



پوهنتون کاردان
KARDAN UNIVERSITY

Faculty Of Engineering and Technology
Department of Computer Science
(Lecture-02)

Khalilullah Akbari

Lecturer

Department of Computer Science

Email: khalil.akbari18@gmail.com, k.akbari@kardan.edu.af

Mobile: +93(0) 729908855

Web Page Designing & Development:



- Array methods
- Object
- Array of objects
- String methods
- Date objects



Array Methods

- push()

```
let fruits = ['apple', 'banana'];  
fruits.push('orange');  
console.log(fruits); // ['apple', 'banana', 'orange']
```

- Pop()

```
let fruits = ['apple', 'banana', 'orange'];  
let lastFruit = fruits.pop();  
console.log(lastFruit);           // 'orange'  
console.log(fruits);              // ['apple', 'banana']
```



Array Methods

- Shift()

```
let fruits = ['apple', 'banana', 'orange'];  
let firstFruit = fruits.shift();  
console.log(firstFruit);           // 'apple'  
console.log(fruits);               // ['banana', 'orange']
```

- Unshift()

```
let fruits = ['banana', 'orange'];  
fruits.unshift('apple');  
console.log(fruits);               // ['apple', 'banana', 'orange']
```



Array Methods

- `slice()`

```
let fruits = ['apple', 'banana', 'orange', 'grape'];  
let citrus = fruits.slice(1, 3);  
console.log(citrus);           // ['banana', 'orange']
```

- `includes()`

```
let fruits = ['apple', 'banana', 'orange'];  
let hasBanana = fruits.includes('banana');  
console.log(hasBanana);       // true
```



Array Methods

- `map()`

Creates a new array populated with the results of calling a provided function on every element in the calling array.

```
let numbers = [1, 2, 3, 4];  
let doubled = numbers.map(num => num * 2);  
console.log(doubled);           // [2, 4, 6, 8]
```

- `filter()`

Creates a new array with all elements that pass the test implemented by the provided function.

```
let numbers = [1, 2, 3, 4, 5];  
let evenNumbers = numbers.filter(num => num % 2 === 0);  
console.log(evenNumbers);       // [2, 4]
```



JavaScript Closure

- A **closure** happens when an inner function **remembers and can access variables from its outer function even after the outer function has finished executing.**

```
function createCounter() {  
  let count = 0;  
  
  return function () {  
    count++;  
    return count;  
  };  
}  
  
const counter = createCounter();  
console.log(counter()); // 1  
console.log(counter()); // 2  
console.log(counter()); // 3
```



JavaScript Objects?

- JavaScript is an object-based language and in JavaScript almost everything is an object or acts like an object.
- A JavaScript object is just a collection of named values. These named values are usually referred to as properties of the object.



Object Keynotes

Concept	Example
Create object	<code>const user = { name: "Ali" }</code>
Access property	<code>user.name</code>
Bracket notation	<code>user["name"]</code>
Add property	<code>user.email = "..."</code>
Modify property	<code>user.age = 25</code>
Delete property	<code>delete user.age</code>
Method	<code>user.login()</code>
Destructuring	<code>const { name } = user</code>



Creating Objects

- An object can be created with curly brackets `{}` with an optional list of properties. A property is a "key: value" pair, where the key (or *property name*) is always a string, and value (or *property value*) can be any data type

```
<script>

    emp={
        id:102,
        name:"Belal",
        salary:40000
    }
    document.write(emp.id+" "+emp.name+" "+emp.salary);

</script>
```



Accessing Object's Properties

- To access or get the value of a property, you can use the dot (.), or square bracket ([]) notation, as demonstrated in the following example:

```
var book = {  
    "name": "C++",  
    "author": "Saleem",  
    "year": 2020  
};  
// Dot notation  
document.write(book.author); // Prints: saleem  
// Bracket notation  
document.write(book["year"]); // Prints: 2020
```



Accessing Object's Properties using loop

```
var person = {  
    name: "Ali",  
    age: 28,  
    gender: "Male",  
};  
for(var x in person)  
{  
    document.write(x+"-----"+person[x]+"<br>");  
}
```



Array of object

- Objects are elements of array:

```
var objs = [  
    {id:12,name:"Ali"},  
    {id:45,name:"Samar"},  
    {id:23,name:"Haron"}  
]
```



Printing Array of objects into table

```
var objs = [  
    {id:12,name:"Ali"},  
    {id:45,name:"Samar"},  
    {id:23,name:"Haron"}  
]  
  
document.write("<table>");  
for(let x of objs){  
    document.write("<tr><td>" + x.id + "</td><td>" + x.name + "</td></tr>");  
}  
document.write("</table>");
```



What is String in JavaScript

- A string is a sequence of letters, numbers, special characters and arithmetic values or combination of all.
- Strings can be created by enclosing the string literal (i.e. string characters) either within single quotes (') or double quotes ("), as shown in the example below:

Example

```
var myString = 'Hello World!'; // Single quoted string
```

```
var myString = "Hello World!"; // Double quoted string
```



Methods of String



replace()	It replaces a given string with the specified replacement.
slice()	It is used to fetch the part of the given string. It allows us to assign positive as well negative index.
toLowerCase()	It converts the given string into lowercase letter.
toUpperCase()	It converts the given string into uppercase letter.
toString()	It provides a string representing the particular object.
split()	It splits a string into substring array, then returns that newly created array.
trim()	It trims the white space from the left and right side of the string.



Methods of String

// String Declaration

```
var s= "my string";
```

// String Methods

```
document.write("<b>Original String </b>_____"+s+"<br><br>");
```

```
document.write("<b>Replace function </b>_____"+s.replace("my", "this is")+"<br>");
```

```
document.write("<b>Slice function </b>_____"+s.slice(-3)+"<br>,<br>" );
```

```
document.write("<b>toLowerCase function </b>_____"+s.toLowerCase()+"<br><br>");
```

```
document.write("<b>toUpperCase function </b>_____"+s.toUpperCase()+"<br><br>");
```

```
document.write("<b>Split function </b>_____"+s.split("")+"<br><br>");
```

```
document.write("<b>trim function </b>_____"+s.trim()+"<br><br>");
```



JavaScript - The Date Object

- The Date object is a datatype built into the JavaScript language. Date objects are created with the **new Date()** as shown below.

```
new Date( )  
new Date(milliseconds)  
new Date(datestring)  
new Date(year,month,date[,hour,minute,second,millisecond ])
```



JavaScript Get Date Methods

- `getFullYear()` Get the year as a four digit number (yyyy)
- `getMonth()` Get the month as a number (0-11)
- `getDate()` Get the day as a number (1-31)
- `getHours()` Get the hour (0-23)
- `getMinutes()` Get the minute (0-59)
- `getSeconds()` Get the second (0-59)
- `getMilliseconds()` Get the millisecond (0-999)
- `getTime()` Get the time (milliseconds since January 1, 1970)
- `getDay()` Get the weekday as a number (0-6)
- `Date.now()` Get the time. ECMAScript 5.



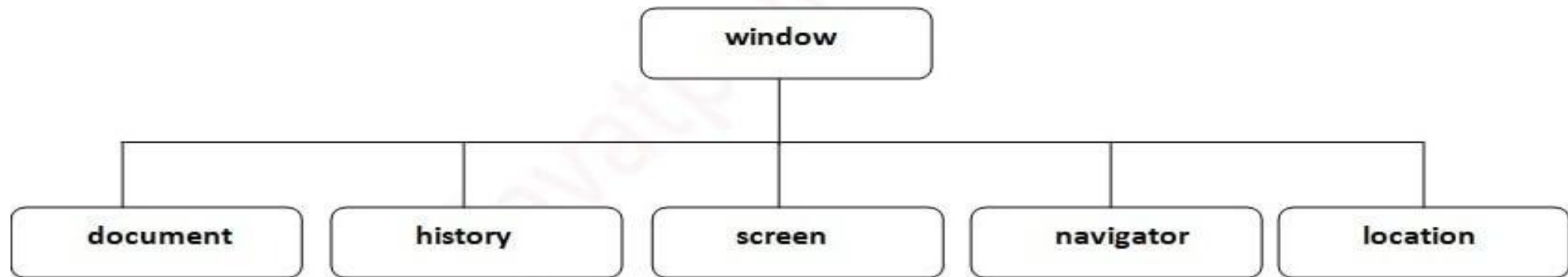
JavaScript Set Date Methods

- `setDate()` Set the day as a number (1-31)
- `setFullYear(1919)` Set the year (optionally month and day)
- `setHours()` Set the hour (0-23)
- `setMilliseconds()` Set the milliseconds (0-999)
- `setMinutes()` Set the minutes (0-59)
- `setMonth()` Set the month (0-11)
- `setSeconds()` Set the seconds (0-59)
- `setTime()` Set the time



Browser Object Model

- The **Browser Object Model** (BOM) is used to interact with the browser. The default object of browser is window means you can call all the functions of window by specifying window or directly.
- For example: You can use a lot of properties (other objects) defined underneath the window object like document, history, screen, navigator, location, inner Height, inner Width,



Properties and methods of windows object

- **window.innerHeight** - the inner height of the browser window (in pixels)
- **window.innerWidth** - the inner width of the browser window (in pixels)
- **Alert()** displays the alert box containing message with ok button.
- **confirm()** displays the confirm dialog box with ok and cancel button.
- **prompt()** displays a dialog box to get input from the user.
- **open()** opens the new window.
- **close()** closes the current window.
- **setTimeout()** performs action after specified time like calling function,





پوهنتون کاردان
KARDAN UNIVERSITY

Thank You...!