i=0;i<tags.length;i++){
         tags[i].style.color='green’
      }
   </script>
</body>
```

Queryselector / QueryselectorAll

```
<body>
  <div class="c">Hello</div>
  <div class="c">Good Morning</div>
  <div class="c">Good Evening</div>
  <script>
   // let tags = document.querySelector('div’) //try #d for id
   // tags.style.color="red“
  //   let tags = document.querySelectorAll("div");
   let tags = document.querySelectorAll(".c");
   for (let i = 0; i < tags.length; i++) {
     tags[i].style.color = "red";
   }
  </script>
 </body>
```