JavaScript ES6

Lesson 2: JavaScript Language

Lesson Objectives

Data Types and Variables
JavaScript Operators
Control Structures and Loops
JavaScript Functions

2.1: Data Types and Variables

Data Types in JavaScript

JavaScript is a free-form language. You do not have to declare all variables, classes, and methods

Variables in JavaScript can be of type:

- Number (4.156, 39)
- String (“This is JavaScript”)
- Boolean (true or false)
- Null (null)

Data Types in JavaScript (Contd..)

JavaScript variables are said to be loosely typed

Defining variables: var variableName = value

JavaScript variables

Can include letters of the alphabet, digits 0-9 and the underscore (_) character and is case-sensitive.

Cannot include spaces or any other punctuation characters.

First character of the variable name must be either a letter or the underscore character.

No official limit on the length of a variable n

2.2: JavaScript Operators

Arithmetic Operator

Operator	Description	Example	Result
+	Addition	2 + 2	4
-	Subtraction	5 – 2	3
*	Multiplication	4 * 5	20
/	Division	5 / 2	2.5
%	Modulus	10 % 8	2
++	Increment	x = 5; x++	x = 6
--	Decrement	x = 5; x--	x = 4

Comparison Operator

Operator	Description	Example	Result
==	is equal to	5 == 8	false
!=	is not equal	5 != 8	true
>	is greater than	5 > 8	false
<	is less than	5 <= 8	true
>=	is greater or equal	5 >= 8	false
<=	is less or equal	5 <= 8	true

2.2: JavaScript Operators

Assignment Operator

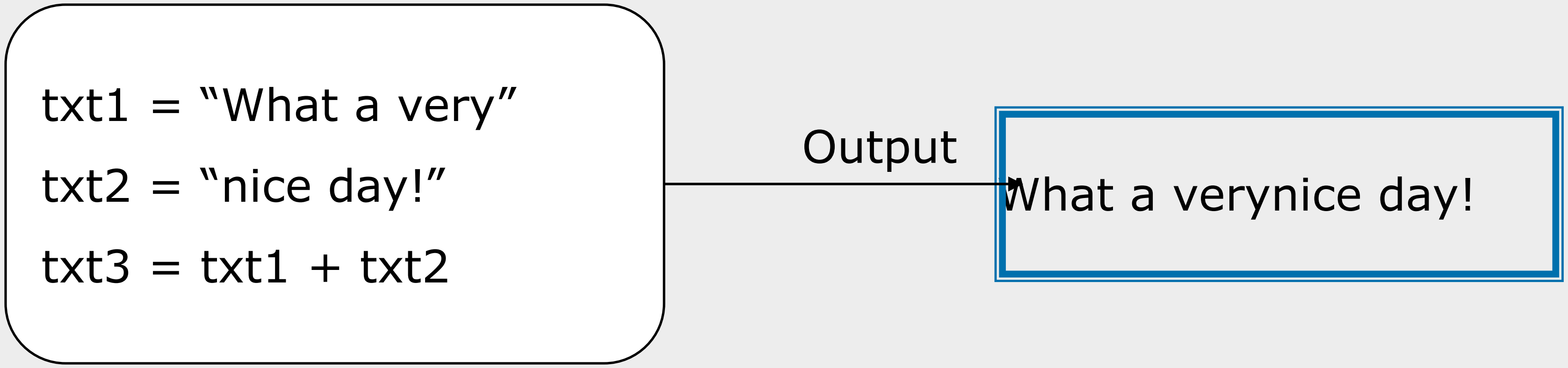
Operator	**Example**	**Is same as**
+=	x += y	x = x + y
-=	x -= y	x = x – y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

2.2: JavaScript Operators

Logical Operator

Operator	**Description**	**Example**
&&	and	x = 6; y = 3 x < 10 && y > 1 returns true
||	or	x = 6; y = 3 x < 10 || y > 5 returns true
!	not	x = false !x returns true

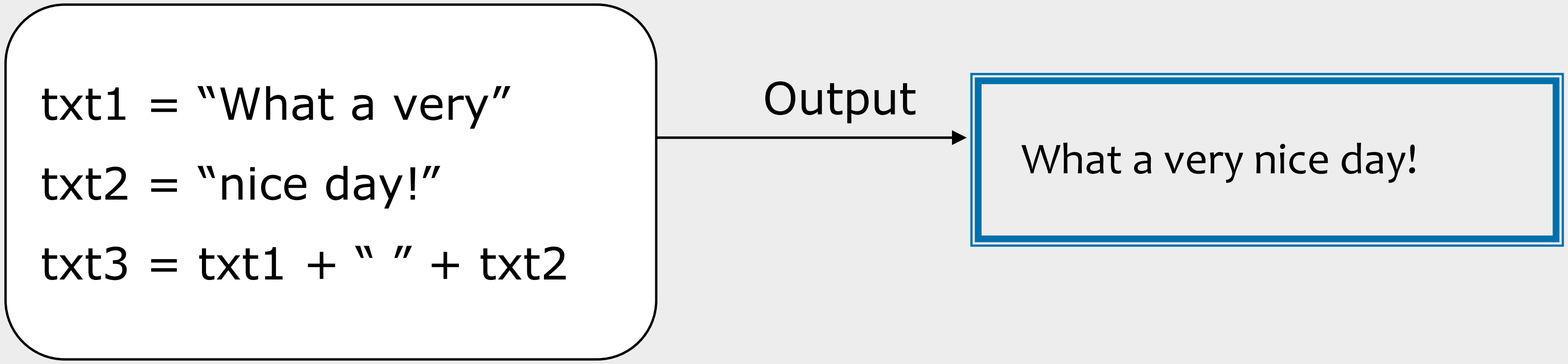
String Operator



```
txt1 = “What a very”
txt2 = “nice day!”
txt3 = txt1 + txt2
```

Output

```
What a verynice day!
```



```
txt1 = “What a very”
txt2 = “nice day!”
txt3 = txt1 + “ ” + txt2
```

Output

```
What a very nice day!
```

Typeof Operator

typeof	undefinedvariable	“undefined”
typeof	33	“number”
typeof	“abcdef”	“string”
typeof	true	“boolean”
typeof	null	“object”

Demo

Typeof_ex.html

Control Structures and Loops

JavaScript supports the usual control structures:

- the conditionals:
- if,
- if...else
- If … else if … else
- switch

iterations:

- for
- while

The if Statement

```
if(condition) {
        statement 1
} else {
        statement 2
}
```

```
if(a>10) {

document.write(“Greater than 10”)

} else {

document.write(“Less than 10”)

}
```

```
document.write( (a>10) ? “Greater than 10” : “Less than 10” );
```

The Switch Statement

Syntax

```
switch (variable) {
  case outcome1 :{
//stmts for outcome 1
 break; }
  case outcome2 :{
 //stmts outcome 2
 break; }
  …
default: {
 //none of the outcomes
 is chosen }
```

The Switch Statement

Code Snippet

```
switch (day) {
  case “Monday” : {
document.write(“weekday”)
break;}
 case “Saturday”: {
document.write(“weekday”)
break}
  …
 default: {
document.write(“Invalid day of the week”)
}
```

The for Statement

Syntax

```
for( [initial expression;][condition;][increment expression] )
 {        statements
}
```

- Code Snippet

```
for(var i=0;i<10;i++){
document.write("Hello");}
```

The while Statement (contd..)

Syntax

```
while(condition) {
		statements
}
```

- Code Snippet

```
while(i<10) {
document.write("Hello");
   i++;}
```

The break and continue Statements

break

- Writing break inside a switch, for, while control structure will cause the program to jump to the end of the block. Control resumes after the block, as if the block had finished

continue

- Writing continue inside a loop will cause the program to jump to the test condition of the structure and re-evaluate and perform instruction of the loop. Control resumes at the next iteration of the loop

Demo

For_ex.html

JavaScript Functions

- The function statement

```
function myFunction (arg1, arg2, arg3)
    {                                        //The return keyword returns a value.
          statements
          return
     }
```

- How to call a function

```
myFunction( "abc", "xyz", 4 )
                    or
myFunction()
```

Argument Arrays and How to call a Function

Syntax for the arguments array:

```
arguments[index]
functionName.arguments[index]
```

index – ordinal number of the argument starting at zero
arguments.length – Total number of arguments

The Function Statement (Contd..)

Syntax

```
function myConcat(separator) {
        result = “”
        for(var index=1; index<arguments.length;index++) {
                result += arguments[index] + separator
        }
        return result
  }
```

Co

```
myConcat( “,” , “red” , “orange” , “blue”)
// returns “red, orange, blue”
```

Predefined Functions

eval:

- Evaluates a string of JavaScript code without reference to a particular object.

```
eval (expr)
        where expr is a string to be evaluated
```

isFinite:

```
isFinite (number)
        where number is the number to evaluate
```

Predefined Functions (Contd..)

- isNaN :
  - Evaluates an argument to determine if it is “NaN” (not a number)

```
isNaN (testValue)
        where testValue is the value you want to evaluate
```

Predefined Functions (Contd..)

- parseInt and parseFloat
  - Returns a numeric value for string argument.

```
parseInt (str)
parseFloat (str)
```

```
    parseInt(str, radix)
//returns an integer of specified radix of the string argument
```

2.4: JavaScript Functions

Predefined Functions (Contd..)

- Number and string
  - Converts an object to a number or a string.

```
Number (objectReference)
String (objectReference)
```

```
today = new Date (430054663215)
now = String(today)
// returns "Thu Aug 18 04:37:43 GMT-0700 (PDT) 1983"
```

Global and Local Variables

- Code Snippet for scope of variables

```
<script>
      var companyName="CAPGEMINI"
      function displayName(){
      var employeeName="Tom"
      document.write("Welcome to "+companyName+", +employeeName)
       }
      </script>
```

Global Variable

Local Variable

Global and Local Variables

Variables that exist only inside a function are called Local variables

The values of such Local variables cannot be changed by the main code or other functions

Variables that exist throughout the script are called Global variables

Their values can be changed anytime in the code and even by other functions

Demo

If_else.html
Switch_ex.html
For_ex.html
Break_con_ex.html
Fun_ex.html
Num_string_fun.html

Lab

Lab 2 :
The JavaScript language

Summary

Data Types & Variables

- Numbers, Strings, Boolean, and Null

Operators & Expressions

Functions

Predefined Functions

- eval, isFinite, isNAN, parseInt & parseFloat, Number & String

Global and Local variables

Review Question

Question 1: Which of the following two variable scopes is supported by JavaScript:

- Global, Local
- Functional, Non functional
- Static, Dynamic

Question 2: The eval function evaluates a string of JavaScript code without reference to a particular object.

- True/False

Review Question: Match the Following

1. Loop statements
2. Arithmetic operators
3. Predefined function
4. Assignment operators
5. Logical operators

1. isNan
2. +=, -=
3. &&,||
4. For, While
5. ++, --, %, *