**Java Script Language Fundamentals**

**Keywords or reserved words**

Java script provides keywords which are used to perform specific operation. Keywords are language related words and the meaning of these words are reserved by java script translator. These words cannot be used as user defined words or identifiers

The list of JavaScript keywords outlined below is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| goto | In | instanceof | Static | finally |
| arguments | public | Do | else | Const |
| Function | class\* | return | let\* | Catch |
| Eval | For | if | This | try |
| Break | debugger | Yield | extends\* | enum\* |
| Continue | export\* | null | switch | private |
| New | throw | while | Case | await\* |
| Delete | super\* | default | Void | Var |
| protected | Package | Interface | False | typeof |
| implements | With | import\* | True | ` |

The keywords which are marked by ‘\*’ are new in ECMAScript 5 and 6 (which are the version of javascript). The lists of reserved words which have been removed from the ECMAScript 5 and 6 standard are given below –

goto

double

transient

volatile

int

synchronized

throws

native

float

short

byte

Boolean

long

abstract

final

char.

**Identifiers**

Identifiers are user defined words. These words are used to identify programming elements

1. Variable name
2. Function name
3. Class name/type name
4. Identifier should not be a reserved word
5. Identifier should not start with digit
6. There should not be any space between identifier
7. Identifier allows one special character \_

If the identifier is having multiple words it is separated with capital letter

rollNumber=1 (Camel Case)

rollno=1 studentName=”naresh”

**Working with java script?**

Browser provides java script console which is used to learn java script, it will not provide environment for developing applications.

1. Open Browser
2. Press F12 Function key 🡪 to open java script console

In development mode, the java script code is included within HTML

in two ways

1. Inline
2. File

|  |  |
| --- | --- |
| Inline  <HTML>  <HEAD>  <SCRIPT LANGAGE=”JAVASCRIPT”>  </SCRIPT>  </HEAD>  <BODY>  </BODY>  </HTML> | File  <HTML>  <HEAD>  <SCRIPT SRC=”JS1.JS”>  <SCRIPT SRC=”JS2.JS”>  </HEAD>  <BODY>  </BODY>  </HTML> |

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>Java Script Test</title>

<script>

    function display()

    {

    document.write("<H1> This is Header displayed using Java Script</H1>");

    }

</script>

</head>

<body style="background-color: aquamarine;">

<h2>Example of Java Script</h2>

<form>

    <input type="button" onclick="display()">

</form>

</body>

</html>

**Java Script Data types**

Java Script data types are classified into different categories

1. Numbers/Numeric Data types
2. String
3. Collection
4. UDT (User Defined Data types)

**Numbers**

These numbers can be,

1. Integers
2. Floats

typeof() : typeof() is a predefined function which returns the type of object hold by variable (OR) variable type

“number” is data type which is used to represent numbers

1. Integers 2.float

**Example:**

a=10

10

b=1.5

1.5

typeof(a)

'number'

typeof(b)

'number'

a+b

11.5

a-b

8.5

a\*b

15

**String**

This string is represented within double quotes or single quotes

s1='python'

'python'

typeof(s1)

'string'

s2="python"

'python'

typeof(s2)

'string'

s2

'python'

**Creating Variables**

Variable is container where the values are stored

The content of variable can be modified or the value variable can be changed

“var” is keyword which is used for crating variable

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>Java Script Test</title>

<script>

    var a=10

    var b=20

    var c=a+b

    document.write("<h2>Sum is ")

    document.write(c)

    document.write("</h2>")

</script>

</head>

<body style="background-color: aquamarine;">

<h2>Example of Java Script</h2>

</body>

</html>

**const**

const is keyword used for creating constants

the value of constants are never changed

const <constant-name>=<value>

const PI=3.147

**var:**The scope of a variable defined with the keyword “var” is limited to the “function” within which it is defined. If it is defined outside any function, the scope of the variable is global.  
**var is “function scoped”.**

**let:**The scope of a variable defined with the keyword “let” or “const” is limited to the “block” defined by curly braces i.e. {} .  
**“let” and “const” are“block scoped”.**

**const:** The scope of a variable defined with the keyword “const” is limited to the block defined by curly braces. However if a variable is defined with keyword const, it cannot be reassigned.  
**“const” cannot be re-assigned to a new value. However it CAN be mutated.**

Java script allows displaying three types of dialog boxes or message boxes

1. alert
2. prompt
3. confirm

**What are alerts in JavaScript?**

The alert() method/function in JavaScript is used to display a virtual alert box. It is mostly used to give a warning message to the users. It displays an alert dialog box that consists of some specified message (which is optional) and an OK button.

**Example:**

<html>

    <head>

    <script language="javascript">

        alert("Submit Data")

    </script>

    </head>

</html>

JavaScript prompt() dialog box. The prompt() method/function in JavaScript is **used to display a prompt box that prompts the user for the input**. It is generally used to take the input from the user before entering the page.

Prompt function returns string value

**Example:**

<html>

    <head>

    <script language="javascript">

        n1=prompt("Enter the value of n1",0)

        n2=prompt("Enter the value of n2",0)

        n3=n1+n2

        alert(n3)

    </script>

    </head>

</html>

**Example:**

<html>

    <head>

    <script language="javascript">

        n1=prompt("Enter the value of n1",0)

        n2=prompt("Enter the value of n2",0)

        n3=Number(n1)+Number(n2)

        alert(n3)

    </script>

    </head>

</html>

The JavaScript confirm() method displays a specified message in a dialog box, containing OK and CANCEL buttons. **A confirm box is used to accept or verify something**. The confirm JavaScript box forces the browser to read the message

<html>

    <head>

    <script language="javascript">

        s=confirm("Submit Data?")

        if(s==true)

        {

            document.write("<h1> OK Button is clicked");

        }

        else

        {

            document.write("<h1> Cancel button is clicked");

        }

    </script>

    </head>

</html>

**What is scripting language?**

Scripting language is one type of programming language.

Scripting languages are used to write scripts.

**What is script?**

A script is small program executed inside another program.

Script cannot executed individually.

Example: java script,vb script, type script, php, perl,…

A programming language can have the features of scripting language.

A scripting language cannot have the features of programming language.

**Q: What is difference between programming language and scripting language?**

|  |  |
| --- | --- |
| **Programming Language** | **Scripting Language** |
| Programs written in programming languages are executed individually or independently | Scripts cannot be executed individually or independently. It required another program to run |
| Every program is executed in separate process. | Script is executed inside existing process. |
| Heavy weight | Light weight |
| Uses compiler as translator | Uses Interpreter as translator |
| Statically typed languages | Dynamically typed languages |

**document**

When an HTML document is loaded into a web browser, it becomes a **document object**. The **document object** is the root node of the HTML document.

**JavaScript Document object is an object** that provides access to all HTML elements of a document. When an HTML document is loaded into a browser window, then it becomes a document object.

Document object provides two write methods

1. write
2. writeln

write and writeln are the same function. The only difference is that **writeln adds a new line at the end of the text**.

<html>

    <head>

    <script language="javascript">

        document.write("This is first line");

        document.write("This is second line");

        document.writeln();

        document.writeln("This is first line");

        document.writeln("This is second line");

    </script>

    </head>

</html>

**Java Script operators**

The operators are used to perform some operations

Based on the operands on which it performs operations operators are classified into different categories

1. unary operators
2. binary operators
3. ternary operations

**Arithmetic operators**

These operators are used to perform arithmetic operations

Operator Description

+ Addition

- Subtraction

\* Multiplication

\*\* Exponentiation

/ Division

% Modulus (Remainder)

++ Increment (unary operator)

-- Decrement (unary operator)

<html>

    <head>

        <script>

            n1=Number(prompt("Enter First Number")) // Global variable

            n2=Number(prompt("Enter Second Number")) // Global variable

            function add()

            {

                alert("Sum is "+(n1+n2))

            }

            function sub()

            {

                alert("Difference is "+(n1-n2))

            }

            function multiply()

            {

                alert("Product is "+(n1\*n2))

            }

            function div()

            {

                alert("Result is "+(n1/n2))

            }

        </script>

    </head>

    <body>

        <form>

            <input type="button" value="Add" onclick="add()">

            <input type="button" value="Sub" onclick="sub()">

            <input type="button" value="Multiply" onclick="multiply()">

            <input type="button" value="division" onclick="div()">

        </form>

    </body>

</html>

+ operator is used to perform two operations

1. Adding numbers
2. Concatenation of strings

If two operands are numbers it performs addition

If anyone operand is string it performs concatenation

**Example:**

10+20

30

10+"20"

'1020'

1.5+2.5

4

"1.5"+2

'1.52'

\*\* : it is called exponent operator or power of operator

4\*\*2

16

% : Modular operator, this operator divide two numbers and get remainder

a=5

5

b=2

2

a%b

1

**Relational Operators**

These operators are used to compare data or values. Relational operators return Boolean values

|  |  |
| --- | --- |
| Operators | Description |
| > | Greater than operator |
| < | Less than operator |
| >= | Greater than or equal operator |
| <= | Less than or equal to operator |
| == | Equal operator (value comparing) |
| != | Not equal operator (value comparing) |
| === | Equality of value and type |
| !== | Not Equality of value and type |

Example:

a>b

true

if(a>b)

alert(a+"is max")

undefined

if(a<b)

alert(a+"is min")

undefined

if(a<b)

alert(a+"is min")

undefined

a>b

true

a<b

false

b>a

false

Example:

<html>

<head>

    <script>

        a=Number(prompt("Enter the value of a"));

        b=Number(prompt("Enter the value of b"));

        function maximum()

        {

            if(a>b)

            alert(a+"is Max")

            else

            alert(b+"is Max")

        }

        function minimum()

        {

            if(a<b)

            alert(a+"is minimum")

            else

            alert(b+"is minimum")

        }

    </script>

</head>

<body>

    <form>

        <input type="button" value="max" onclick="maximum()">

        <input type="button" value="min" onclick="minimum()">

    </form>

</body>

</html>

**Example:**

<html>

<head>

<script>

    function login()

    {

        user=prompt("UserName :") // local variables

        pwd=prompt("Password :") // local variables

        if(user=="nit" && pwd=="nit123")

        alert(user+"Welcome")

        else

        alert("invalid username or password")

    }

</script>

</head>

    <body>

    <form>

        <input type="button" value="login" onclick="login()">

    </form>

    </body>

</html>

**Example:**

a=10

10

b="10"

'10'

a==b

true

typeof(a)

'number'

typeof(b)

'string'

a===b

false

a===b

false

c="10"

'10'

b==c

true

d=10

10

a===d

true

b===c

true

**Logical operators**

This operators are used to combine two conditions or Boolean expressions

|  |  |
| --- | --- |
| Operator | Description |
| && | And operator |
| || | Or |
| ! | Not operator |

**Assignment operators or update operators**

This operator performs two operations

1. Binary operation
2. Assignment operation

|  |  |  |
| --- | --- | --- |
| +=  -=  \*=  /=  %=  \*\*= | a=10  10  a+=5  15  x=10  10  x-=5  5 | a=5  5  a\*=2  10  a  10  b=4  4  b\*\*=2  16  b  16  x=5  5  x%=2  1 |

**Control statements**

1. Conditional statements
   1. If statement
   2. Switch statement
2. Loop control statements
   1. While loop
   2. For loop

**If statement**

If is a conditional statement

This is used to execute block of statements based on condition

The following keywords are used to work with if

1. If
2. else
3. else if (checking multiple conditions)

Note: only two keywords 1. If 2. Else

**Simple if syntax: if without else**

If(condition)

{

Statement-1;

Statement-2;

}

**Syntax If with else**

If(condition)

{

Statement-1;

Statement-2;

} else {

Statement-3;

Statement-4;

}

**If statement**

If is a conditional statement

This is used to execute block of statements based on condition

The following keywords are used to work with if

1. If
2. else
3. else if (checking multiple conditions)

Note: only two keywords 1. If 2. Else

**Simple if syntax: if without else**

If(condition)

{

Statement-1;

Statement-2;

}

**Syntax If with else**

If(condition)

{

Statement-1;

Statement-2;

} else {

Statement-3;

Statement-4;

}

**Syntax if..else if ..else (if else ladder)**

This syntax is used for checking multiple conditions.

if(condition1)

{

Statement-1;

}

else if (condition2)

{

Statement-2;

}

else if (condtion3)

{

Statement-3

}

else

{

Statement-4;

}

Example:

<html>

<head>

<script>

n1=Number(prompt("Input First Number"))

n2=Number(prompt("Input second Number"))

n3=Number(prompt("Input thrid number"))

if(n1>n2 && n1>n3)

{

    alert(n1+" is Max")

}

else if(n2>n3)

{

    alert(n2+" is Max")

}

else

{

    alert(n3+" is Max")

}

</script>

</head>

</html>

**switch statement**

switch is a branching statement, which is used to move the execution control from one case to another case based on input value. Switch is used to verity equality

**Syntax:**

switch(variable/expression)

{

case <value>:

statement-1

break

case <value>:

statement-2

break

case <value>:

statement-3

break

default:

statement-4

}

Break is used to terminate switch statement

If case not terminated with break statement, control is moved to next case without checking condition.

Example:

<html>

<head>

    <script>

        course=prompt("Enter Course Name")

        switch(course)

        {

            case "java":

                alert("Java Fee 2000")

                break

            case "python":

                alert("Python Fee 4000")

                break

            default:

                alert("Course is Not Avl")

        }

    </script>

</head>

</html>

**Looping control statements**

Javascript support 3 types of looping control statements

1. while loop 🡪 entry controlled loop
2. for loop 🡪 entry controlled loop
3. do..while loop 🡪 exit control loop

looping control statements are used to repeat one or more than one statement number of times or until given condition

**for loop**

for loop is used in different ways in javascript

* for - loops through a block of code a number of times
* for/in - loops through the properties of an object
* for/of - loops through the values of an iterable object

**for - loops through a block of code a number of times**

This syntax allows executing a block of statements number of times

It is similar to the syntax of for loop in “C” language

for(variable;condition;update)

{

Statement-1

Statement-2

}

For loop repeat statmenet-1 and statement-2 until given condition is True, when condition becomes false, it will stop executing

for(let x=1;x<=10;x++)

{

document.write(x)

}

let,var,const

|  |  |
| --- | --- |
| let | var |
| the variable created using let, same variable cannot redeclared | The variable created using var, same variable can be redeclared |

**Example:**

<html>

<head>

    <script>

        n1=Number(prompt("Enter Any number"))

        for(let i=1;i<=10;i++)

        {

            document.write("<h2>"+n1+"x"+i+"="+(n1\*i)+"</h2>")

        }

    </script>

</head>

</html>

**for/in - loops through the properties of an object**

The JavaScript for in statement loops through the properties of an Object:

for (key in object) {  
  // *code block to be executed*  
}

**Note: it is similar to for in of python**

<html>

<head>

    <script>

        student={'rollno':101,'name':'naresh','course':'python'}

        alert(typeof(student))

        for(let stud in student)

        {

            document.write("<h2>"+stud+"</h2>")

            document.write("<h2>"+student[stud]+"</h2>")

        }

    </script>

</head>

</html>

for/of - loops through the values of an iterable object

The forEach() method calls a function (a callback function) once for each array element.

<html>

<head>

    <script>

        arr1=Array(10)

        arr1[0]=100

        arr1[1]=200

        arr1.forEach(x=> {

            document.write(x)

        });

    </script>

</head>

</html>

<html>

<head>

    <script>

        arr1=Array(10)

        arr1[0]=100

        arr1[1]=200

        arr1.forEach(print);

        function print(x)

        {

            document.write(x)

        }

    </script>

</head>

</html>

**while loop**

while is an entry controlled loop

while loop is used to repeat one or more than one statement until given condition is True.

**Syntax:**

while(condition)

{

Statement-1

Statement-2

}

**do..while**

do..while is called exit controlled loop

this loop is executed at least one time

do

{

Statement-1

Statement-2

}while(condition)