Javascript

1.data types in javascript.
String
object
array
boolean
number
null
undefined
2. ,

3. HELLO,WORLD

4. var name = {color:”red”}

5. 9 times it will executed.

6.inner.HTML()
  console.log()
  window.alert()
  document.Write()

7. the length of that array is 51

8.

```
<html>
  <head></head>
<body>
<script>
function avgsum() {
var sum=0;

for (var i=1;i<6; i++){
sum +=i;
}

var avg=sum/5;
return "average is" +avg+"." "Sumis "+sum+".";
}
document.write(avgsum());

</script>
</body>
</html>
```

9.

```
<html>
<head></head>
<body>
<script>
function rev(words){
return word.split("").reverse(),
join("");
}
</script>
</body>>
</html>
```

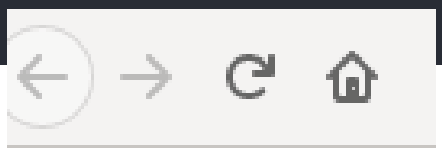
10.

```
<html>
<head></head>
<body>
  <button onclick="myFunction()">Try it</button>

  <p id="demo"></p>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = Math.max(5, 10);
}
</script>

</script>
</body>>
</html>
```



Try it

10

11.== : it is used to check the value equal to
=== : it is used to check the value equal and same data type

12.

age.html

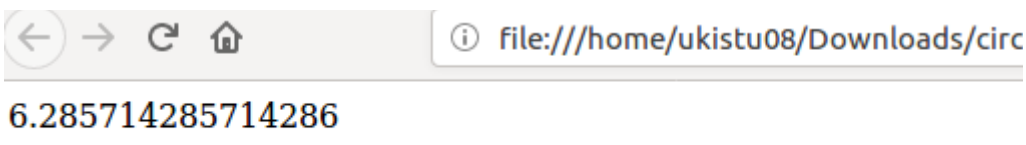
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <script>
    function dogAge(age) {
      var x;
      return x = " my dogAge is " + age * 7 + " years old in dog years ";

    }
    document.write(dogAge(3))
</script>
  </body>
</html>
```

.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <script>
    function calcCircumference( radius) {
      var circumference;
      return circumference = radius * (22/7)

    }
    document.write(calcCircumference(2))
    </script>
  </body>
</html>
```



```
<div>

<script>
function calcArea(r){
  var cir = Math.PI*r*r ;
}
console.log("The Area is "+ cir)

</script>
</div>
```

14. **`*const*`** is a signal that **the identifier won't be reassigned.**

`*let*`, is a signal that **the variable may be reassigned**, such as a counter in a loop, or a value swap in an algorithm. It also signals that the variable will be used **only in the block it's defined in**, which is not always the entire containing function.

```
<div>

<button onclick="findanimal()">Try it</button>

<p id="demo"></p>

<script>
function findanimal(animal) {

    if (animal == "car") {
        greeting = "meow";
    } else if (animal == "dog") {
        greeting = "bowbow";
    } else if (animal == "goat"){
        greeting = "baaa";
    } else if (animal == "duck") {
      greeting = "quck";

    }else {
        greeting = "Unknown Animal ";
    }
document.getElementById("demo").innerHTML = greeting;

</script>
</div>
```