Pakistan Blockchain Institute

MODULE-1

JAVASCRIPT CRASH COURSE

Class-3

Edversity.

Raja Rizwan Saleem
Lead Blockchain Trainer

Arithmetic Operators in JavaScript

Arithmetic Operators

- **Mathematical Calculations**: Enable performing basic mathematical operations such as addition, subtraction, multiplication, division, and modulus.
- **Data Manipulation**: Facilitate manipulating numerical data, which is essential for developing dynamic web applications.
- **Algorithm Implementation**: Essential for implementing algorithms that require mathematical operations, such as sorting and searching algorithms.

Arithmetic Operators

- **Financial Calculations**: Important for applications involving financial calculations, such as interest rates, pricing, and discounts.
- **Gaming Logic**: Used in developing games to calculate scores, player positions, and other game mechanics.
- **Animation**: Enable creating smooth animations by calculating positions, velocities, and other properties over time.
- **Form Validation**: Useful in validating numerical input in forms, ensuring that data entered by users meets specific criteria.

Arithmetic Operators

- **Statistical Analysis**: Important for performing statistical operations, such as averages, standard deviations, and other statistical measures.
- **Data Analysis and Visualization**: Facilitate data analysis and visualization by performing calculations on datasets and rendering graphs or charts.
- **Conditional Logic**: Often used in conjunction with conditional statements to perform specific actions based on calculated values.
- **Performance Optimization**: Arithmetic operations can be optimized for performance, making them efficient for high-frequency calculations.

Arithmetic Operators

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Modulus (%)
- Increment (++)
- Decrement (--)

Addition Operator (+)

- Example:
- var a = 10;
- var b = 5;
- var sum = a + b;
- console.log(sum); // Output: 15

Subtraction Operator (-)

- Example:
- var a = 10;
- var b = 5;
- var difference = a - b;
- console.log(difference); // Output: 5

Multiplication Operator (*)

- Example:
- var a = 10;
- var b = 5;
- var product = a * b;
- console.log(product); // Output: 50

Division Operator (/)

- Example:
- var a = 10;
- var b = 5;
- var quotient = a / b;
- console.log(quotient); // Output: 2

Modulus Operator (%)

- Example:
- var a = 10;
- var b = 3;
- var remainder = a % b;
- console.log(remainder); // Output: 1

Increment Operator (++) (detail discussion in later slides)

- Example:
- var a = 5;
- console.log(++a); // Pre-increment, Output: 6
- console.log(a++); // Post-increment, Output: 6
- console.log(a); // Output after post-increment: 7

Decrement Operator (--) (detail discussion in later slides)

- Example:
- var a = 5;
- console.log(--a); // Pre-decrement, Output: 4
- console.log(a--); // Post-decrement, Output: 4
- console.log(a); // Output after post-decrement: 3

Working with Negative Numbers

- Example:
- var a = -10;
- var b = 5;
- console.log(a + b); // Output: -5
- console.log(a - b); // Output: -15
- console.log(a * b); // Output: -50
- console.log(a / b); // Output: -2

Operator Precedence

- Example:
- var a = 10;
- var b = 5;
- var c = 2;
- var result = a + b * c; // Multiplication first, then addition
- console.log(result); // Output: 20

Using Parentheses

- Example:
- var a = 10;
- var b = 5;
- var c = 2;
- var result = (a + b) * c; // Parentheses change the order
- console.log(result); // Output: 30

Floating Point Arithmetic

- Example:
- var a = 0.1;
- var b = 0.2;
- var sum = a + b;
- console.log(sum); // Output: 0.30000000000000004

Summary of Arithmetic Operators

Operator	Description	Example
** Let X = 10 and Y = 5**		
+	Addition	X+Y = 15
-	Subtraction	X-Y = 5
*	Multiplication	X*Y = 50
/	Division	X/Y = 2
%	Modulus	X%Y = 0
++	Increment	X++ = 11
--	Decrement	X-- = 4

Increment / Decrement Operators

Increment and Decrement Operator

1. While working on application you will frequently required to increase variable by 1 or decrease variable by 1
2. For this situation you can use increment and decrement operator
3. ++ increment operator
4. -- decrement operator
5. These operators can be used as prefix and postfix

Increment and Decrement Operator

1. We can increase or decrease value using existing addition and subtraction operators

```
var a = 12;
a = a + 1;              // 13
a = a - 1;              // 12
var b = 12;
b += 1;                 // 13
b -= 1;                 // 13
```

Increment and Decrement Operator

Prefix Increment and Decrement

```
var age =  12;

++age;
alert(age); //Result 13
--age;
alert(age); //Result 12
```

Increment and Decrement Operator

Postfix Increment and Decrement

```
var age = 12;

age++;
alert(age); //Result 13, same as prefix
age--;
alert(age); //Result 12, same as postfix
```

Increment and Decrement Operator

1. What is the difference between Prefix and Postfix
2. You will NOT find any difference if you will not assign result of prefix and postfix to any other variable
3. Prefix operator first increase/decrease the value in variable and then assign result to other variable
4. Postfix operator first assign the value in other value then increase/decrease value in actual variable

Increment and Decrement Operator

Prefix Increment

```
var age =  12;
var newAge = ++age;

alert(age);        //Result 13
alert(newAge);     //Result 13
```

Increment and Decrement Operator (Practice)

Postfix Increment

```
var age =  12;
var newAge = age++;
alert(age);          //Result 13
alert(newAge);       //Result 12 , see the difference
```

Increment and Decrement Operator

Prefix Decrement

```
var age =  12;
var newAge = --age;

alert(age);        //Result 11
alert(newAge);     //Result 11
```

Increment and Decrement Operator

Postfix Decrement

```
var age =  12;
var newAge = age--;
alert(age);          //Result 11
alert(newAge);       //Result 12 , see the difference
```

Increment and Decrement Operator

Example

```
var a = 4;
var b = 2;
var c = a++ - b; // first value of a placed here
which is 4  then  increase 1  in a  so will
```

```
alert(a);          // 5
alert(b);          // 2
alert(c);          // 2
```

Increment and Decrement Operator

Example

```
var a = 4;
var b = 2;
var c = ++a + b; // first value of a increased so
will become 5 then value of a will be placed
here 5
alert(a);          // 5
alert(b);          // 2
alert(c);          // 7
```

Increment and Decrement Operator

Example

```
var a = 4;
var b = 3;
var c = a++ + --b - --a;
alert(a);          // 4
alert(b);          // 2
alert(c);          // 2
```

Increment and Decrement Operator (practice)

Example

```
var a = 4;
var b = 3;
var c = ++a + b++ - a + --b;
alert(a);          // 5
alert(b);          // 3
alert(c);          // 6
```

String Concatenation

Concatenating Text strings

Strings cannot be divided, multiplied, or subtracted, but the + operator can be used on them. It does not add, but it concatenates—it glues two strings together. The following line will produce the string "concatenate":

"con" + "cat" + "e" + "nate"

Concatenating Text strings

1. The + operator can also be used for concatenating strings
2. E.g:

```
var firstName = Shahbaz";

var lastName = "Hussain";

//concatenate firstName, space character and lastName

var fullName = firstName + " " + lastName;

alert(fullName);
```

Concatenating strings and numbers

1. Adding two numbers, will return the sum, but adding a number and a string will return a string:

```
var a   = "6" + 2;   // "62"

var b   = 3 + "6";   // "36"

var c   = "Hello " +    2; // "Hello 2"

var    d   = 2 +           ";            // "2 Hello"
"Hello
```

Concatenating strings with variables

```
let text1 = "sea"; let
text2 = "food"; let
result =
  text1.concat(text2);
consol.log(result);
```

Concatenating strings with variables

```
let text1 = "Hello";
let text2 = "world!";
let resul  = text1.concat(" ", text2);
console.log(result);
```

Concatenating strings with variables

```
let text1 = "Hello";
let text2 = "world!";
let text3 = "Have a nice day!";
let result = text1.concat(" ", text2, " ", text3);
console.log(result);
```

Prompt and Parsing String

prompt or window.prompt()

1. The prompt()/window.prompt() method displays a dialog box that prompts the visitor for input.
2. In a prompt, you can specify a second string. This is the default response that appears in the field when the prompt displays.
3. The prompt() method returns the input value if the user clicks "OK". If the user clicks "cancel" the method returns null.

prompt or window.prompt()

```
var question = "What is your name?";
var defaultAnswer = "Rizwan";

var name = prompt(question, defaultAnswer);
console.log("Name = "+ name);
```

prompt or window.prompt()

1. If you ask for number in prompt and try apply addition operator on it then it will concatenate the value because prompt returns string

```
var question = "What is your age?";
var age = prompt(question);// Assuming input 12
var newAge = age + 5; // It will concatenate
console.log("New Age = "+ newAge); //result 125
```

prompt or window.prompt()

```
let age = prompt("What is your age?");
console.log(age);
```

The confirm() Method

The confirm() method displays a dialog box with a message, an OK button, and a Cancel button. The confirm() method returns true if the user clicked "OK", otherwise false. A confirm box is often used if you want the user to verify or accept something. A confirm box takes the focus away from the current window, and forces the user to read the message. Do not overuse this method. It prevents the user from accessing other parts of the page until the box is closed.

```
let deletePost = confirm("Do you really want to
delete this post?");
console.log(deletePost);
```

Convert string to integer

1. Addition operator in string concatenate values, even if value in string is number

```
var     value1   = "3";
var     value2   = "5";
var     value3   = value1+value2;
console.log(value3);// result, 35
```

Convert string to integer

1. prompt function also returns string even if you provide number in input box

```
//Assuming we will provide value 5 in input

var  age  =  prompt("What   is  your  age");
var  num  =  4;

var  sum  =  age + num;
console.log(sum); // result, 54
```

Increment and Decrement Operator

1. What is the difference between Prefix and Postfix
2. You will NOT find any difference if you will not assign result of prefix and postfix to any other variable
3. Prefix operator first increase/decrease the value in variable and then assign result to other variable
4. Postfix operator first assign the value in other value then increase/decrease value in actual variable

Increment and Decrement Operator

Prefix Increment

```
var age =  12;
var newAge = ++age;

alert(age);        //Result 13
alert(newAge);     //Result 13
```

Increment and Decrement Operator (Practice)

Postfix Increment

```
var age =  12;
var newAge = age++;
alert(age);          //Result 13
alert(newAge);       //Result 12 , see the difference
```

Increment and Decrement Operator

Prefix Decrement

```
var age =  12;
var newAge = --age;

alert(age);        //Result 11
alert(newAge);     //Result 11
```

Increment and Decrement Operator

Postfix Decrement

```
var age =  12;
var newAge = age--;
alert(age);          //Result 11
alert(newAge);       //Result 12 , see the difference
```

Increment and Decrement Operator

Example

```
var a = 4;
var b = 2;
var c = a++ - b; // first value of a placed here
which is 4  then  increase 1  in a  so will
```

```
alert(a);          // 5
alert(b);          // 2
alert(c);          // 2
```

Increment and Decrement Operator

Example 2

```
var a = 4;
var b = 2;
var c = ++a + b; // first value of a increased so
will become 5 then value of a will be placed
here 5
alert(a);          // 5
alert(b);          // 2
alert(c);          // 7
```

Increment and Decrement Operator

Example 3

```
var    a   =   4;
var    b   =   3;
var    c   =   a++     +   --b      - --a;
alert(a);                          // 4
alert(b);                          // 2
alert(c);                          // 2
```

Increment and Decrement Operator (practice)

Example 4

```
var  a  =  4;
var  b  =  3;
var  c  =  ++a  +  b++  - a  +  --b;
alert(a);                  // 5
alert(b);                  // 3
alert(c);                  // 6
```

Increment and Decrement Operator (practice)

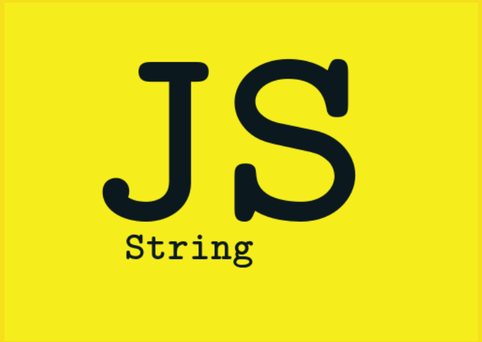
Example 4

This expression involves both pre-increment (`++a`) and post-increment (`b++`) operators.

- `++a` (pre-increment): This operation increases the value of `a` by 1 before using it in the expression. So, `++a` changes `a` from 5 to 6, and the value used in the expression is 6.
- `b++` (post-increment): This operation uses the current value of `b` in the expression and then increases `b` by 1. So, `b++` uses the value 3 in the expression, and then `b` becomes 4.

Combining these, the expression `++a + b++` is evaluated as:

```
c = 6 + 3;  // `++a` is 6, `b++` is 3
```



Working with JavaScript Strings

Introduction to Strings

- Strings are used to store and manipulate text.
- In JavaScript, strings are sequences of characters enclosed in single or double quotes.

Creating Strings

- Example:
- var str1 = "Hello, World!";
- var str2 = 'Hello, JavaScript!';

String Length

- Example:
- var str = "Hello, World!";
- var length = str.length; // length is 13
- console.log(length);

Accessing Characters

- Example:
- var str = "Hello";
- var firstChar = str[0]; // firstChar is 'H'
- console.log(firstChar);

Changing Case

- Example:
- var str = "Hello, World!";
- var upperStr = str.toUpperCase(); // "HELLO, WORLD!"
- var lowerStr = str.toLowerCase(); // "hello, world!"
- console.log(upperStr);
- console.log(lowerStr);

String Concatenation

- Example:
- var str1 = "Hello";
- var str2 = "World";
- var greeting = str1 + ", " + str2 + "!";
- console.log(greeting); // "Hello, World!"

Finding a Substring

- Example:
- var str = "Hello, World!";
- var position = str.indexOf("World"); // position is 7
- console.log(position);

Extracting Substrings

- Example:
- var str = "Hello, World!";
- var subStr = str.slice(0, 5); // "Hello"
- console.log(subStr);

Replacing Substrings

- Example:
- var str = "Hello, World!";
- var newStr = str.replace("World", "JavaScript");
- console.log(newStr); // "Hello, JavaScript!"

Splitting Strings

- Example:
- var str = "Hello, World!";
- var arr = str.split(", ");
- console.log(arr); // ["Hello", "World!"]

Template Literals

- Example:
- var name = "JavaScript";
- var greeting = `Hello, ${name}!`;
- console.log(greeting); // "Hello, JavaScript!"

String Trim

- Example:
- var str = " Hello, World! ";
- var trimmedStr = str.trim();
- console.log(trimmedStr); // "Hello, World!"

String Repeat

- Example:
- var str = "Hello";
- var repeatedStr = str.repeat(3);
- console.log(repeatedStr); // "HelloHelloHello"

Checking for Substrings

- Example:
- var str = "Hello, World!";
- var containsHello = str.includes("Hello"); // true
- console.log(containsHello);

String Starts and Ends

- Example:
- var str = "Hello, World!";
- var startsWithHello = str.startsWith("Hello"); // true
- var endsWithWorld = str.endsWith("World!"); // true
- console.log(startsWithHello);
- console.log(endsWithWorld);

Comparing Strings

- Example:
- var str1 = "Hello";
- var str2 = "hello";
- var areEqual = str1 === str2; // false
- console.log(areEqual);

String Concatenation using concat()

- Example:
- var str1 = "Hello";
- var str2 = "World";
- var greeting = str1.concat(", ", str2, "!");
- console.log(greeting); // "Hello, World!"

Extracting Characters using charAt()

- Example:
- var str = "Hello";
- var char = str.charAt(0); // 'H'
- console.log(char);

Extracting Characters using charCodeAt()

- Example:
- var str = "Hello";
- var charCode = str.charCodeAt(0); // 72
- console.log(charCode);

Thanks
End of Module-2 (Class-3)