TLE/ICT 9

Third Quarter

Lesson 5

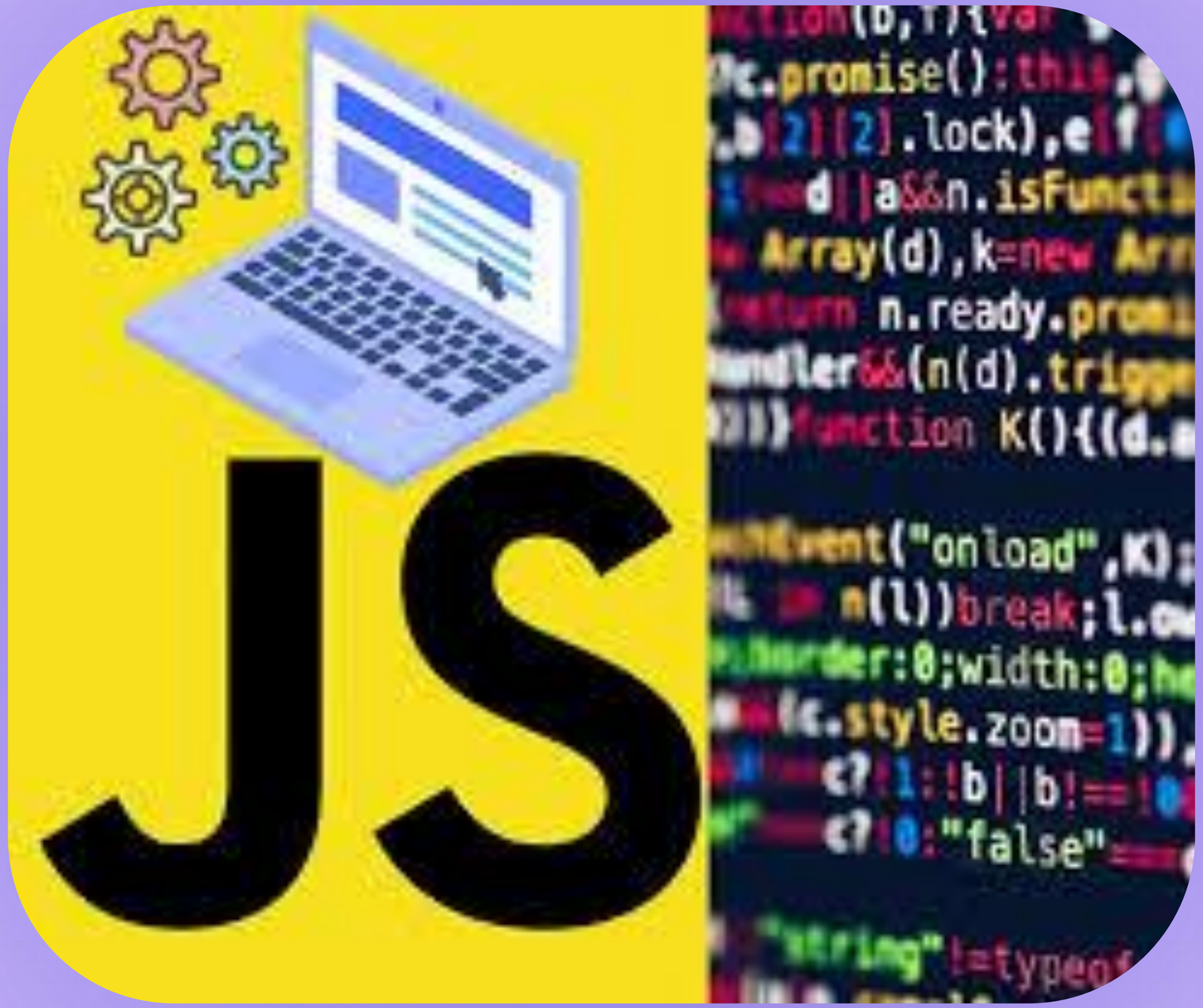

Group
task


FAITH


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

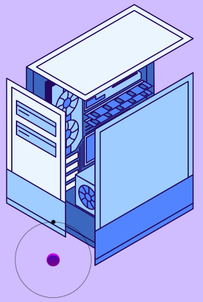

FAMILY


fidelity
shutterstock


JS

JavaScript Operators

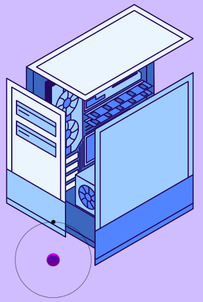
The set of JavaScript operators are the same



Operator	Name	Example Result
==	Returns a true value if the items are the same	Counter==10> Returns the value ”true” if the counter’s value is currently equal to the number 10.
+=	Shortcut for adding to the current value	clicks +=2 Sets the variable named counter to equal to the current value plus two.
-=	Shortcut for subtracting to the current value	clicks -=2 Sets the variable named counter to equal to the current value minus two.
*=	Shortcut for multiplying to the current value	clicks *=2 Sets the variable named counter to equal to the current value multiplied by two.

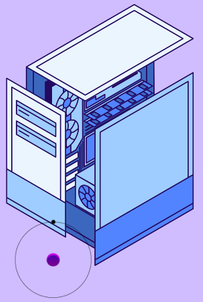
JavaScript Operators

The set of JavaScript operators are the same



Operator	Name	Example Result
=	Sets one value equal to another	counter=0 Sets the counter equal to the number 0
&&	Looks at two expressions and returns a value of “true” if both expressions are true.	if day=‘friday’&&date=13 then alert(“Are A user Superstitious?”)
||	Looks at two expressions and returns a value of “true” if either one-but not both-of the expressions are true.	if day=‘friday’&&date=13 then alert(“Are A user Superstitious?”) else if day=‘friday’||date=13 then alert(“Aren’t a user glad it isn’t Friday the 13th?”)

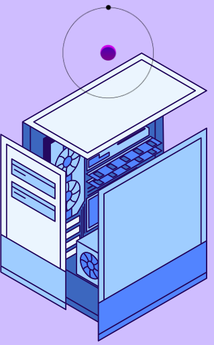
JavaScript Operators



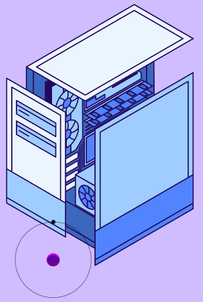
The most common operation performed with strings is concatenation.

Concatenation – is the process of combining two string texts into one larger string.

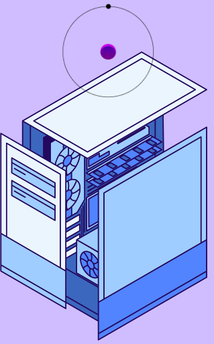
Example: combining the strings “Scripting” and “Master” into a new string as “Scripting Master”.



JavaScript Operators



What symbol is used? The ”+” operator is used. Note this symbol is also used as a mathematical addition operator in JavaScript. So using the “+” with numerical values, it will add the two values; and using the same operator with two strings, it will concatenate (combine into one) the two strings.



JavaScript Operators

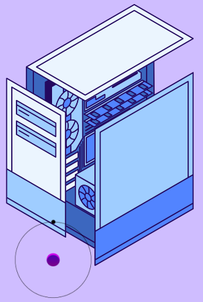
Concatenate.txt

File Edit View

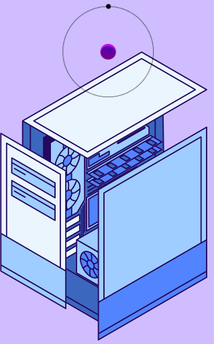
```
<html>
<head><title>Concatenation</title></head>
<body>
<script language="javascript">
var String1,String2,StringConcatenated;
String1="peanut";//first string
String2="butter";//second string
StringConcatenated=String1+String2;//Concatenating Strings
document.write(StringConcatenated);//printing the Concatenated String
</script>
</body>
</html>
```

Note: there are 3 string variables

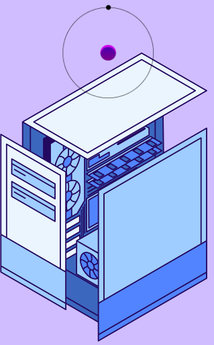
JavaScript Operators



Note: a string is enclosed in either single or double quotation marks.
On line 5, StringConcatenated=String1+String2;, The ”+” operator concatenates two strings. Before the concatenation, there are two Strings: “peanut” and “butter”. After the concatenation, there was a combined longer String: “peanutbutter”.



Another code of a more complex example of the concatenate operator:

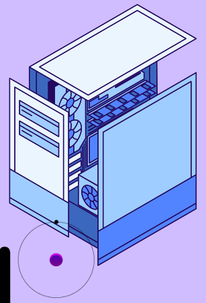


complex.html

File Edit View

```
<html>
<head><title>Complex Concatenation</title>
</head>
<body>
<script language="javascript">
var strOut;
strOut="<table width='300' border='1' cellspacing='0' cellpadding='3'>";
strOut=strOut+"<tr>";
strOut=strOut+"<th colspan='2'>Compound Words Use Concatenation</th>";
strOut=strOut+"</tr>";
strOut=strOut+"<tr>";
strOut=strOut+"<th>First Word+Second Word</th>";
strOut=strOut+"<th>Result</th>";
strOut=strOut+"</tr>";
strOut=strOut+"<tr>";
strOut=strOut+"<td>back + ground</td>";
strOut=strOut+"<td>background</td>";
strOut=strOut+"</tr>";
strOut=strOut+"<tr>";
strOut=strOut+"<td>basket + ball</td>";
strOut=strOut+"<td>basketball</td>";
strOut=strOut+"</tr>";
strOut=strOut+"</table>";
document.write(strOut);
</script>
</body>
</html>
```

Explanation of the sample code



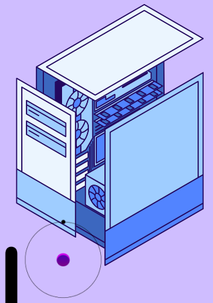
In the previous example, only one variable was used which is the strOut.

A complex String is assigned to variable strOut. Notice single quotes were used for attribute values of a table tag. Then the String “<tr>” is concatenated with the String “<table width=‘300’ border=‘1’ cellspacing=‘0’ cellpadding=‘3’>” and results in:

<table width=‘300’ border=‘1’ cellspacing=‘0’ cellpadding=‘3’>

<tr>

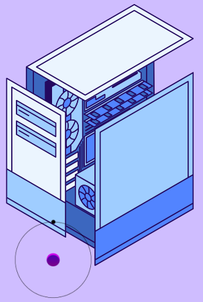
JavaScript Functions



A *function* is a piece of code that stays dormant until it is referenced or called upon to do its purpose. In addition to controllable execution, functions are also a great time saver for doing repetitive tasks.

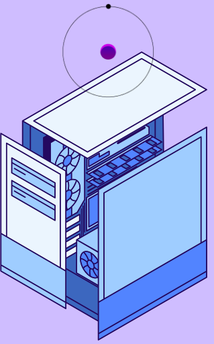
Instead of typing out the same code snippet every time a user needed it. It was made simpler by declaring a function which can be called multiple times yet getting the same effect. This benefit is also known as “code reusability”.

JavaScript Functions

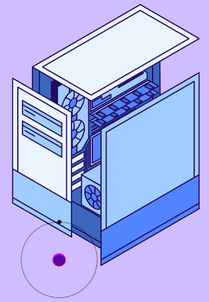


A function that does not execute when the page loads should be placed inside the head of your HTML document. Creating a function is really easy:

1. Browsers must be told when making a function.
2. Function name must be given.
3. Write the JavaScript like normal.

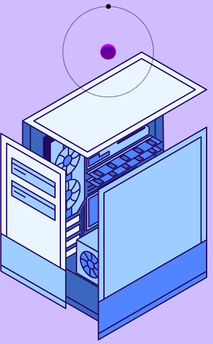


JavaScript Functions



JavaScript functions are defined using the keyword “function” followed by its name and variables being passed to it in the syntax:

```
function fname(value1,value2,...)
{statement1
  statement2}
```

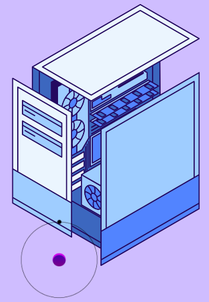


function.txt

File Edit View

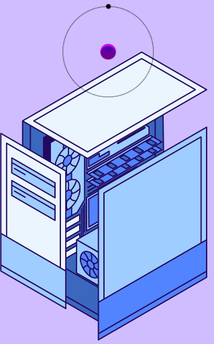
```
<html>
<head><title>Function</title>
<script type="text/javascript">
<!--
function popup()
{
alert("Welcome to My Website")
}
//-->
</script>
</head>
<body>
<input type="button" onclick="popup()" value="popup">
</body>
</html>
```

Explanation



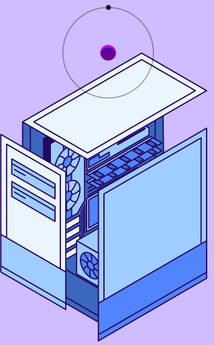
In the given example, the browser was told that a function is used by typing “function”. Next, the function is named “popup” since the program intends to show popup boxes.

The curly braces”{ }” define the boundaries of the function code. All popup function code must be contained within the curly braces.

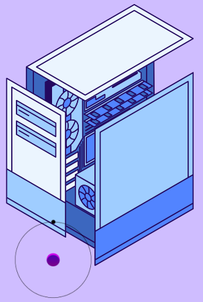


Explanation

Something that might be slightly confusing is that within the “popup” function, there is another function called “alert”, which brings up a popup box with the text that was set. It is perfectly OK to use functions within functions, like what was done. This is one of the great things about using function.

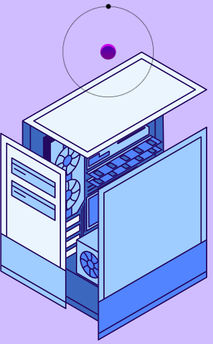


Returning a Value

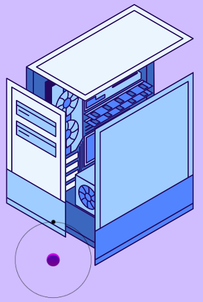


Syntax:

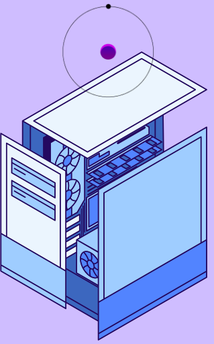
```
function name(parameter1,parameter2,...)
{
//set of statements that will be executed
return value;
}
```



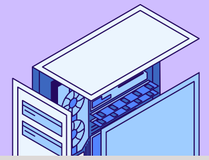
Returning a Value



A function can also return a value after doing the specified operations. In the example that will be shown, the requirement is to calculate $x^2/2$. After the value is returned, this will divide it by 2. Then the final answer would appear.



Returning a Value



value.html

File Edit View

```
<html>
<head>
<title>Function</title>
</head>
<body>
<script language="javascript">
function square(number1)
{
var num=number1*number1;
//Now we will return the result
return num;
}
var i=6;
//here we invoke the function and capture the result
var des=square(i);
var res=des/2;
document.write("The result -"+res);
</script>
</body>
</html>
```

In the example, we calculate x^2 in the function “square(number1)” and then we returned the result.

Getting the Result

As the function returns a value while invoking the function, a variable is assigned to capture the result as “var des=square(i);”.

The result of x^2 is assigned to the variable des.

Using the variable ‘des’ further operations were completed.

Note: Once the return statement is called in a function, it will break (ex. no further statements below it will be executed.)

Summary of the Lesson