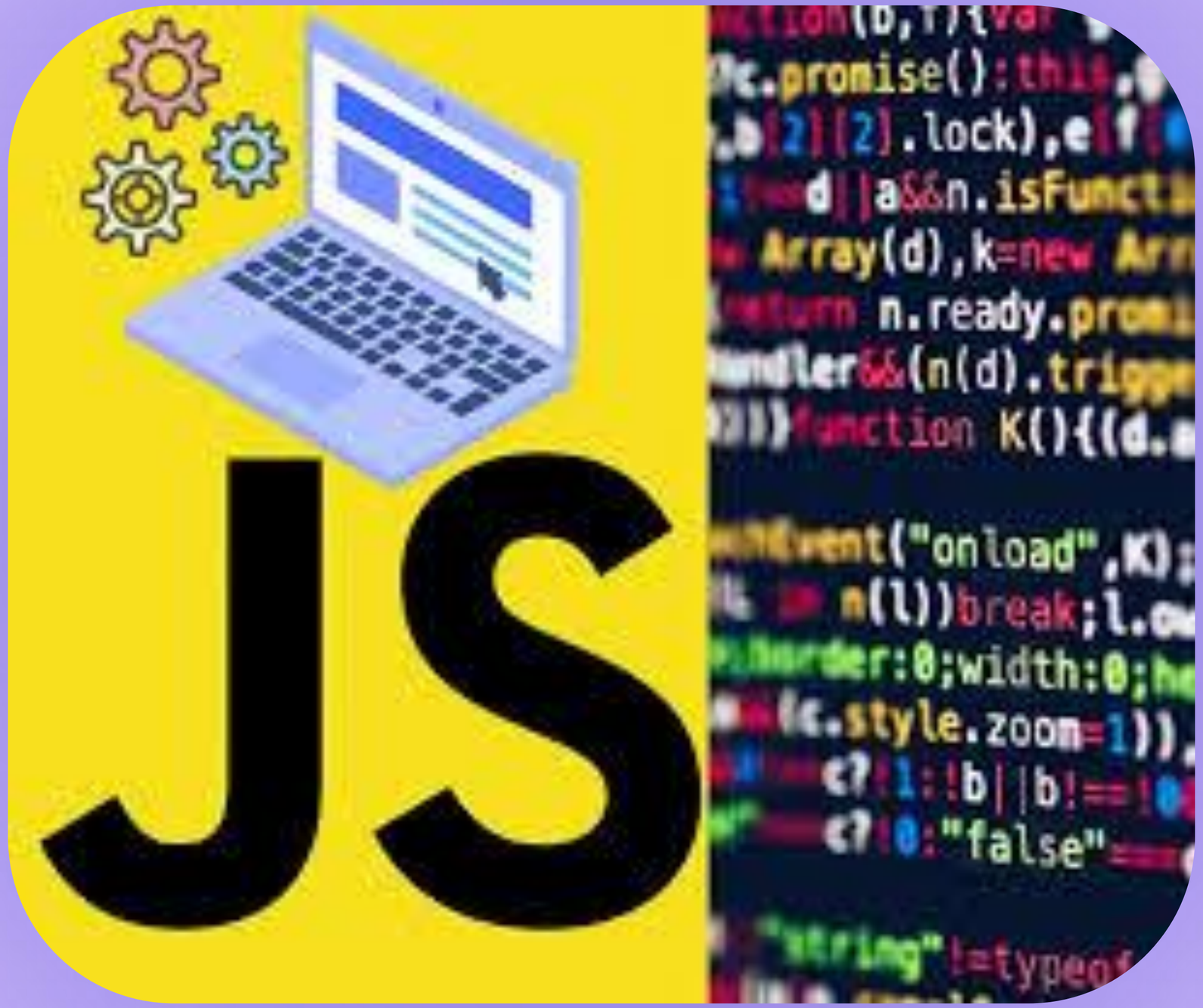
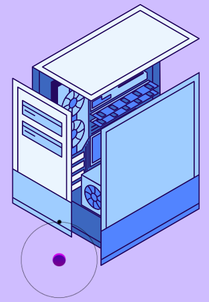
TLE/ICT 9

Fourth Quarter

Lesson 3

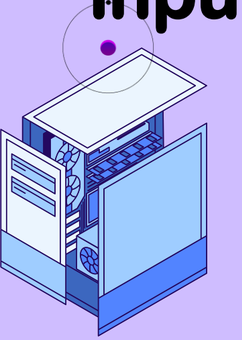

JS

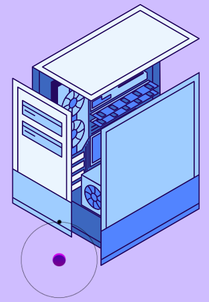


Switch Statement

In the previous example, we used the If Else If statement to test multiple conditions. What if there are a lot of options? The If Else If statement can be tedious. To address this issue, JS offers an easier way to implement multiple conditions, that is through the Switch statement.

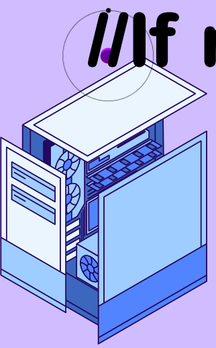
Switch statement is used when a condition may have multiple results and a different set of operation is done based on each result or input.





Syntax:

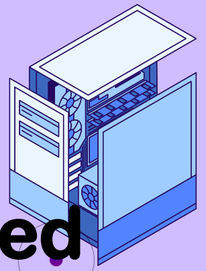
```
switch(condition)
{
case result1:
//operation for result 1

case result2:
//operation for result 2
.
.
default:
//If result belongs to none of the case specified. }
```



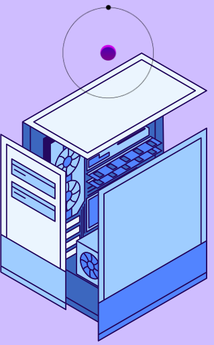
Try this Sample script

```
<!DOCTYPE html>
<head><title>Switch</title>
<script type="text/javascript">
function bookcheck()
{ var book=document.frmbook.cmbbook.value
switch (book)
{case "1": alert("J.R.R Tolkien")
break
case "2": alert("C.S. Lewis")
break
case "3": alert("J.K Rowling")
break
case "4": alert("Mark Twain")
break
default: alert("undetermined")}}
</script></head>
```

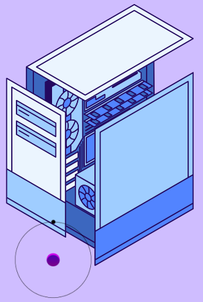
```
<body>
<p>Select a book below to find the name of the author</p>
<form name=frmbook>
<select name=cmbbook>
<option value=1>Lord of the Rings</option>
<option value=2>Chronicles of Narnia</option>
<option value=3>Harry Potter</option>
<option value=4>The Adventures of Huckleberry Finn</option></select>
<input type="button" name="combo" value="SUBMIT" onClick=bookcheck()>
</form>
</body>
</html>
```



In the example, we started by creating a custom function named bookcheck(). Within that function we declared a variable declaring a variable called “book” and using that variable to get the value of the selected item in the combo box. We opened a Switch statement, passing in the variable we want to test which is “book”. This is is followed by a set of cases. The “break” construct after each case prevents the code from running into the next case. Prior to the closing bracket, we placed a default condition. This will be executed if none of the previous cases is true.



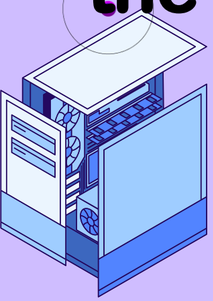
Conditions, Controls and Loops

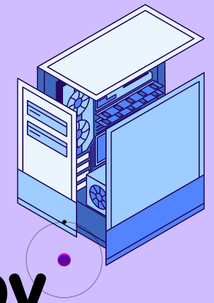


Functions:

A function is a self-contained piece of code that performs a particular “function” when it is called and it is also a set of statement or blocks of codes combined together for a particular use, also known as method. It is usually used so that you will not have to retype your codes again when you need them, all you have to do is to call it and execute it.

A function can be called anywhere within the page or even from an external JS file. Functions can be defined anywhere in the head section or body section; however, it is ideal to put the function in the head section so that you are assured that it is loaded properly.





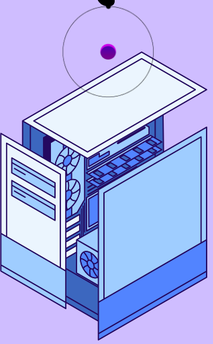
Functions:

JS functions are defined using the keyword “function” followed by its name and then open and close parentheses. This is known as an argument.

Argument provides additional information needed by the function to process.

Syntax:

```
function
functionname(var1,var2,var3…,varN)
{ //Place your codes here }
```

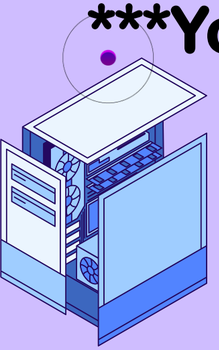


functionname is the name of the function. var1, var2, var3...,VarN are the parameters. These are the values that are passed on to the function whenever it is triggered. Open and close braces are required in functions. A function without parameters should include the open and close parentheses () after the name of the function and remember the function must be in lowercase, otherwise, a JS error will be called.

Remember document.write(), add() and check()? All of these are functions:

document.write() – a predefined function
add() and check() – user-defined functions

***You can check other predefined functions in the internet.



Function.html

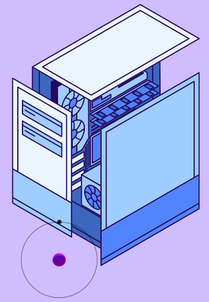
File Edit View

```
<!DOCTYPE html>
<html>
<head><title>Function</title>
<script type="text/javascript">
function displaymessage()
{
alert("Hi there!");
}
</script>
</head>
<body>
<form>
<input type="button" value="Click Me!" onClick="displaymessage()"/>
</form>
</body>
</html>
```

function1.txt

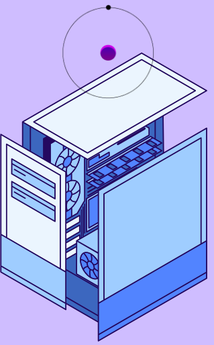
File Edit View

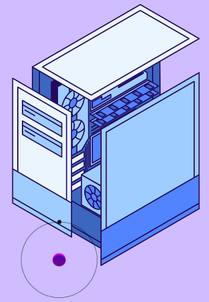
```
<!DOCTYPE html>
<html>
<head><title>function</title>
<script type="text/javascript">
function add()
{var A=Number(document.example.firstno.value);
var B=Number(document.example.secondno.value);
var C=A+B;
document.example.sumno.value=C;
}
</script>
</head>
<body>
<form name="example">
<input type="type" name="firstno" value="">+
<input type"type"  name="secondno" value="">=
<input type="type" name="sumno" value="">
<br>
<input type="button" name="addno" value="Submit" onClick=add()>
</form>
</body>
</html>
```



Note:

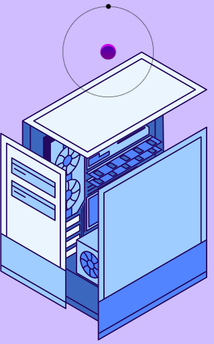
Variables that are declared within a function will not last outside the function. If you want to save or reuse the value of the variable, you can use a return function. You can also put a return statement in the function so that the value stored while using the function can be returned.





The return Statement

A function may require specifying a resulting output, for example that of a computation, the return statement will be used to give that output. Therefore, functions that give out a value must use the return statement.



Return.txt

File Edit View

```
<!DOCTYPE html>
<html>
<head><title>Return</title>
<script type="text/javascript">
function product(x,y)
{
return x*y;
}
</script>
</head>
<body>
<script language="javascript">
document.write("The product of x and y is");
document.write(product(5,7));
</script>
</body>
</html>
```