**JAVA SCRIPT FUNDAMENTALS**

1. Variables.
2. Datatypes.
3. Operators.
4. Control Statements.
5. Loops.
6. Functions.
7. Arrays.
8. **Variables.**

A JavaScript variable is simply a name of storage location. There are two types of variables in JavaScript: local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore (\_), or dollar ($ ) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

**Correct JavaScript variables**

var x = 10;

var \_value="gowtham";

**Example of JavaScript variable**

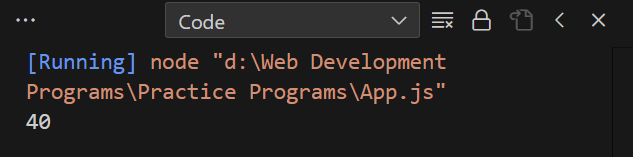
var x = 20;

var y = 20;

var z=x + y;

console.log(z);

Output of the above example:



**JavaScript local variable Program**

function scope(){

var x=10;//local variable

}

**JavaScript global variable Program**

var data=200;//global variable

function scope(){

console.log(data);

}

function scope2(){

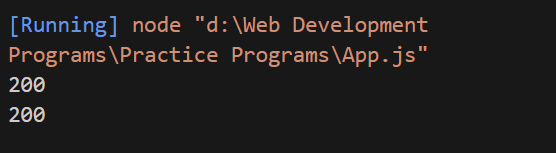
console.log(data);

}

scope(); //calling JavaScript function

scope2 ();

**Output**:

****

1. **Datatypes.**

JavaScript provides different data types to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

**JavaScript primitive data types:**

There are five types of primitive data types in JavaScript.

|  |  |
| --- | --- |
| Data Type | Description |
| String | represents sequence of characters e.g., "hello" |
| Number | represents numeric values e.g., 100 |
| Boolean | represents Boolean value either false or true |
| Undefined | represents undefined value |
| Null | represents null i.e., no value at all |

**Example:**

// Data Types in JavaScript

var string = "gowtham"; // It holds the string datatype

var num = 104; // It holds the number datatype

var bool = true; // It holds the boolean datatype

var notdefined = undefined; // It holds the undefined datatypes that is not defined the value

var value = null; // It holds the nothing null

console.log(string);

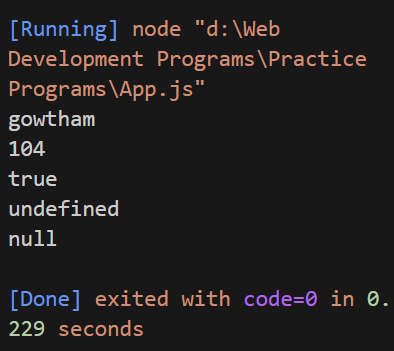
console.log(num);

console.log(bool);

console.log(notdefined);

console.log(value);

**OUTPUT:**

****

**JavaScript non-primitive data types:**

|  |  |
| --- | --- |
| Data Type | Description |
| Object | represents instance through which we can access members. |
| Array | represents group of similar values |
| Reg Exp | represents regular expression |

**Example:**

// Object Datatype

var emp = new Object();

emp.name = "gowtham";

emp.id = 104;

emp.location = "chennai";

console.log(emp.id);

console.log(emp.name);

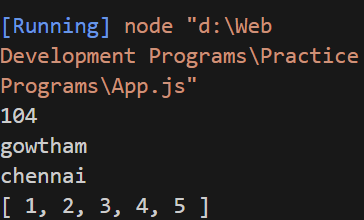
console.log(emp.location);

// Array Datatype

var arr = new Array(1,2,3,4,5);

console.log(arr);

**OUTPUT:**



1. **Operators.**

There are following types of operators in JavaScript.

1. Arithmetic Operators.
2. Comparison (Relational) Operators.
3. Bitwise Operators.
4. Logical Operators.
5. Assignment Operators.
6. Special Operators.
7. **Arithmetic Operators:**
8. **+ (Addition):**

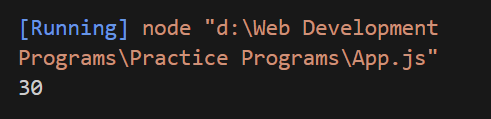
var x = 10;

var y = 20;

var z=x+y;

console.log(z);

**OUTPUT:**

****

1. **- (Subtraction):**

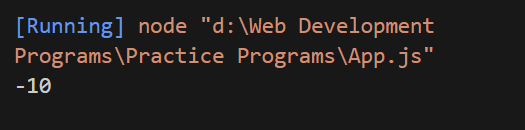
var x = 10;

var y = 20;

var z=x-y;

console.log(z);

**OUTPUT:**

****

1. **\* (Multiplication):**

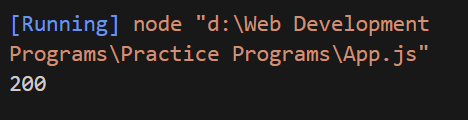
var x = 10;

var y = 20;

var z=x\*y;

console.log(z);

**OUTPUT:**

****

1. **/ (Division):**

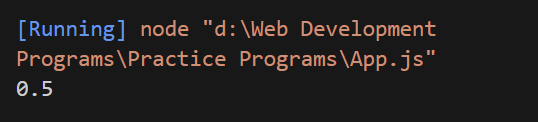
var x = 10;

var y = 20;

var z=x/y;

console.log(z);

**OUTPUT:**

****

1. **% (Modulus):**

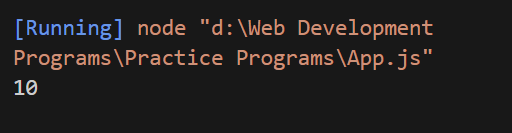
var x = 10;

var y = 20;

var z=x%y;

console.log(z);

**OUTPUT:**

****

1. **++ (Increment):**

var x = 10;

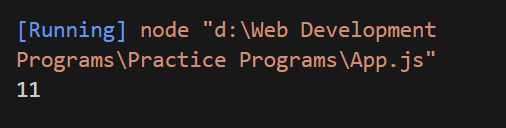
var y = 20;

var z=++x;

var a=++y;

console.log(z);

**OUTPUT:**

****

1. **-- (Decrement):**

var x = 10;

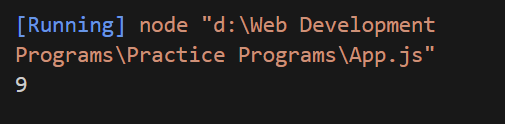
var y = 20;

var z=--x;

var a=--y;

console.log(z);

**OUTPUT:**

****

1. **Comparison (Relational) Operators:**

|  |  |
| --- | --- |
| Operator | Description |
| = = | Is equal to |
| = = = | Identical (equal and of same type) |
| ! = | Not equal to |
| ! = = | Not identical |
| > | Greater than |
| > = | Greater than or equal to |
| < | Less than |
| < = | Less than or equal to |

**Example:**

// Relational Operators - (==,===,!=,<=,>=,<,>)

var a = 100;

var b = 200;

var c = 100;

if(a == c){

console.log("a is equal to c");

}

else{

console.log("a is not equal to c");

}

if(a<b){

console.log("a is less than b");

}

else{

console.log("a is greater than b");

}

if(b>a){

console.log("b is greater than a");

}

else{

console.log("b is lessthan a");

}

if(a === b){

console.log("a is equal to b and assign the value to tha a by b");

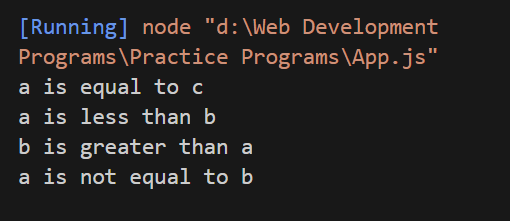
}

if(a!=b){

console.log("a is not equal to b");

}

**OUTPUT:**



1. **Bitwise Operator:**

|  |  |
| --- | --- |
| Operator | Description |
| & | Bitwise AND |
| | | Bitwise OR |
| ^ | Bitwise XOR |
| ~ | Bitwise NOT |
| << | Bitwise Left Shift |
| >> | Bitwise Right Shift |
| >>> | Bitwise Right Shift with Zero |

**Examples:**

// AND Operator (&)

let x = 5 & 7;

console.log(x);

// OR Operator

let y = 5 | 10;

console.log(y);

// XOR Operator

let z = 3 ^ 9;

console.log(z);

// Left Shift

let a = 5 << 7;

console.log(a);

// Right Shift

let b = 8 >> 6;

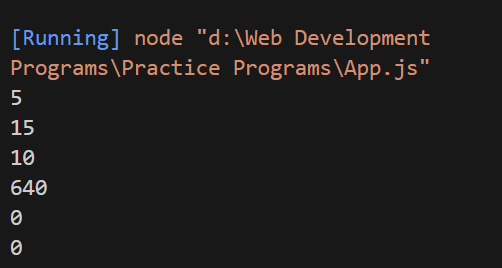
console.log(b);

// Zerofill Right Shift

let c = 6 >>> 7;

console.log(c);

**OUTPUT:**

****

1. **Logical Operators:**

|  |  |
| --- | --- |
| Operator | Description |
| && | Logical AND |
| || | Logical OR |
| ! | Logical Not |

**Example:**

// Logical AND (&&)

let name = "gowtham";

let myname = "gowtham";

let a = 10;

let b = 12;

let c = 12;

if(a<b && c == b){

console.log("a is lessthan b and c is same as b");

}

if(b==c || a>b){

console.log("Second if Statement is true");

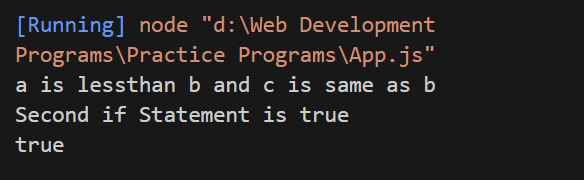
}

if(1){

console.log(!(a==b));

}

**OUTPUT:**

****

1. **Assignment Operators:**

|  |  |
| --- | --- |
| Operator | Description |
| = | Assign |
| + = | Add and assign |
| - = | Subtract and assign |
| \* = | Multiply and assign |
| / = | Divide and assign |
| % = | Modulus and assign |

**Example:**

let a = 10;

a = 20;

let b = 18;

console.log(a);

console.log(a+=b);

console.log(a-=b);

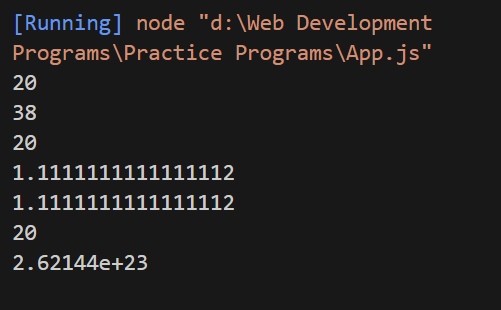
console.log(a/=b);

console.log(a%=b);

console.log(a\*=b);

console.log(a\*\*=b);

**OUTPUT:**

****

1. **Ternary Operator:**

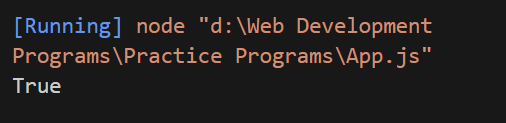
**Example:**

let b = 10;

let c = 20;

(b<c) ? console.log("True") : console.log("False");

**OUTPUT:**

****

1. **Control Statements:**

The **JavaScript if-else statement** is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.

1. **If Statement:**

**Example:**

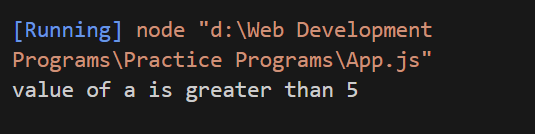
var a=10;

if(a>5){

console.log("value of a is greater than 5");

}

**OUTPUT:**

****

1. **If Else Statement:**

**Example:**

var a=20;

if(a%2==0){

console.log("a is even number");

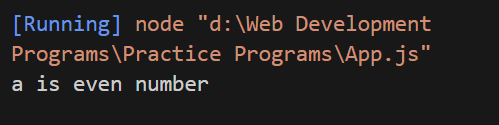
}

else{

console.log("a is odd number");

}

**OUTPUT:**

****

1. **IfElseIf Statements:**

**Example:**

var a=20;

if(a==10){

console.log("a is equal to 10");

}

else if(a==15){

console.log("a is equal to 15");

}

else if(a==20){

console.log("a is equal to 20");

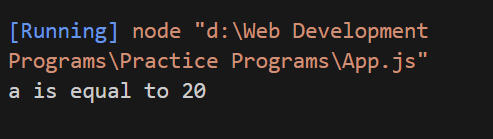
}

else{

console.log("a is not equal to 10, 15 or 20");

}

**OUTPUT:**

****

1. **Swich Statement Example:**

var grade='A';

var result;

switch(grade){

case 'A':

result="A Grade";

break;

case 'B':

result="B Grade";

break;

case 'C':

result="C Grade";

break;

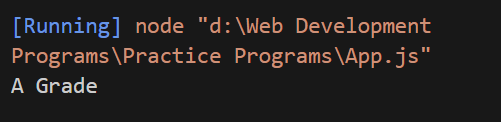
default:

result="No Grade";

}

console.log(result);

**OUTPUT:**

****

1. **Loops:**

The **JavaScript loops** are used to iterate the piece of code*using* for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. **For loop:**

// For Loop in Js

for(let i = 0;i<10;i++){

console.log(i);

}

for(let a = 10;a<=20;a++){

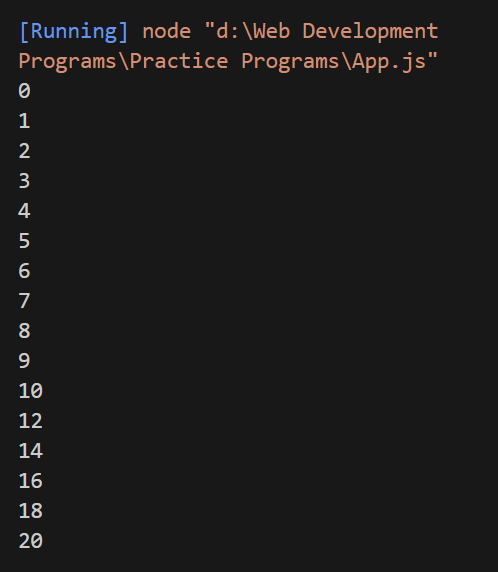
if(a%2 == 0){

console.log(a);

}

}

**OUTPUT:**

****

1. **While Loop:**

var i=11;

while (i<=15)

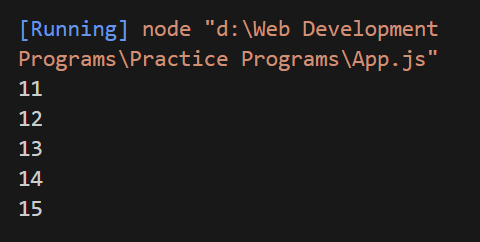
{

console.log(i);

i++;

}

**OUTPUT:**

****

1. **Do While Example:**

var i=21;

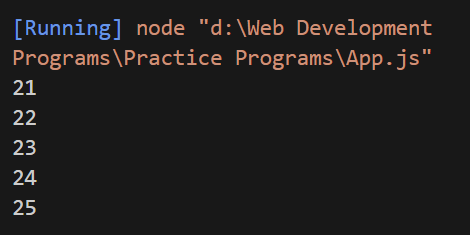
do{

console.log(i);

i++;

}while (i<=25);

**OUTPUT:**

****

1. **For Of Loop:**

const con = ["India", "pakistan", "china"];

let text = "";

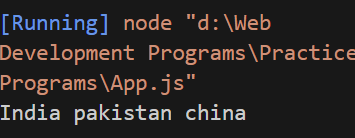
for (let x of con) {

text += x + " ";

}

console.log(text);

**OUTPUT:**

****

1. **For In Loop:**

const user = {fname:"gowtham", lname:"Suresh", age:20};

let text = "";

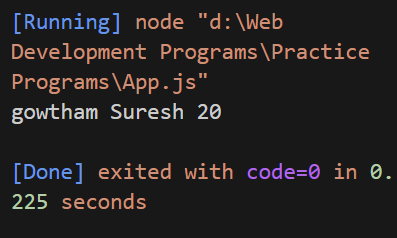
for (let x in user) {

text += user[x] + " ";

}

console.log(text);

**OUTPUT:**

****

1. **Arrays**

**JavaScript array** is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

1. **By array literal:**

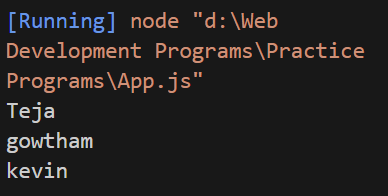
var emp=["Teja","gowtham",”kevin”];

for (i=0;i<emp.length;i++){

console.log(emp[i]);

}

**OUTPUT:**

****

1. **By Array Using Constructor:**

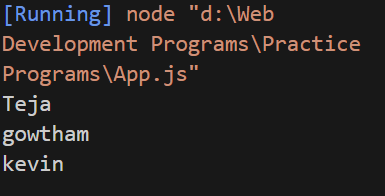
var emp = new Array("Teja","gowtham","kevin");

for (i=0;i<emp.length;i++){

console.log(emp[i]);

}

**OUTPUT:**

****

1. **By Array Using New Keyword:**

var i;

var emp = new Array();

emp[0] = "gowtham";

emp[1] = "kevin";

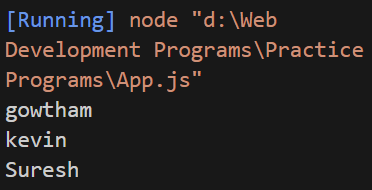
emp[2] = "Suresh";

for (i=0;i<emp.length;i++){

console.log(emp[i]);

}

**OUTPUT:**

****

1. **Functions:**

**Function Example:**

function sum(a,b)

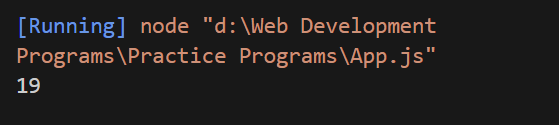
{

console.log(a + b);

}

var result = sum(9,10);

**OUTPUT:**

****

**Function With Return Type:**

function add(){

let a = 10;

let b = 20;

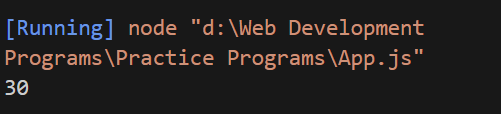
return a + b;

}

var a = add();

console.log(a);

**OUTPUT:**

****