TLE/ICT 9

Third Quarter

Lesson 4

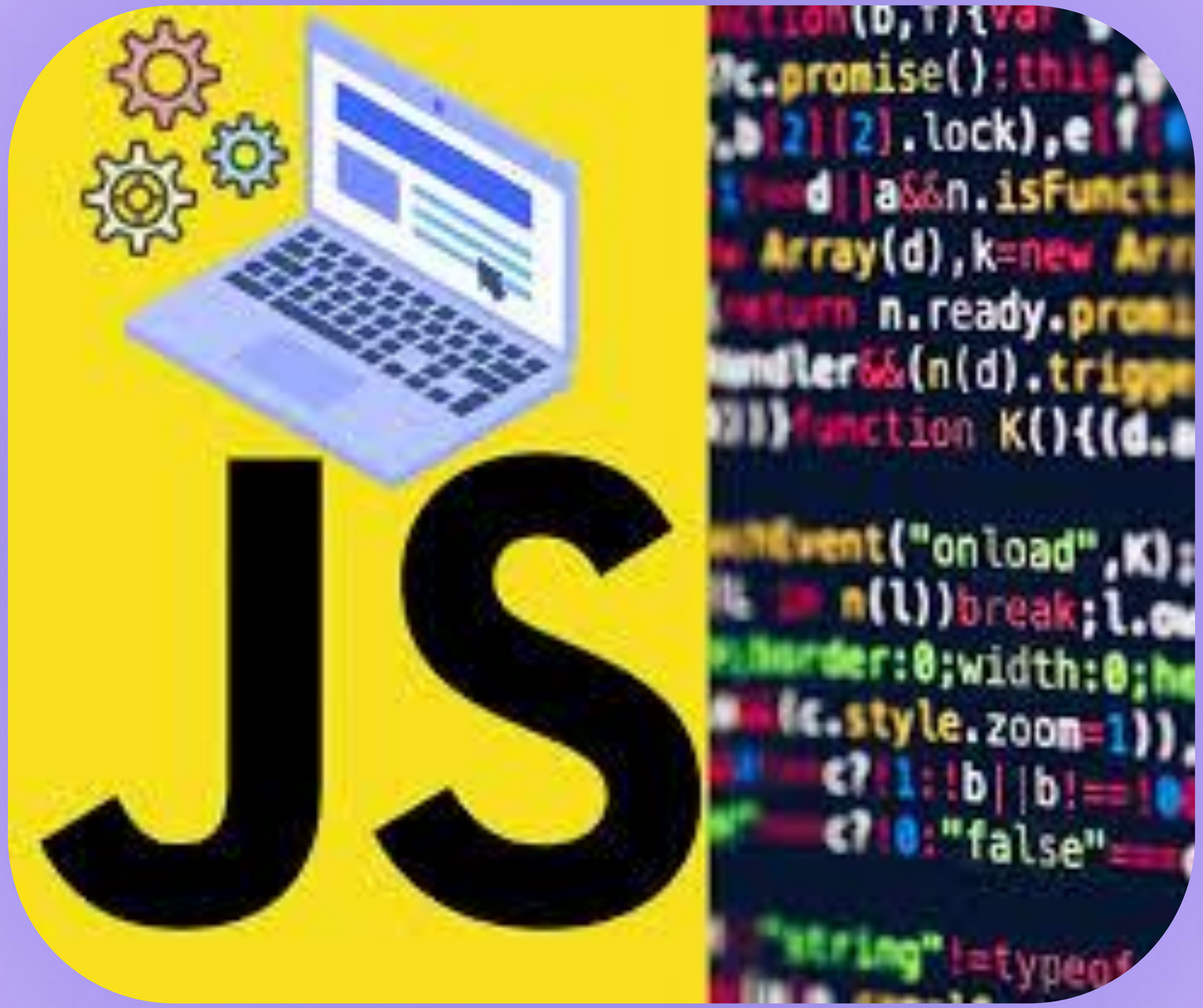

Group
task


FAITH

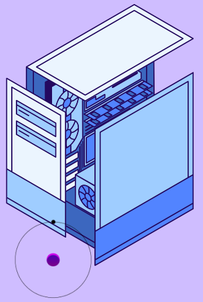

Take turns and share
Play by the rules
FAIRNESS
Avoid blaming others
Listen to the ideas of others


FAMILY

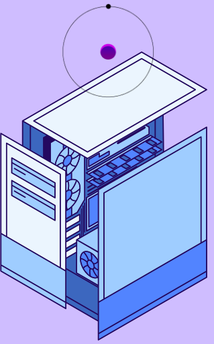

fidelity
shutterstock


JS

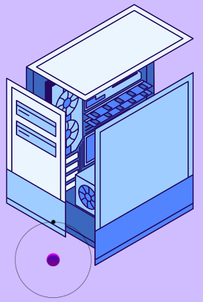
JavaScript Variables



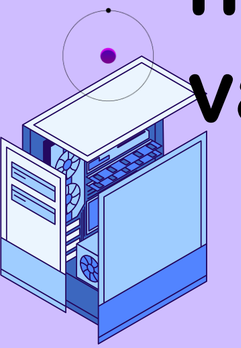
- The purpose of variables are to store information so that they can be used anytime by the program.
- It is a good practice to make it clear when a variable is being used for the first time in the program
- In JavaScript, variables are container that contains a value, which can change as required.

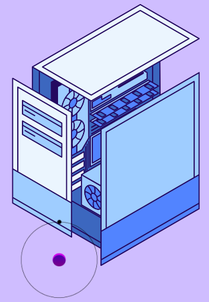


JavaScript Variables



- A variable is a symbolic representation denoting a quantity or expression.
- In mathematics, a variable often represents an “unknown” quantity that has the potential to change.
- In computer science, a variable represents a place where a quantity can be stored. It has a value that has the property of being associated with another value.

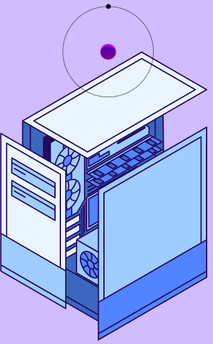




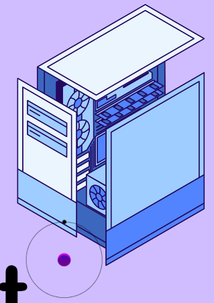
JavaScript Variables

- In JavaScript, variables are container that contains a value, which can change as required.

Example: var VariableName=value;



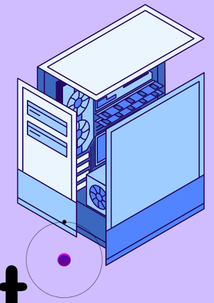
Data Types



Here are some of the data types that a JavaScript variable may contain.

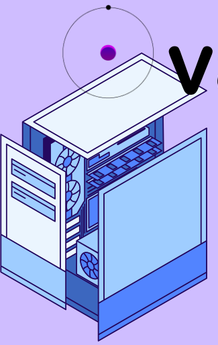
- Character – any single character or symbol such as x, ‘@’, ‘y’ etc.
- String – a series of combination of characters like “somevariable” or “64kgold”
- Integer – number of the set Z={…,-2,-1,0,1,2,…}
- Float (or double) – any decimal number like 97.1 or -97.1

Data Types

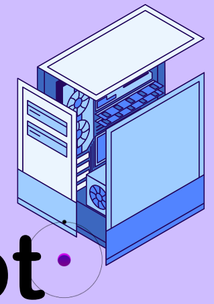


Here are some of the data types that a JavaScript variable may contain.

- Boolean – truth values. This can either be TRUE or FALSE.
- Function – set of instructions or arguments.
- Object – items that exist in the browser that one can manipulate.
- Array – a type of object that contains a set of values that can be mapped into keys.



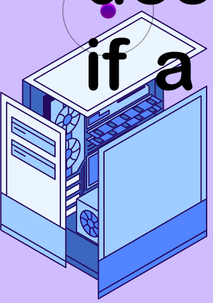
Data Types

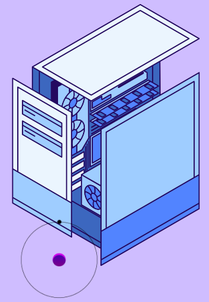


Here are some of the data types that a JavaScript variable may contain.

- Undefined – a variable that has not been given a value yet.
- Null – a variable that has been defined but has been assigned the value of null.

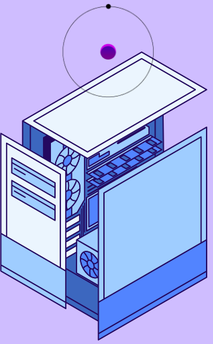
Note: It is important to note that types in JavaScript need not be declared and they may interchange as necessary. This means that if a variable is a string at one minute, it can be an integer the next.





Declaring Variables

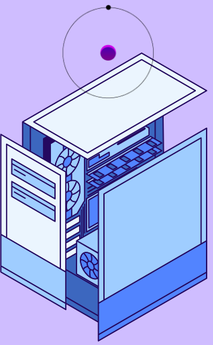
Variables are declared with a *var* statement. It is always good practice to pre-define your variables using *var*. If you declare a variable inside a function, it can be accessed anywhere within the function.



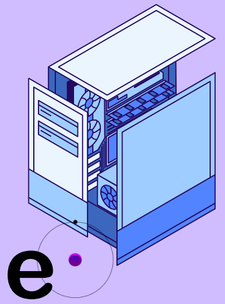
Declaring Variables

The example below illustrates how to declare a variable:

```
// declaring one variable
    var firstname;
//  declaring several variables
    var firstname, lastname;
//  declaring one variable and assigning a value
    var firstname="Jocelyn";
//  declaring several variables and assigning a value
    var firstname="Juan", lastname="Delacruz";
```



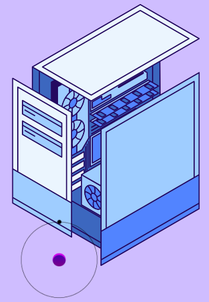
Declaring Variables



JavaScript is fairly lenient, and will create a variable anyway if you do not use the *var* keyword to declare a variable. However, this can lead to problems in your code. Hence, it is always good to use *var*.

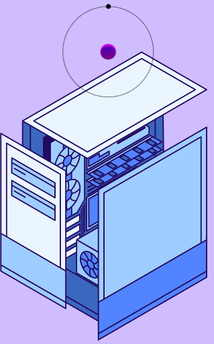
It is also important to note that variables declared outside any function, or used without being declared with ‘var’ are global and can be used by the entire program.

Declaring Variables without Var Command

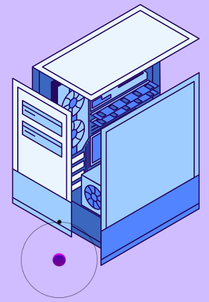


In JavaScript, however, the user can also declare variables implicitly by using the assignment operator to assign a value to the new variable.

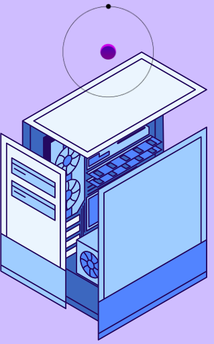
Example: year=1995;
This declares a variable called year and assigns the integer value 1995 to this variable.



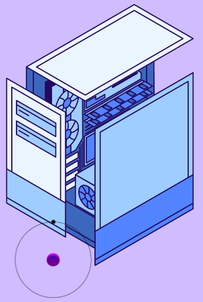
Lifetime of a Variable



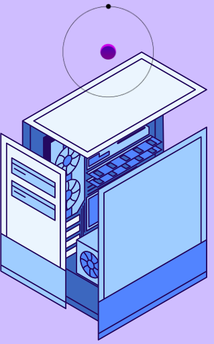
When declaring a variable within a function, the variable can only be accessed within a particular function. When the function exits, the variable is destroyed. These variables are called *local variables.* Local variables can have similar name in different functions, because each is recognized only by the function in which it is declared.



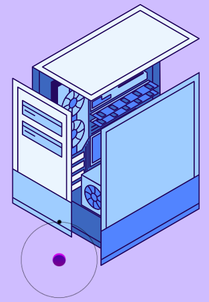
Lifetime of a Variable



When declaring a variable within a function, the variable can only be accessed within a particular function. When the function exits, the variable is destroyed. These variables are called *local variables.* Local variables can have similar name in different functions, because each is recognized only by the function in which it is declared.



Example



var.html

File Edit View

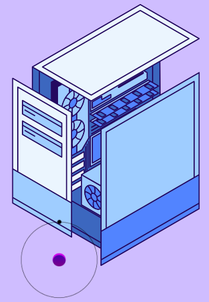
```
<html>
<head><title>JS</title>
<script type="text/Javascript">
        var greet1="Hello"
        var greet2="World!";
        document.write(greet1+""+greet2)
</script>
</head>
<body>
</body>
</html>
```

JS

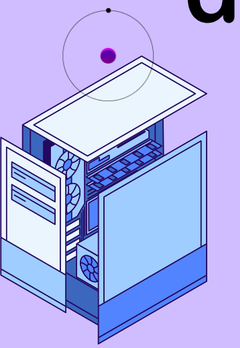
File C:/Users/username/Desktop/var.html

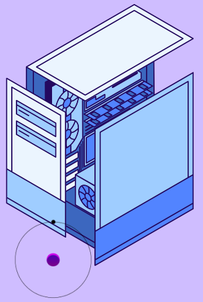
HelloWorld!

What is an Operator?

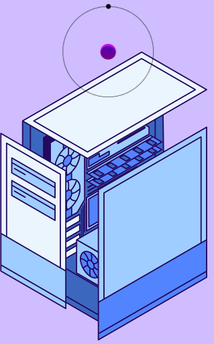


- Normally, you'd use operators in programming to build an expression just as how you build a mathematical expression in your algebra class.
- You use an operator to evaluate (or to operate on) one or more values and then return a value resulting from whatever evaluation or operation it did on the values(s).

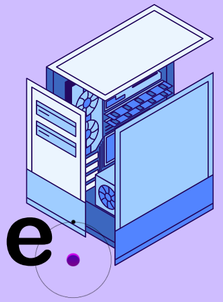




- An expression is anything that has a value. Expressions are important building blocks of JavaScript because you use a lot of these in many of the applications you write in JavaScript.
- One very easy way to understand the concepts of operators in JavaScript is to relate it with the operators in your mathematics class.



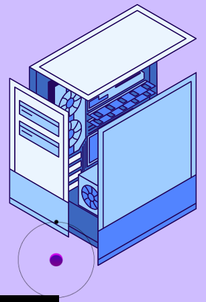
Math/Arithmetic Operators



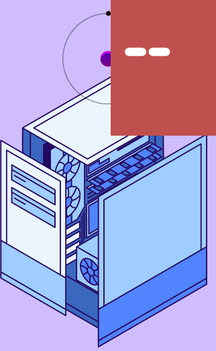
Arithmetic operators in JavaScript are similar to the arithmetic operators in Mathematics, they work exactly in the same manner. Below is a part of the refresher:

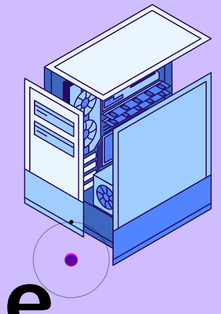
Operator	Name	Example	Result
+	Addition	myVariable=3+4	myVariable=7
-	Subtraction	myVariable=15-7	myVariable=8
*	Multiplication	myVariable=3*5	myVariable=15
/	Division	myVariable=20/4	myVariable=5

Math/Arithmetic Operators

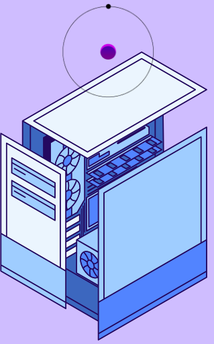


Operator	Name	Example	Result
%	Modulus	myVariable=24%10	myVariable=4
-	Negation	myVariable=-10	myVariable=-10
++	Increment	See example under While statement (Controls and Loops)	
--	Decrement		

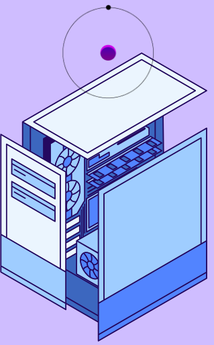




- All operators seem familiar except perhaps the modulus, increment and decrement operations.
- The increment operator ++ simply adds one to the value of your variable while the decrement subtracts one. These operators are normally used within a loop or control structure, which will be discussed later.
- Modulus is simply the remainder of a division operation.



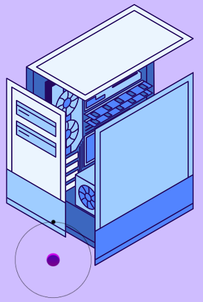
Try this example in your text editor then open in your browser then show the browser output:



mod.html

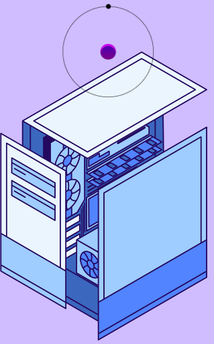
File Edit View

```
<html>
<head><title>mod</title>
</head>
<body>
<script style="text/javascript">
        var num1=37%10;
        document.write(num1)
</script>
</body>
</html>
```



Assignment Operators

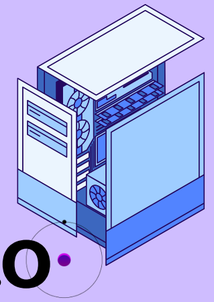
- Assignment operators, except for the “=“ operator, which only assigns value, are used to both assign and perform math operators. It is some sort of shorthand for some less simple math operations.
- Specifically, most of these operators first perform a math operation between two variables and then assign the result to one of the variables.



Operator	Name	
=	Assign var	
+=	Add and Assign	
-=	Subtract and Assi	
*=	Multiply and Assign	x*=y is the same as x=x*y
/=	Divide and Assign	x/=y is the same as x=x/y
%=	Modulus and Assign	x%=y is the same as x=x%y

Similar to increment and decrement operators, you should use these operators within controls or loop structures.

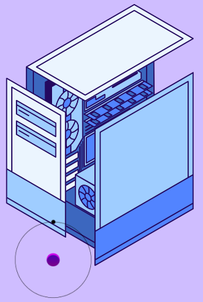
Comparison Operators



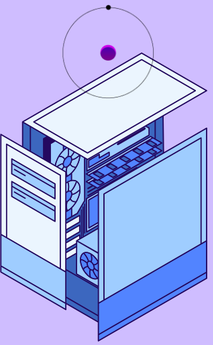
- Comparison operators are used when you want to compare two values. The table lists down the JavaScript comparison operators and some examples.

Operator	Name	Example	Result
==	Is equal to	3==5	False
===	Is identical to (equal and of same type)	2===2	True
!=	Is not equal to	3!=3	False
!==	Is not identical to	4!==3	True

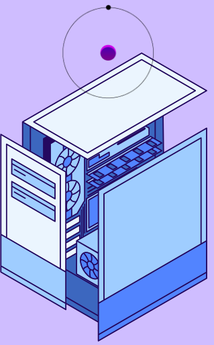
Comparison Operators



Operator	Name	Example	Result
>	Greater than	4>3	True
>=	Greater than or equal to	4>=3	True
<	Less than	4<3	False
<=	Less than or equal to	4<=3	False



Try this example in your text editor then open in your browser then show the browser output:



comparison.txt

File Edit View

```
<html>
<head><title>com</title>
</head>
<body>
<script style="text/javascript">
        var myComparison=10<=8;
        document.write(myComparison)
</script>
</body>
</html>
```

Summary of the Lesson