

Observation Report

JavaScript Hands On 1

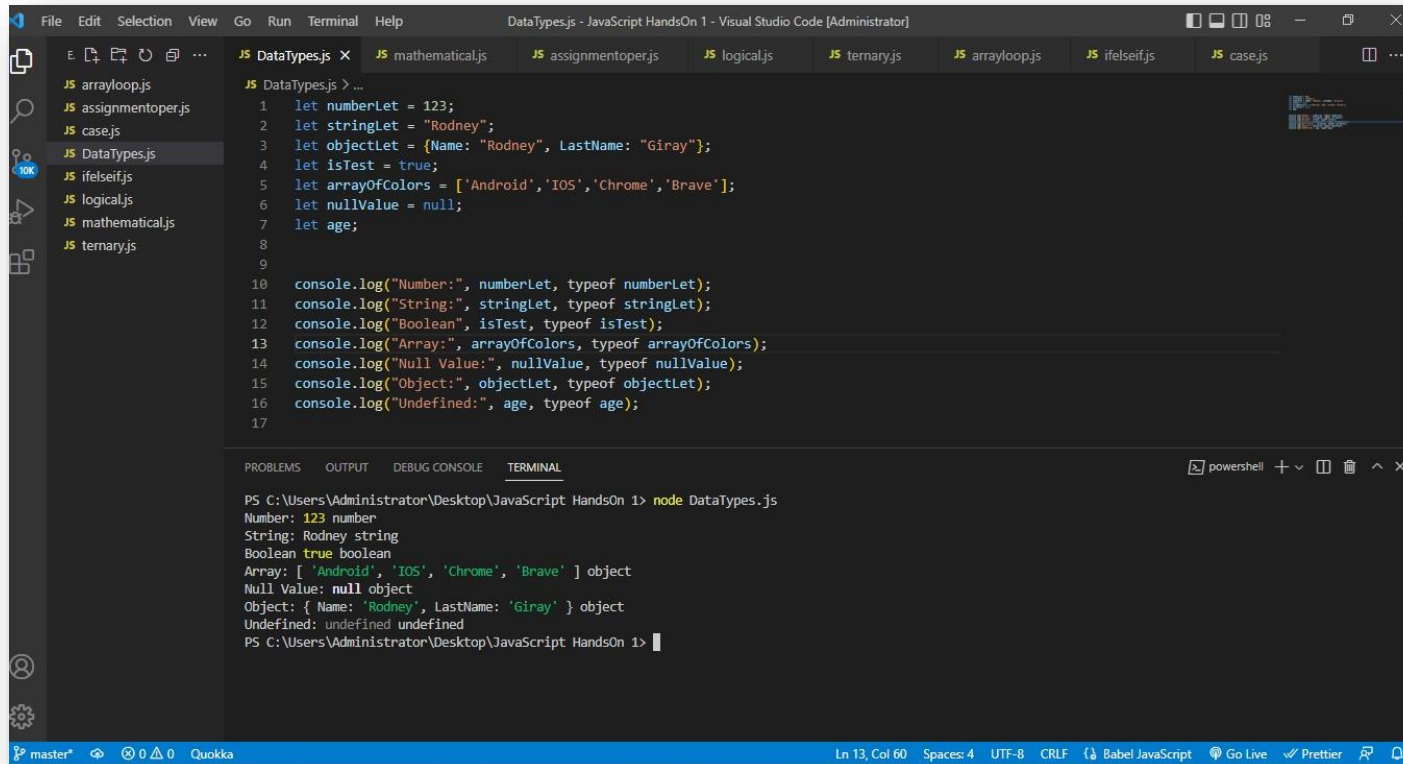
Rodney R. Giray

Instructor: Alfren James Cabuquit

Date: January 1, 2023

WD37

Data Types – I noticed, if we use the `typeof` it will show the type of Data types. In addition, the array is `typeof Data` is Object, it has the same Data Type to Object and displaying output is vertically.



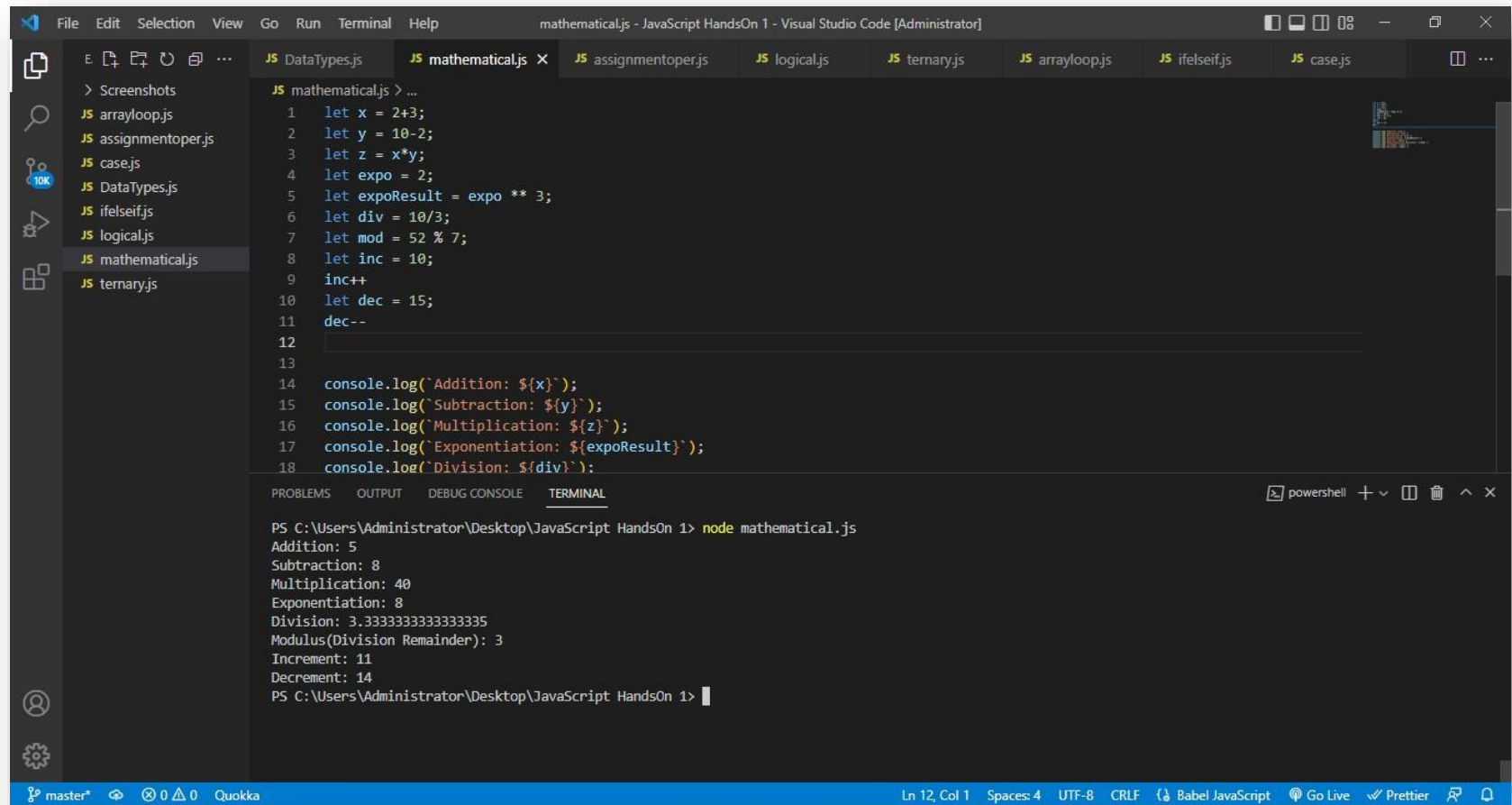
```
File Edit Selection View Go Run Terminal Help
DataTypes.js - JavaScript HandsOn 1 - Visual Studio Code [Administrator]

JS DataTypes.js X JS mathematical.js JS assignmentoper.js JS logical.js JS ternary.js JS arrayloop.js JS ifelseif.js JS case.js
JS arrayloop.js
JS assignmentoper.js
JS case.js
JS DataTypes.js
JS ifelseif.js
JS logical.js
JS mathematical.js
JS ternary.js

1 let numberLet = 123;
2 let stringLet = "Rodney";
3 let objectLet = {Name: "Rodney", LastName: "Giray"};
4 let isTest = true;
5 let arrayOfColors = ['Android', 'IOS', 'Chrome', 'Brave'];
6 let nullValue = null;
7 let age;
8
9
10 console.log("Number:", numberLet, typeof numberLet);
11 console.log("String:", stringLet, typeof stringLet);
12 console.log("Boolean", isTest, typeof isTest);
13 console.log("Array:", arrayOfColors, typeof arrayOfColors);
14 console.log("Null Value:", nullValue, typeof nullValue);
15 console.log("Object:", objectLet, typeof objectLet);
16 console.log("Undefined:", age, typeof age);
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node DataTypes.js
Number: 123 number
String: Rodney string
Boolean true boolean
Array: [ 'Android', 'IOS', 'Chrome', 'Brave' ] object
Null Value: null object
Object: { Name: 'Rodney', LastName: 'Giray' } object
Undefined: undefined undefined
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>
```

Arithmetic Operators – I observed that when using arithmetic operators, it's very accurate even the exact decimals display on the output.



The screenshot shows the Visual Studio Code editor with a file named `mathematical.js` open. The file contains the following JavaScript code:

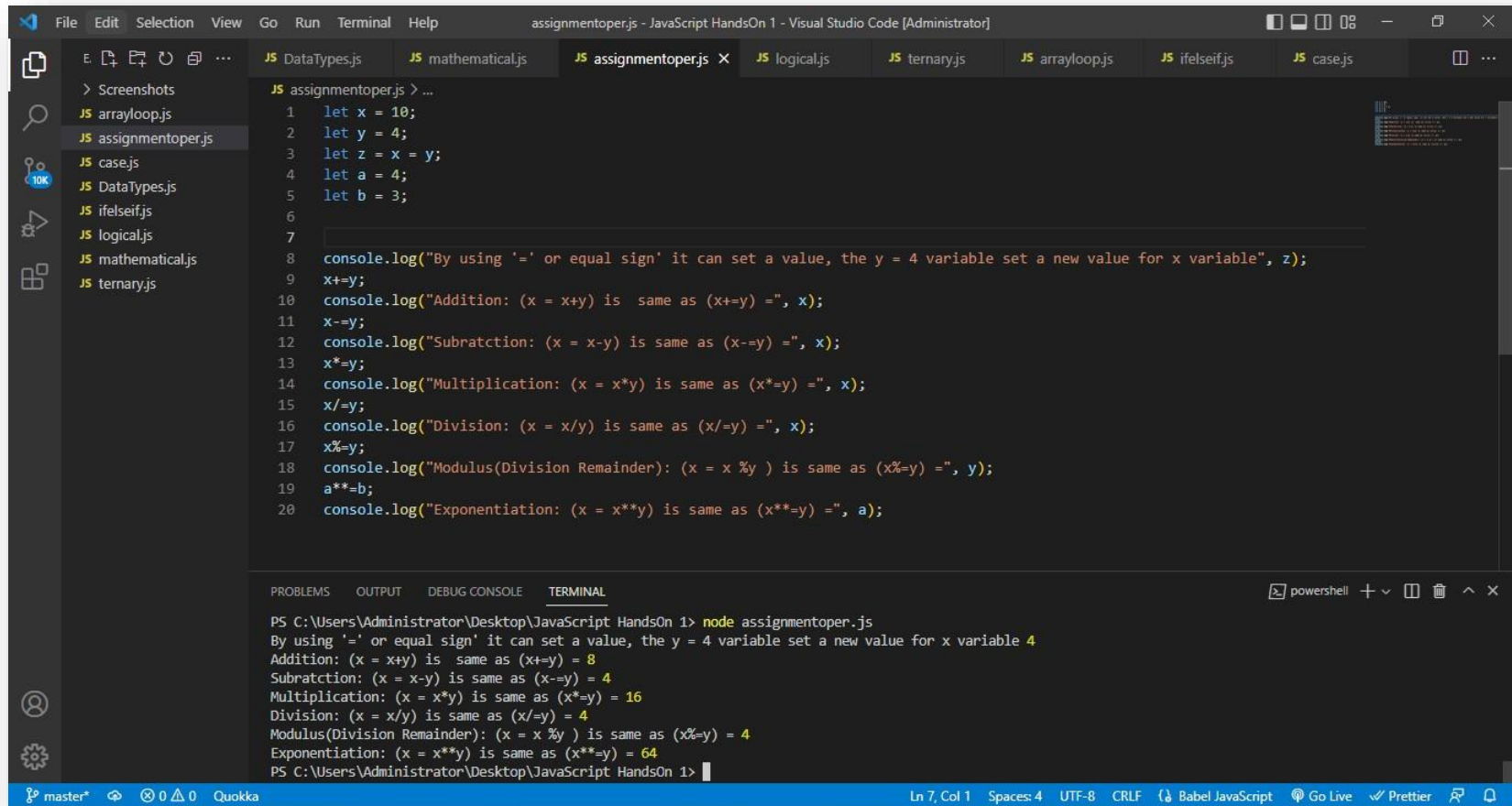
```
1 let x = 2+3;
2 let y = 10-2;
3 let z = x*y;
4 let expo = 2;
5 let expoResult = expo ** 3;
6 let div = 10/3;
7 let mod = 52 % 7;
8 let inc = 10;
9 inc++;
10 let dec = 15;
11 dec--;
12
13
14 console.log(`Addition: ${x}`);
15 console.log(`Subtraction: ${y}`);
16 console.log(`Multiplication: ${z}`);
17 console.log(`Exponentiation: ${expoResult}`);
18 console.log(`Division: ${div}`);
```

The terminal output shows the results of running the code:

```
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node mathematical.js
Addition: 5
Subtraction: 8
Multiplication: 40
Exponentiation: 8
Division: 3.3333333333333335
Modulus(Division Remainder): 3
Increment: 11
Decrement: 14
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>
```

The status bar at the bottom indicates the file is on line 12, column 1, using UTF-8 encoding, CRLF line endings, and the Babel JavaScript parser. It also shows that the Go Live extension is active and the Prettier formatter is configured.

Assignment Operators – I noticed, it easy to use this shorthand for assignment operators and when the value is display it's very accurate.

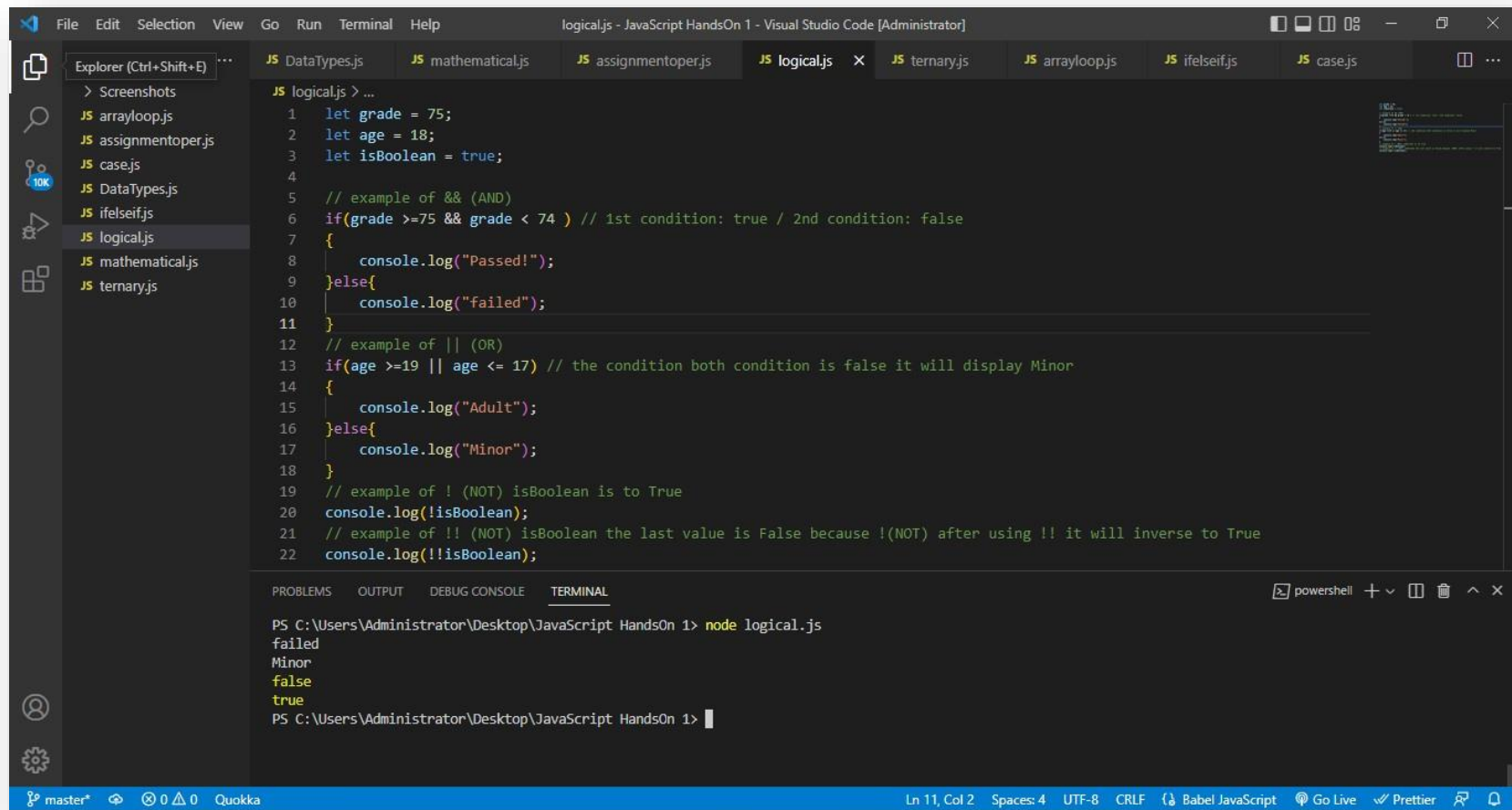


```
File Edit Selection View Go Run Terminal Help assignmentoperator.js - JavaScript HandsOn 1 - Visual Studio Code [Administrator]
JS DataTypes.js JS mathematical.js JS assignmentoperator.js X JS logical.js JS ternary.js JS arrayloop.js JS ifelseif.js JS case.js
> Screenshots
JS arrayloop.js
JS assignmentoperator.js
JS case.js
JS DataTypes.js
JS ifelseif.js
JS logical.js
JS mathematical.js
JS ternary.js

JS assignmentoperator.js > ...
1 let x = 10;
2 let y = 4;
3 let z = x = y;
4 let a = 4;
5 let b = 3;
6
7
8 console.log("By using '=' or equal sign' it can set a value, the y = 4 variable set a new value for x variable", z);
9 x+=y;
10 console.log("Addition: (x = x+y) is same as (x+=y) =", x);
11 x-=y;
12 console.log("Subratction: (x = x-y) is same as (x-=y) =", x);
13 x*=y;
14 console.log("Multiplication: (x = x*y) is same as (x*=y) =", x);
15 x/=y;
16 console.log("Division: (x = x/y) is same as (x/=y) =", x);
17 x%=y;
18 console.log("Modulus(Division Remainder): (x = x %y ) is same as (x%=y) =", y);
19 a**=b;
20 console.log("Exponentiation: (x = x**y) is same as (x**=y) =", a);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node assignmentoperator.js
By using '=' or equal sign' it can set a value, the y = 4 variable set a new value for x variable 4
Addition: (x = x+y) is same as (x+=y) = 8
Subratction: (x = x-y) is same as (x-=y) = 4
Multiplication: (x = x*y) is same as (x*=y) = 16
Division: (x = x/y) is same as (x/=y) = 4
Modulus(Division Remainder): (x = x %y ) is same as (x%=y) = 4
Exponentiation: (x = x**y) is same as (x**=y) = 64
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>
```

Logical Operators – In this activity it was fun to play with Boolean specially using || or NOT operator.



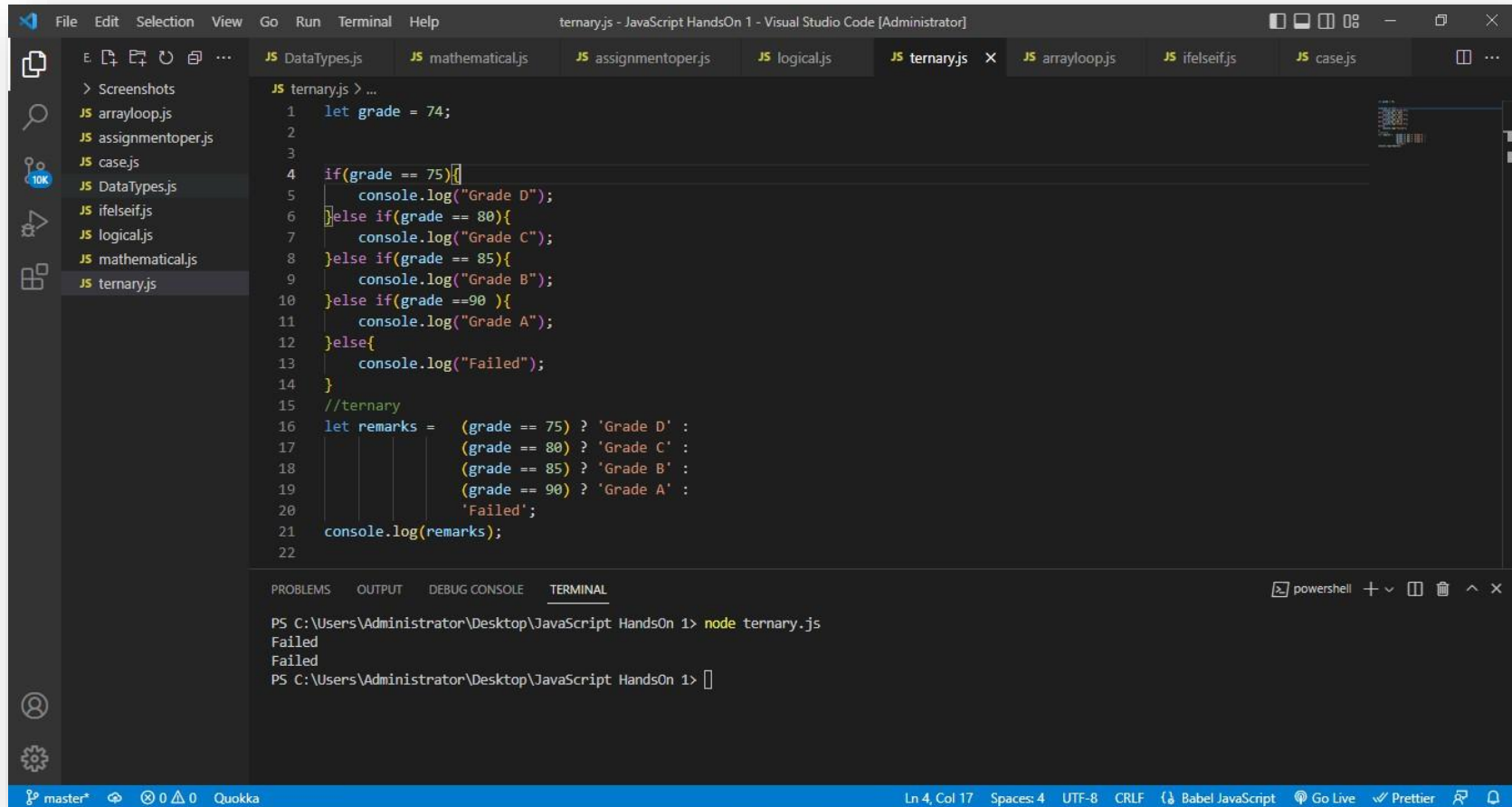
The screenshot shows the Visual Studio Code editor with the file `logical.js` open. The code defines variables `grade` (75), `age` (18), and `isBoolean` (true). It then uses logical operators to check conditions and log results to the console.

```
1 let grade = 75;
2 let age = 18;
3 let isBoolean = true;
4
5 // example of && (AND)
6 if(grade >=75 && grade < 74 ) // 1st condition: true / 2nd condition: false
7 {
8     console.log("Passed!");
9 }else{
10    console.log("failed");
11 }
12
13 // example of || (OR)
14 if(age >=19 || age <= 17) // the condition both condition is false it will display Minor
15 {
16     console.log("Adult");
17 }else{
18     console.log("Minor");
19 }
20
21 // example of ! (NOT) isBoolean is to True
22 console.log(!isBoolean);
23
24 // example of !! (NOT) isBoolean the last value is False because !(NOT) after using !! it will inverse to True
25 console.log(!!isBoolean);
```

The terminal window shows the output of running `node logical.js`:

```
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node logical.js
failed
Minor
false
true
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>
```

If else statement convert to Ternary – In this activity, I'm glad that JavaScript implemented this ternary, it gives developer to choose which one want them to use.



```
File Edit Selection View Go Run Terminal Help
ternary.js - JavaScript HandsOn 1 - Visual Studio Code [Administrator]

JS DataTypes.js JS mathematical.js JS assignmentoper.js JS logical.js JS ternary.js x JS arrayloop.js JS ifelseif.js JS case.js

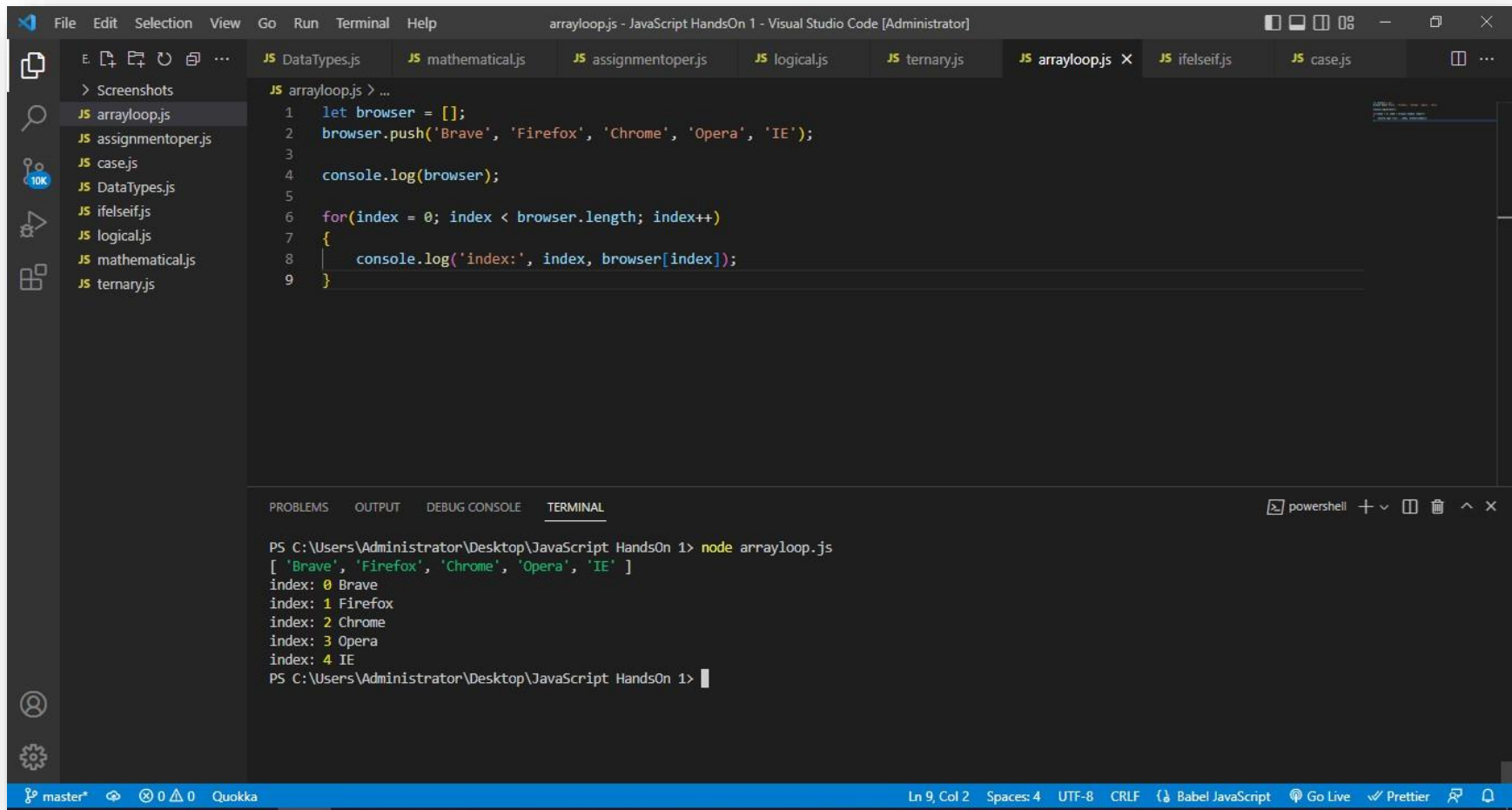
> Screenshots
JS arrayloop.js
JS assignmentoper.js
JS case.js
JS DataTypes.js
JS ifelseif.js
JS logical.js
JS mathematical.js
JS ternary.js

JS ternary.js > ...
1 let grade = 74;
2
3
4 if(grade == 75){
5     console.log("Grade D");
6 }else if(grade == 80){
7     console.log("Grade C");
8 }else if(grade == 85){
9     console.log("Grade B");
10 }else if(grade == 90){
11     console.log("Grade A");
12 }else{
13     console.log("Failed");
14 }
15 //ternary
16 let remarks = (grade == 75) ? 'Grade D' :
17               (grade == 80) ? 'Grade C' :
18               (grade == 85) ? 'Grade B' :
19               (grade == 90) ? 'Grade A' :
20               'Failed';
21 console.log(remarks);
22

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node ternary.js
Failed
Failed
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>

Ln 4, Col 17 Spaces: 4 UTF-8 CRLF Babel JavaScript Go Live Prettier
```

Array and Loop – I noticed, if you want to loop the stack of array you need to use index for default.



The screenshot shows the Visual Studio Code editor with a file named `arrayloop.js` open. The file contains the following JavaScript code:

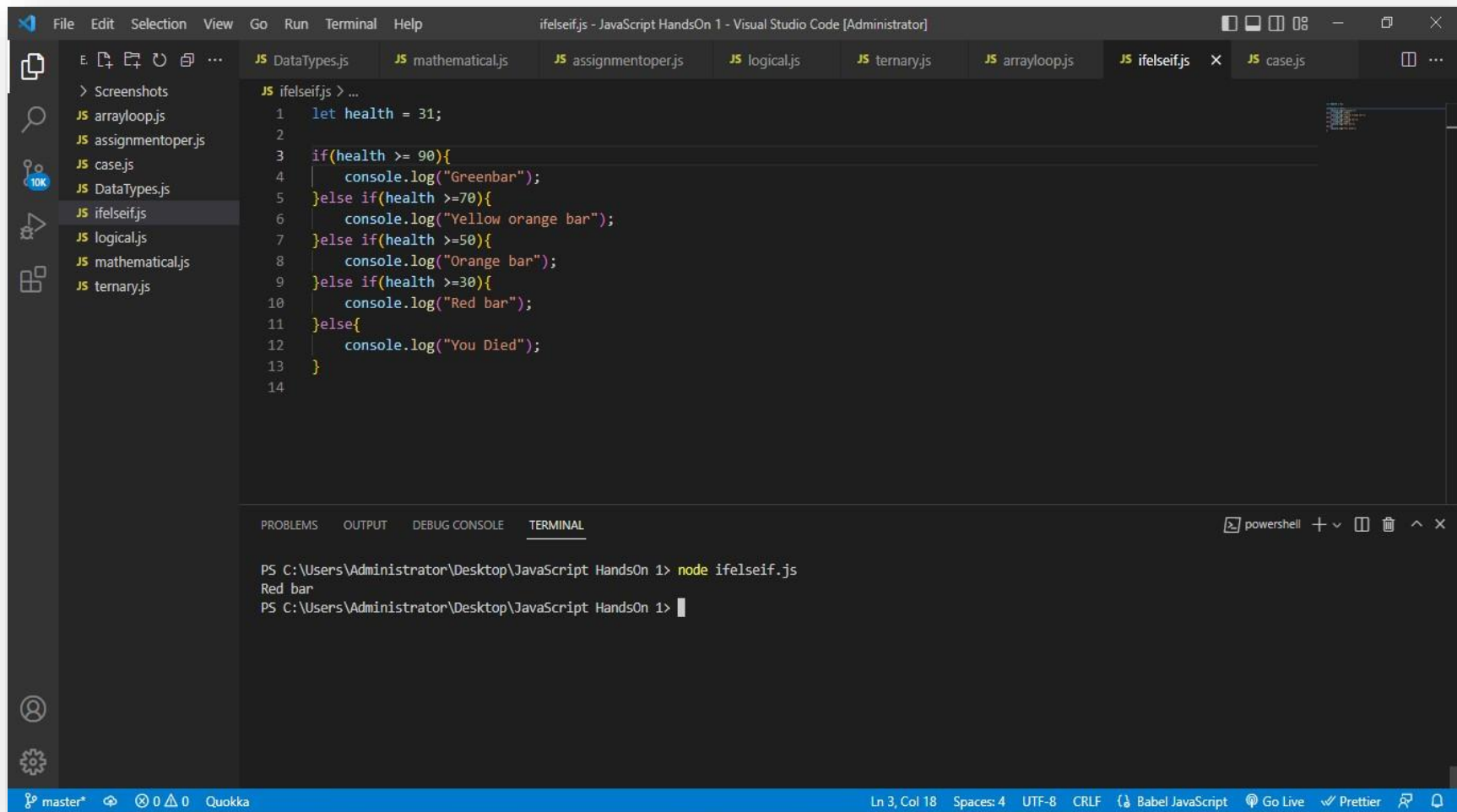
```
1 let browser = [];  
2 browser.push('Brave', 'Firefox', 'Chrome', 'Opera', 'IE');  
3  
4 console.log(browser);  
5  
6 for(index = 0; index < browser.length; index++)  
7 {  
8   console.log('index:', index, browser[index]);  
9 }
```

The terminal output shows the execution of the script:

```
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node arrayloop.js  
[ 'Brave', 'Firefox', 'Chrome', 'Opera', 'IE' ]  
index: 0 Brave  
index: 1 Firefox  
index: 2 Chrome  
index: 3 Opera  
index: 4 IE  
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1>
```

The status bar at the bottom indicates the current line and column (Ln 9, Col 2), the encoding (UTF-8), and the format (CRLF). It also shows the Babel JavaScript extension and the Go Live and Prettier extensions.

If else if – In this activity, by using if and else if it will this the block of code if the statement is true. While else it will display the block of code if the statement is false.



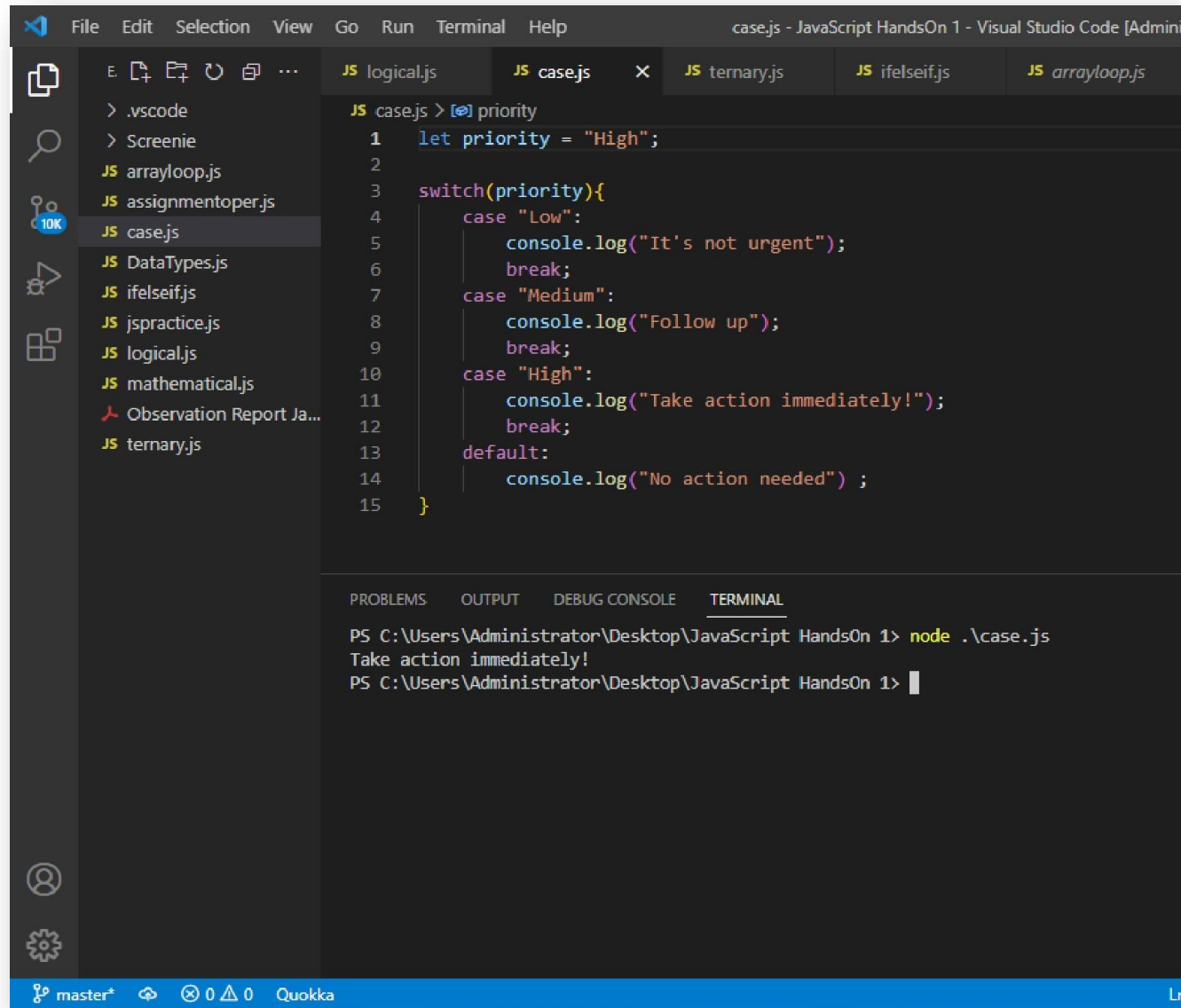
```
File Edit Selection View Go Run Terminal Help ifelseif.js - JavaScript HandsOn 1 - Visual Studio Code [Administrator]
JS DataTypes.js JS mathematical.js JS assignmentoper.js JS logical.js JS ternary.js JS arrayloop.js JS ifelseif.js JS case.js
> Screenshots
JS arrayloop.js
JS assignmentoper.js
JS case.js
JS DataTypes.js
JS ifelseif.js
JS logical.js
JS mathematical.js
JS ternary.js

JS ifelseif.js > ...
1 let health = 31;
2
3 if(health >= 90){
4     console.log("Greenbar");
5 }else if(health >=70){
6     console.log("Yellow orange bar");
7 }else if(health >=50){
8     console.log("Orange bar");
9 }else if(health >=30){
10    console.log("Red bar");
11 }else{
12    console.log("You Died");
13 }
14

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node ifelseif.js
Red bar
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> |
```

master* 0 0 Quokka Ln 3, Col 18 Spaces: 4 UTF-8 CRLF Babel JavaScript Go Live Prettier

Case Statement – I noticed is easy to use but it's limited. It's very useful if you have a specific value or statement you want to choose.



The screenshot shows the Visual Studio Code editor with a file explorer on the left and a terminal at the bottom. The file explorer lists several JavaScript files, with 'case.js' selected. The main editor window displays the code for 'case.js', which uses a switch statement to log messages based on the 'priority' variable. The terminal shows the command 'node .\case.js' being executed, resulting in the output 'Take action immediately!'.

```
case.js - JavaScript HandsOn 1 - Visual Studio Code [Admini]
File Edit Selection View Go Run Terminal Help
JS logical.js JS case.js X JS ternary.js JS ifelseif.js JS arrayloop.js
> .vscode
> Screenie
JS arrayloop.js
JS assignmentoper.js
10K JS case.js
JS DataTypes.js
JS ifelseif.js
JS jspractice.js
JS logical.js
JS mathematical.js
Observation Report Ja...
JS ternary.js

JS case.js > [?] priority
1 let priority = "High";
2
3 switch(priority){
4     case "Low":
5         console.log("It's not urgent");
6         break;
7     case "Medium":
8         console.log("Follow up");
9         break;
10    case "High":
11        console.log("Take action immediately!");
12        break;
13    default:
14        console.log("No action needed") ;
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> node .\case.js
Take action immediately!
PS C:\Users\Administrator\Desktop\JavaScript HandsOn 1> 
```

master* 0 0 Quokka