Variables and Datatypes in JavaScript

The latest ECMAScript(ES6) standard defines seven data types: Out of which six data types are Primitive(predefined).

- **Numbers**: 5, 6.5, 7 etc.
- **String**: “Hello Internity” etc.
- **Boolean**: Represent a logical entity and can have two values: true or false.
- **Null**: This type has only one value : *null.*
- **Undefined**: A variable that has not been assigned a value is *undefined.*
- **Object**: It is the most important data-type and forms the building blocks for modern JavaScript. We will learn about these data types in details in further articles.

Variables in JavaScript:

Variables in JavaScript are containers which hold reusable data. It is the basic unit of storage in a program.

- The value stored in a variable can be changed during program execution.
- A variable is only a name given to a memory location, all the operations done on the variable effects that memory location.
- In JavaScript, all the variables must be declared before they can be used.

Before ES2015, JavaScript variables were solely declared using the *var* keyword followed by the name of the variable and semi-colon.
Below is the syntax to create variables in JavaScript:

```
var  var_name;
var  x;
```

The var_name is the name of the variable which should be defined by the user and should be unique. These type of names are also known as **identifiers**. The rules for creating an identifier in JavaScript are, the name of the identifier should not be any pre-defined word(known as keywords), the first character must be a letter, an underscore (_), or a dollar sign ($). Subsequent characters may be any letter or digit or an underscore or dollar sign.

Functions in JavaScript

A function is a set of statements that take inputs, do some specific computation and produces output. Basically, a function is a set of statements that performs some tasks or does some computation and then returns the result to the user.

The idea is to put some commonly or repeatedly done task together and make a function so that instead of writing the same code again and again for different inputs, we can call that function.

Like other programming languages, JavaScript also supports the use of functions. You must already have seen some commonly used functions in JavaScript like alert(), this is a built-in function in JavaScript. But JavaScript allows us to create user-defined functions also.

We can create functions in JavaScript using the keyword *function*. The basic syntax to create a function in JavaScript is shown below.
Syntax:

```
function functionName(Parameter1, Parameter2, ..)

{  // Function body }
```

To create a function in JavaScript, we have to first use the keyword *function*, separated by name of function and parameters within parenthesis. The part of function inside the curly braces {} is the body of the function.

Function Definition

Before, using a user-defined fuction in JavaScript we have to create one. We can use the above syntax to create a function in JavaScript. Function definition are sometimes also termed as function declaration or function statement.

Below are the rules for creating a function in JavaScript:

- Every function should begin with the keyword *function* followed by,
- A user defined function name which should be unique,
- A list of parameters enclosed within paranthesis and separated by commas,
- A list of statement composing the body of the function enclosed within curly braces {}.

Example:

```
function calcAddition(number1, number2)
{
   return number1 + number2;
}
```

In the above example, we have created a function named calcAddition, this function accepts two numbers as parameters and returns the addition of these two numbers.

Function Parameters

Till now we have heard a lot about function parameters but haven\'t discussed them in details. Parameters are additional information passed to a function. For example, in the above example, the task of the function *calcAddition* is to calculate addition of two numbers. These two numbers on which we want to perform the addition operation are passed to this function as parameters. The parameters are passed to the function within parentheses after the function name and separated by commas. A function in

JavaScript can have any number of parameters and also at the same time a function in JavaScript can not have a single parameter.

Calling Functions: After defining a function, the next step is to call them to make use of the function. We can call a function by using the function name separated by the value of parameters enclosed between parenthesis and a semicolon at the end. Below syntax shows how to call functions in JavaScript:

```
functionName( Value1, Value2, ..);
```

Below is a sample program that illustrate working of functions in JavaScript:

```
<script type = "text/javascript">

// Function definition
function welcomeMsg(name) {
  document.write("Hello " + name + " welcome to Internity");
}

// creating a variable
var nameVal = "Admin";

// calling the function
welcomeMsg(nameVal);

</script>
```

Output:

```
Hello Admin welcome to Internity
```

Return Statement: There are some situations when we want to return some values from a function after performing some operations. In such cases, we can make use of the return statement in JavaScript. This is an optional statement and most of the times the last statement in a JavaScript function. Look at our first example with the function named as *calcAddition*. This function is calculating two numbers and then returning the result. The most basic syntax of using the return statement is:

```
return value;
```

The return statement begins with the keyword *return* separated by the value which we want to return from it. We can use an expression also instead of directly returning the value.