

4.5 Variables

We will begin the programming using JavaScript by learning about the primitives, operations and expressions.

4.5.1 Primitive Types

JavaScript defines two entities primitives and objects. The primitives are for storing the values whereas the object is for storing the reference to the actual value.

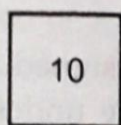
There are following primitive types used in JavaScript

1. Number
2. String
3. Boolean
4. Undefined
5. Null

There are three type of predefined objects in JavaScript

1. Number
2. String
3. Boolean

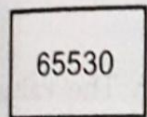
These objects are called **wrapper objects**. These wrapper objects provide properties and methods which can be used by primitive types.



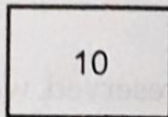
X

X = 10;

Fig. 4.1 (a) Representation of primitive type



obj



65530

obj contains the address at which the value is stored.

Fig. 4.1 (b) Representation of object

4.5.2 Literals

There are two types of literals used in JavaScript and those are numeric literals and string literals. The numeric literals are called **numbers**. These numbers can include integer values, floating point or double precision values. For example -

10
10.3
10.0
10.
10E3
10.2E4
10.e2
10e-3

are all valid numeric literals.

The string literals are the sequence of characters. It can be written in double quotes " " or in single quotes ' '. For example

'Rain Rain come soon'

4.5.3 Other Primitive Types

The other primitive types are

- Boolean

The Boolean values are **true** and **false**. These values can be compared with the variables or can be used in assignment statement.

- Null

The null value can be assigned by using the reserved word **null**. The **null** means no value. If we try to access the **null** value then a runtime error will occur.

- Undefined

If a variable is explicitly declared and not assigned any value to it then it is an undefined value. If we try to display the undefined value then on the browser the word "undefined" will be displayed. There is no reserved word for undefined primitive type.

4.5.4 Variable Declaration

In JavaScript we can declare the variable using the reserved word **var**. The value of this variable can be any thing; it can be numeric or it can be string or it can be a Boolean value.

JavaScript[VarDemo.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Variables in Javascript </title>
</head>
<body>
<script type="text/javascript">
var a,b,c;
var string;
a=2;
b=3;
c=a+b;
string="The result = ";
```

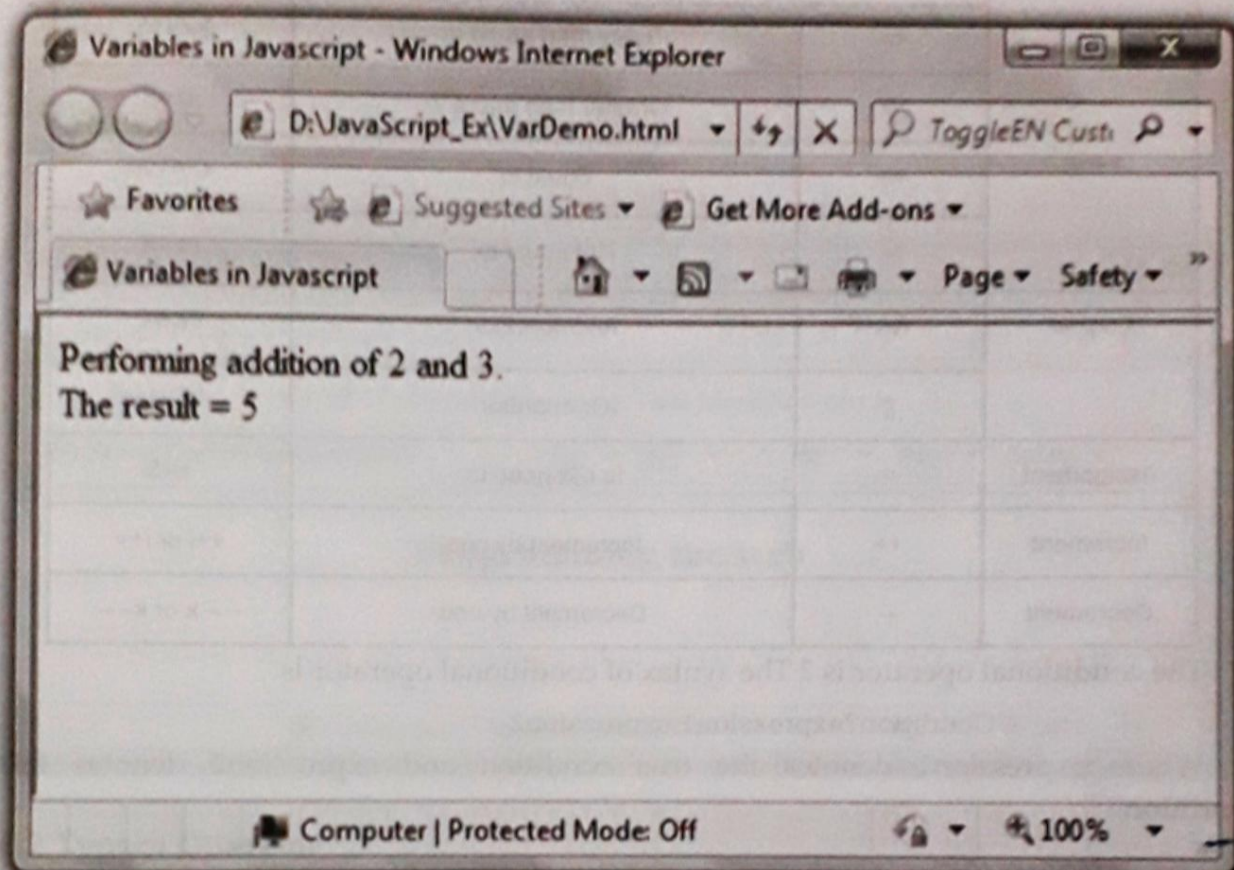
Variable declaration is done using **var**. Note that there is no data type required for handling variables.


```

document.write("Performing addition of 2 and 3. "+"<br/>");
document.write(string);
document.write(c);
</script>
</body>
</html>

```

Output



Note that using `var` we can define the variable which is of type numbers(2 , 3 or 5) as well as the string "The result".

4.6 Operators

Various operators used by JavaScript are as shown in following table -

Type	Operator	Meaning	Example
Arithmetic	+	Addition or unary plus	c=a+b
	-	Subtraction or unary minus	d= -a
	*	Multiplication	c=a*b

	/	Division	c=a/b
	%	Mod	c=a%b
Relational	<	Less than	a<4
	>	Greater than	b>10
	<=	Less than equal to	b<=10
	>=	Greater than equal to	a>=5
	==	Equal to	x==100
	!=	Not equal to	m!=8
Logical	&&	And operator	0&&1
		Or operator	0 1
Assignment	=	Is assigned to	a=5
Increment	++	Increment by one	++i or i++
Decrement	--	Decrement by one	--k or k--

The conditional operator is ? The syntax of conditional operator is

Condition?expression1:expression2

Where expression1 denotes the true condition and expression2 denotes false condition.