

NAME - SHREYA RAKSHIT

COLLEGE - IIT (ISM) DHANBAD

DOMAIN - FULL STACK WEB DEVELOPMENT

TASK NO. - 1

Importance Of Data Types, Variables, And Operators In Javascript

Data types:

Data type is an extremely important component to deal with various operations. Data type basically decides how a variable should be treated in terms of various operations. As JavaScript has dynamic data types, a single variable can hold different types of data at different moment. A variable basically behaves depending upon its current data type. Data Types of JavaScript are as follows:

- 1. String:** String is a series of characters that can hold any name, place, paragraph etc. It is extremely useful to display something on the website.
- 2. Number:** Number datatype basically represents double (64-bit floating point) numbers. It can hold any kind of number, with or without floating point, within its range. Exponential Notation can also be used as well.
- 3. BigInt:** BigInt came into existence in 2020 to store too large numbers to store in normal number data type.
- 4. Boolean:** It can store two values: true or false. It can be used to check whether a condition is fulfilled or not.
- 5. Undefined:** A variable without a value has undefined value as well as undefined data type. It is basically used to empty a variable.
- 6. Null:** Null variable stores a null value. It is an object of a variable whose value is null. It can be converted to 0 while performing arithmetic operations on it;
- 7. Symbol:** Symbol is a built-in object whose constructor returns a unique symbol. Symbols are often used to add unique property keys to an object that won't collide with keys any other code might add to the object. It also enables a form of weak encapsulation, or a weak form of information hiding.

8. Object: Object is written with { } and as a comma-separated name:value pairs. The object data type can contain: Object, Array and Date. Array is basically a sequential comma-separated list of items belonging to the same data type.

Variable:

Variables are used to store data. It can be declared using const, let, var or nothing. Var is the older whereas const and let are newer. Const is used to declare variables that have constant value. Var retains its value on re-declaration while let and const don't support re-declaring. To identify various variables, identifying names or identifiers are required. Identifiers can contain letters, digits, underscores, and dollar signs and must begin with a letter/ \$/ _ but can't contain reserved words like keywords. They are case sensitive as well. We can declare variables using assignment operator ('='). We can also declare many variables in one statement.

Operators:

Operators in JavaScript are as follows:

1. Arithmetic Operators: Arithmetic operators are used to perform various arithmetic operations.

Operators	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation
%	Modulus
++	Increment
--	Decrement

+ can also be used to concatenate strings.

2. Assignment Operators: Assignment Operators are used to assign values to the variables.

Operators	Examples	Same as
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y

<code>*=</code>	<code>x*=y</code>	<code>x=x*y</code>
<code>/=</code>	<code>x/=y</code>	<code>x=x/y</code>
<code>%=</code>	<code>x%=y</code>	<code>x=x%y</code>
<code>**=</code>	<code>x**=y</code>	<code>x=x**y</code>

`+=` can be used to concatenate strings as well.

3. Comparison Operators: Comparison operators are used to compare two or more variables and return an answer of bool type (true or false).

Operators	Description
<code>==</code>	Equal to
<code>===</code>	Equal value and equal type
<code>!=</code>	Not equal
<code>!==</code>	Not equal value or not equal type
<code>></code>	Greater than
<code><</code>	Less than
<code>>=</code>	Greater than or equal to
<code><=</code>	Less than or equal to
<code>?</code>	Ternary operator

4. Logical Operators: It performs logical operations on variables.

Example:-

`true && true = true`, `true && false = false`, `false && true = false`, `false && false = false`;

`true || true = true`, `true || false = true`, `false || true = true`, `false || false = false`;

`!true = false`, `!false = true`;

Operators	Description
<code>&&</code>	Logical and
<code> </code>	Logical or
<code>!</code>	Logical not

5. Type Operators:

Operators	Description
typeof	Returns the type of a variable
instanceof	Returns true if an object is an instance of an object type

6. Bitwise Operators: It works on 32 bit numbers. First the operands get converted into 32 bit binary numbers then the operations get performed. Finally the result is converted back to number data type and is returned.

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~ 0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Left shift	5 << 1	0101 << 0001	1010	10
>>	Right shift	5 >> 1	0101 >> 0001	0010	2
>>>	Unsigned right shift	5 >>> 1	0101 >>> 0001	0010	2

REFERENCES:

<https://www.w3schools.com/js/>

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Symbol

<https://www.scaler.com/topics/javascript/null-and-undefined-in-javascript/>