[9] Expressions and Operators

An expression is any valid unit of code that produces a value. It can be as simple as a single constant value or as complex as a combination of variables, operators, and function calls.

Arithmetic Expressions

These involve arithmetic operations like addition, subtraction, multiplication, and division. For example:

```
const result = 5 + 3 * (2 / 4); // This is an arithmetic expression
```

String Concatenation

Combining strings is an example of an expression:

```
const greeting = "Hello, " + "world!"; // This is a string concatenation expression
```

Logical Expressions

These involve logical operations such as AND (&&), OR (||), and NOT (!). For example:

```
const isTrue = true
const isFalse = false
const result = isTrue && !isFalse; // This is a logical expression
```

Comparison Expressions

These involve comparing values using comparison operators like equal (==), not equal (!=), greater than (>), less than (<), etc. For example:

```
const x = 5;
const y = 10;
const isEqual = x === y; // This is a comparison expression
```

Function Calls

```
function add(a, b) {
return a + b;
}
const sum = add(3, 4); // This is a function call expression
```

Array and Object Access

Accessing elements in an array or properties of an object is an expression:

```
const numbers = [1, 2, 3];
const firstNumber = numbers[0]; // This is an array access expression
const person = { name: "John", age: 30 };
const personName = person.name; // This is an object property access expression
```

Operators:

Operators are symbols or keywords used to perform operations on operands. Operands are values or expressions that an operator acts upon. JavaScript provides a wide range of operators, including:

1. Arithmetic Operators: Used for mathematical calculations, like +, -, *, /, % (remainder).
2. Assignment Operators: Used to assign values to variables, such as =, +=, -=, *=, /=, %=.
3. Comparison Operators: Used to compare values, like ==, !=, ===, !==, >, <, >=, <=.
4. Logical Operators: Used for logical operations, such as && (AND), || (OR), ! (NOT).
5. Conditional (Ternary) Operator: Allows you to assign a value based on a condition, like condition ? valueIfTrue : valueIfFalse.
6. Bitwise Operators: Used for bit-level operations on integers, such as &, |, ^, ~, <<, >>, >>>.
7. Unary Operators: Operate on a single operand, like ++ (increment), -- (decrement), - (negation), + (unary plus).
8. Type Operators: Used to check the type of a value, such as typeof, instanceof.
9. String Operators: Used for string concatenation, like +.
10. Other Operators: JavaScript also includes other operators for various purposes, such as the comma operator (,), the in operator for checking if an object has a specific property, and the delete operator for removing object properties.