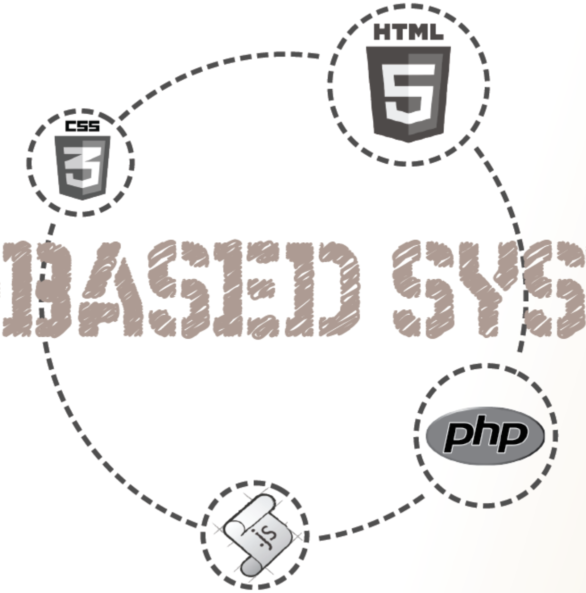
WEB-BASED SYSTEMS

IS 323



>> JavaScript: Functions, Arrays and Objects

(c) Dr. Mohammed Misbhauddin

FUNCTIONS IN JS

Function Declaration

- Mainly used for event-handling in Web Development
- Can also be called from other functions (Reuse)

SYNTAX

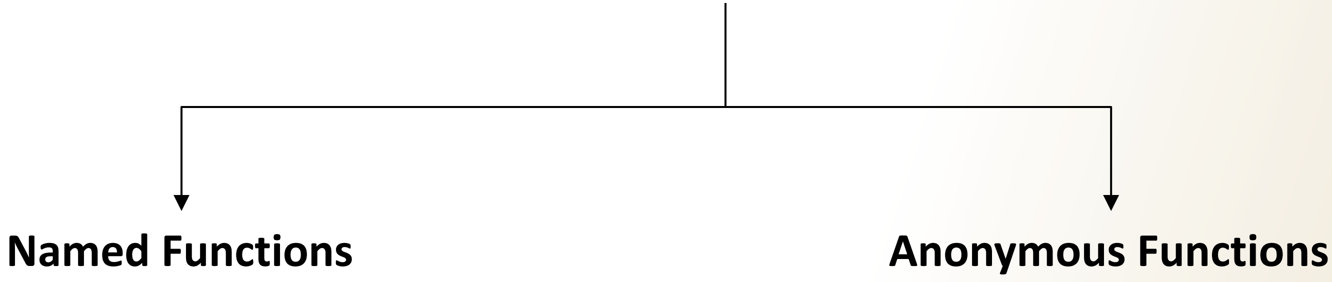
```
 keyword
function name(param1, param2,…..)
{

}
```

- You can use as many parameters as you like
- Can also return values (numbers, strings, Boolean)
  - Use **return** statement

```
function name(parameters)
{
     return b;
}
```

Functions in JS



Named Functions

Anonymous Functions

Named Functions: Functions that are given a name so that we can call them later in the code.

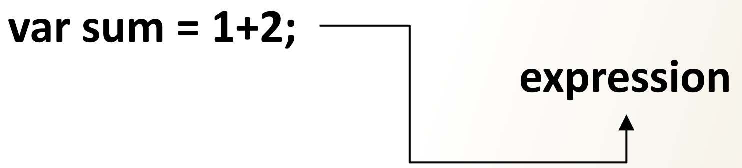
Function declaration

```
function area( width, height)
{
  return width* height;
}

var size = area(3, 4);
```

Anonymous Functions

Typically, after an = sign, the JavaScript interpreter expects to see an **expression** such as



var sum = 1+2;

expression

If there is a function declared after the equal sign, this is called a **function expression.**

In other words, it is function stored in a variable.

Anonymous Functions

```
var area = function( width, height)
{
   return width* height;
}

var size = area(3, 4);
```

- **Anonymous functions – functions created using the expression**
- **These are functions with no name**
  - Used mainly in calls and event handling
  - Used to create immediately invoked expressions (next slide)

Anonymous Functions

- **An important use of anonymous function**
- **Immediately Invoked Function Expression (IIFE)**
  - Pronounced “iffy”
  - They are executed once

```
var area = (function( width, height)
{
  return width* height;
}());
```

🟦 **Grouping parenthesis –** to treat it as an expression

🟥 **Final parenthesis –** to call the function immediately

Anonymous Functions for Event Handling

- **Using onclick function**

```
button.onclick = function(){


}
```

- Using the addEventListener() function

```
button.addEventListener("click", function(){


});
```

Note: Using addEventListener(), you can add as many listeners as possible. But this function has limited IE support

Arrays

- **Store multiple values in a Single Object/Variable**

SYNTAX

var array-name = **new** Array();

keyword

Built-in

<u>SYNTAX for Initializers</u>

Elements

var array-name = **new** Array(value1, value2,……….);

[0] *[1]* *Index*

<u>Other Initialization Method</u>

[0]…….[size-1]

var array-name = **new** Array(*size*);
array-name[*index*] = "value";

Values in Arrays

Accessed as if they are in a **numbered** list.
Important: Numbering of this list starts from zero (not one)

NUMBERING ITEMS IN AN ARRAY

Each item is given a number automatically called as **index**

```
var colors;
colors = [‘white’, ‘black’, ‘custom’];
```

INDEX	VALUE
0	white
1	black
2	custom

Values in Arrays

ACCESSING ITEMS IN AN ARRAY

Accessing requires specification of array name along with index number

USAGE: array-name[index-value]

Note: Index value should be in-bounds otherwise "*undefined*" error

```
itemThree = colors[2];
```

Accesses the third element in the array colors

Arrays – Literal Method

- **Another way of creating Arrays**

SYNTAX

$\underbrace{\textbf{var}}_{\textit{keyword}}$ array-name = [element-1, element-2, ………]

Types of Arrays

Types of Arrays

1. **Heterogeneous Arrays:** Arrays with elements of different data types

   var lists = [1, “name”, true];

2. **Multi-dimensional Arrays:** Arrays with multiple rows

   var lists = [[1, 2], [3, 4], [5, 6];

3. **Jagged Arrays:** Arrays with multiple rows, each row with different no. elements

   var lists = [[1, 2], [3, 4, 5], [5, 6, 7, 8];

Arrays - Functions

Arrays are Objects. Hence, they have properties & methods *(just like strings)*

- **Properties**
  - **.length –** returns the length of the array

- **Methods**
  - array1.**concat(**array2**)** – *joins two arrays*
  - **.join()** – *turn an array into a string*
  - **.pop() –** *Returns the last element & deletes it*
  - **.push(**element**) –** *Adds element to the end of the array*
  - **.reverse() –** *Reverses the array*
  - **.sort()** - *Sorts the Array*

Remember Stacks

Displaying Arrays

- **Display an Array**
  - Use for loop – for(var i=0; i<array-name.length;i++)
  - Use Join
  - Use for loop again – for(var i in array-name)

For Loop Variation

```
for(var i=0; i<array-name.length; i++)
```

- Can loop from any point to any other point in the given array.
- Starting point is the initialization
- Ending point is the condition satisfaction

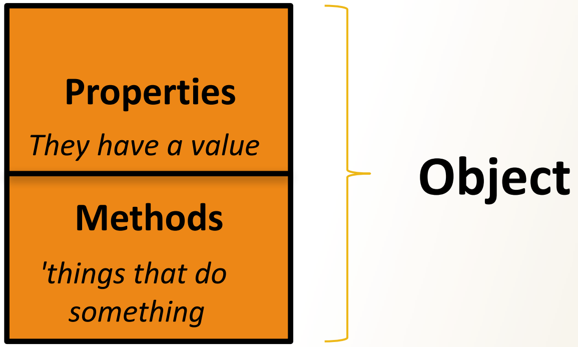
```
for(var i in array-name)
```

- Used when you want to loop the entire array
- The value of i goes from 0 to the length-1 of the array

OBJECTS IN JS

Object-Oriented JS

JavaScript is Object-Oriented



Example

var abc = “Hello World”;
abc.length;

property

document.write(“Hello World”);

method

Object Definition

Objects

- Objects allow us to represent in code real world things and entities
- Such as a person or bank account

SYNTAX

```
 keyword

var object-name =
{
        ..........
}
```

Key-Value: Properties of an Object

Properties

- Each piece of information we include in an object is known as a **property**.
- Each property has a **key**, followed by : and then the **value** of that property

SYNTAX

```
keyword
var object-name =
{
        key: value,
}
```

Access value of a key

object-name[“key”];
or
object-name.key

Note: Separate each property using a comma (,) and not a semicolon as in Java
Key names can have quotations. But if there is a space in the name (first name), quotation is necessary

Function Assignment to Objects

Methods

- A **method** is just like a function associated with an object

SYNTAX

keyword

```
var object-name =
{
        key: value,
        function-name: function() {
                    -----------
        }
}
```

Access the function

```
object-name.function-name();
```

this Keyword

This keyword

- Used to refer to the current object inside the function in the object

```
var hotel =
{
        rooms: 50,
        booked: 30,
        checkAvailability: function() {
                return this.rooms – this.booked;
        }
}
```

Arrays: Exercise

- Download the Exercise_Instruction.pdf file.
- Download the Exercise HTML file
- Follow the instructions from the PDF file and complete the exercise