

OBSERVATION REPORT

JavaScript Hands On 1

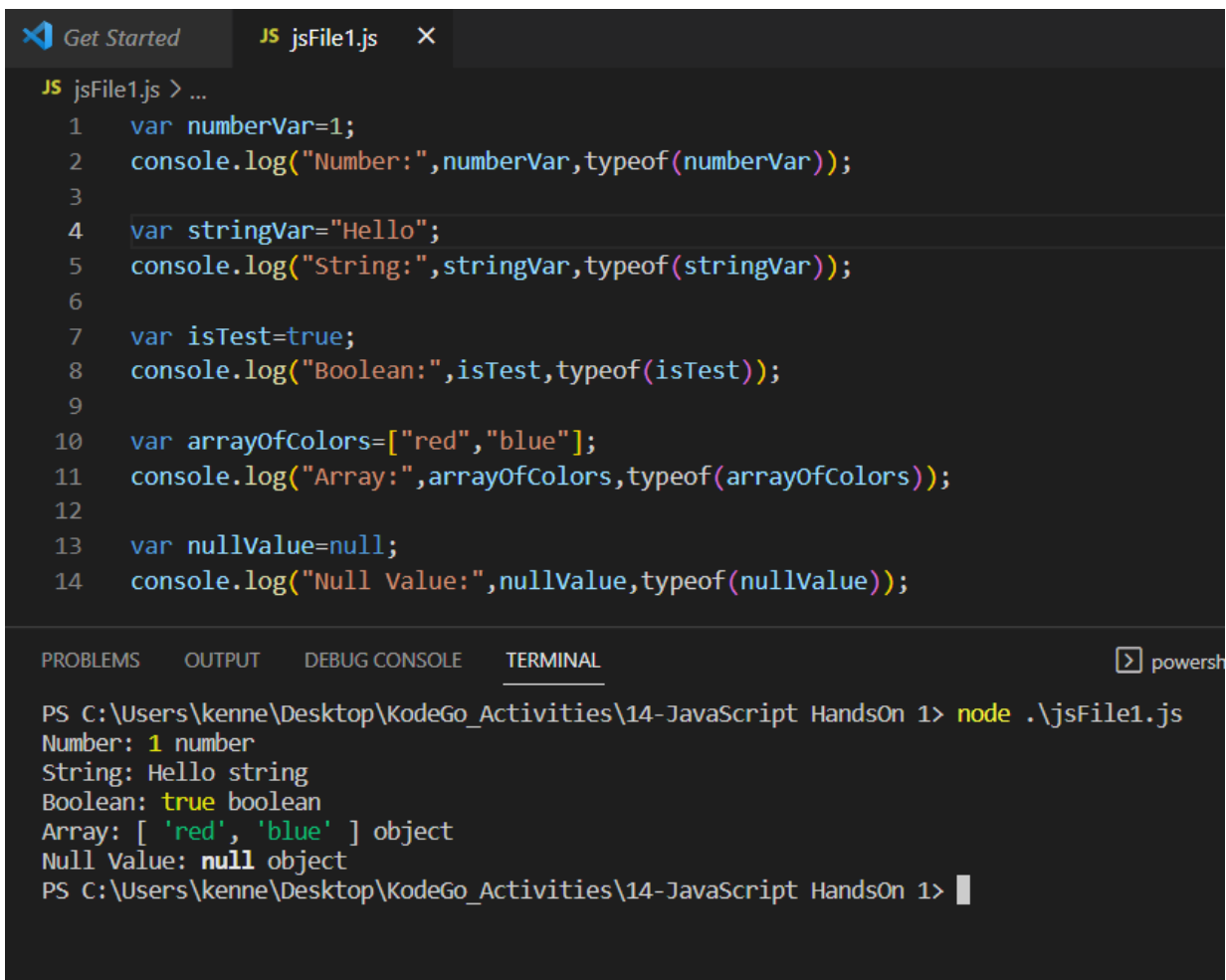
Kenneth Tan
Batch WD37

December, 06, 2022
Instructor: Sir Alfren

1. Create a Js file that will show different Data Types(Number, String, Object). Used typeof keyword to show the data types.

example:

```
console.log("Number: ", numberVar, typeof numberVar);  
console.log("String: ", stringVar, typeof stringVar);  
console.log("Boolean: ", isTest, typeof isTest);  
console.log("Array: ", arrayOfColors, typeof arrayOfColors);  
console.log("Null Value: ", nullValue, typeof nullValue);
```



The screenshot shows the Visual Studio Code editor with a file named `jsFile1.js` open. The code in the editor defines five variables and uses `console.log` to display their values and types. The terminal at the bottom shows the output of running `node .\jsFile1.js`.

```
JS jsFile1.js > ...  
1  var numberVar=1;  
2  console.log("Number:",numberVar,typeof(numberVar));  
3  
4  var stringVar="Hello";  
5  console.log("String:",stringVar,typeof(stringVar));  
6  
7  var isTest=true;  
8  console.log("Boolean:",isTest,typeof(isTest));  
9  
10 var arrayOfColors=["red","blue"];  
11 console.log("Array:",arrayOfColors,typeof(arrayOfColors));  
12  
13 var nullValue=null;  
14 console.log("Null Value:",nullValue,typeof(nullValue));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

```
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile1.js  
Number: 1 number  
String: Hello string  
Boolean: true boolean  
Array: [ 'red', 'blue' ] object  
Null Value: null object  
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> |
```

2. Create Js file to show different Mathematical Operations(+, -, *, /, **, %, ++, --).

```
JS jsFile2.js X
JS jsFile2.js > ...
1  var num1=4, num2=2;
2  console.log("Addition:",num1+num2);
3  console.log("Subtraction:",num1-num2);
4  console.log("Multiplication:",num1*num2);
5  console.log("Division:",num1/num2);
6  console.log("Exponent:",num1**num2);
7  console.log("Mod:",num1%num2);
8  console.log("Increment:",++num1);
9  console.log("Decrement:",--num1);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile2.js
Addition: 6
Subtraction: 2
Multiplication: 8
Division: 2
Exponent: 16
Mod: 0
Increment: 5
Decrement: 4
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> 
```

3. Create Js File to show different Assignment Operators (=, +=, -=, *=, /=, %=, **=).

```
JS jsFile3.js X
JS jsFile3.js
1  num1=5;
2  console.log("Equal:",num1);
3  console.log("Plus Equal:",num1+=1);
4  console.log("Minus Equal:",num1-=1);
5  console.log("Multiply Equal:",num1*=2);
6  console.log("Divide Equal:",num1/=2);
7  console.log("Mod Equal:",num1%=3);
8  console.log("Exponent Equal:",num1**=2);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile3.js
Equal: 5
Plus Equal: 6
Minus Equal: 5
Multiply Equal: 10
Divide Equal: 5
Mod Equal: 2
Exponent Equal: 4
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> 
```

4&5. Create Js file to show different Logical Operators (&& , || , !) using if-else statement and convert this to ternary operator.

```
JS jsFile4&5.js X
JS jsFile4&5.js
1 //jsFile4 and jsFile5
2 //Logical Operator, If Else Statement and Ternary Operator
3 num1=4;
4 if(num1>1 && num1<5){
5 |   console.log("&& result: true")
6 | }
7 if(num1>1 || num1<5){
8 |   console.log("|| result: true")
9 | }
10 if(num1!=0){
11 |   console.log("! result: true")
12 | }
13 console.log("Ternary result:",num1==4?true:false);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

```
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node '.\jsFile4&5.js'
&& result: true
|| result: true
! result: true
Ternary result: true
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1>
```

6. Create Js file that has a blank array, then push or assign 5 elements inside the array, after you add items in the array loop and show all the elements of the array.

```
JS jsFile6.js X
JS jsFile6.js > ...
1 arr=[];
2 arr.push(1,2,3,4,5);
3 for(var i=0;i<arr.length;i++){
4 |   console.log(arr[i]);
5 | }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

```
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile6.js
1
2
3
4
5
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1>
```

7. Create Js file to program if, else if , else statement.

```
JS jsFile7.js X
JS jsFile7.js
1  num1=4;
2  if(num1<5){
3      console.log("result: if satisfied");
4  }
5
6  if(num1>5){
7      console.log("result: if satisfied");
8  } else if(num1<5){
9      console.log("result: else if satisfied");
10 }
11
12 if(num1>5){
13     console.log("result: if satisfied");
14 } else {
15     console.log("result: else");
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

```
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile7.js
result: if satisfied
result: else if satisfied
result: else
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> 
```

8. Create Js file to program switch case statement.

```
JS jsFile8.js X
JS jsFile8.js > ...
1  var level=2;
2  switch(level){
3      case 1: console.log("You're level 1");
4      case 2: console.log("You're level 2");
5  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell

```
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> node .\jsFile8.js
You're level 2
PS C:\Users\kenne\Desktop\KodeGo_Activities\14-JavaScript HandsOn 1> 
```