



SNS COLLEGE OF TECHNOLOGY



(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

JAVA SCRIPT : Objects

COURSE

19CAT601 -
Web Programming
Essentials

UNIT II

CSS

I Semester /
I MCA



Java Script Objects



- JS is object based language and Everything is an object
- Object is an entity having state and behavior (properties and method)
- JavaScript is template based not class based.
- Here, we don't create class to get the object
- Objects can be containers for data values
- Create own objects / use built in objects



Car Object



Properties

Make
Color
Model
Year
Price
Fuel type



Methods

Start
Apply break
Speed
Park



Programming paradigms



- **Procedural** programmers concentrate on writing functions
- Group common task into functions
- Group functions to form programs

- **Object-oriented** programmers concentrate on creating their own user-defined types called classes
- Each class contains data as well as the set of functions that manipulate that data and provide services to clients



Object Technology




Data
components
called
properties

Function
components
called Methods

- Nouns in a system specification help you determine the set of classes
- Classes have relationship with other classes
- Groups of related classes are often packaged
- Build much of the new software you'll need by combining existing classes



Built in Objects



Array

- **Properties**
 - length
- **Methods**
 - sort()
 - reverse()
 - push()
 - pop()



String

- **Properties**
 - length
-



Math

- **Properties**
 - pi
- **Methods**
 - round()
 - random()



Date

- **Properties**
 -
- **Methods**
 - getDate()/setDate()
 - getDay() /setDay()
 - getMonth()/SetMonth()
 - getFullYear()/setFullYear()



Math object



Method	Description
<code>abs(x)</code>	absolute value of x
<code>ceil(x)</code>	rounds x to the smallest integer not less than x
<code>cos(x)</code>	trigonometric cosine of x (x in radians)
<code>exp(x)</code>	exponential method e^x
<code>floor(x)</code>	rounds x to the largest integer not greater than x
<code>log(x)</code>	natural logarithm of x (base e)
<code>max(x, y)</code>	larger value of x and y
<code>min(x, y)</code>	smaller value of x and y
<code>pow(x, y)</code>	x raised to power y (x^y)
<code>round(x)</code>	rounds x to the closest integer
<code>sin(x)</code>	trigonometric sine of x (x in radians)
<code>sqrt(x)</code>	square root of x
<code>tan(x)</code>	trigonometric tangent of x (x in radians)



String object



- A string is a series of characters treated as a single unit
- It include letters, digits and various special characters, such as +, -, *, /, and \$
- JavaScript supports Unicode



String object



charAt(index)	Returns a string containing the character at the specified index
charCodeAt(index)	Returns the Unicode value of the character at the specified index
Concat(string)	Concatenates its argument to the end of the string that invokes the method
fromCharCode(value1, value2)	Converts a list of Unicode values into a string containing the corresponding characters.
indexOf(substring, index)	Searches for the first occurrence of substring starting from position index in the string that invokes the method
lastIndexOf(substring, index)	Searches for the last occurrence of substring starting from position index and searching toward the beginning of the string that invokes the method
replace(searchString, replaceString)	Searches for the substring searchString, and replaces the first occurrence with replaceString and returns the modified string,



String object



slice(start, end)	Returns a string containing the portion of the string from index start through index end
split(string)	Splits the source string into an array of strings (tokens), where its string argument specifies the delimiter
substr(start, length)	Returns a string containing length characters starting from index start in the source string
substring(start, end)	Returns a string containing the characters from index start up to but not including index end in the source string.
toLowerCase()	Returns a string in which all uppercase letters are converted to lowercase letters
toUpperCase()	Returns a string in which all lowercase letters are converted to uppercase letters



String object



Methods that generate XHTML tags

anchor(name)	Wraps the source string in an anchor element (<code><a></code>) with name as the anchor name.
fixed()	Wraps the source string in a <code><tt></tt></code> element (same as <code><pre></pre></code>).
link(url)	Wraps the source string in an anchor element (<code><a></code>) with url as the hyperlink location.
strike()	Wraps the source string in a <code><strike></strike></code> element.
sub()	Wraps the source string in a <code><sub></sub></code> element.
sup()	Wraps the source string in a <code><sup></sup></code> element.



Date Object



- provides methods for date and time manipulations
- Based on a local time zone and a UTC version
- Empty parentheses after an object name indicate a call to the object's constructor with no arguments

```
var dte = new Date();  
document.writeln(dte);
```



Date Object



setFullYear(y, m, d) setUTCFullYear(y, m, d)	Sets the year in local time or UTC. The second and third arguments representing the month and the date are optional.
setHours(h, m, s, ms) setUTCHours(h, m, s, ms)	Sets the hour in local time or UTC. The second, third and fourth arguments, representing the minutes, seconds and milliseconds (optional)
setMilliseconds(ms) setUTCMilliseconds(ms)	Sets the number of milliseconds in local time or UTC.
setMinutes(m, s, ms) setUTCMinutes(m, s, ms)	Sets the minute in local time or UTC. The second and third arguments, representing the seconds and milliseconds, are optional



Date Object



setSeconds(s, ms)	Sets the second in local time or UTC. The second argument, representing the milliseconds, is optional
setUTCSeconds(s, ms)	
setTime(ms)	Sets the time based on its argument—the number of elapsed milliseconds since January 1, 1970.
toLocaleString()	Returns a string representation of the date and time in a form specific to the computer's locale
toUTCString()	Returns a string representation of the date and time in the form: 15 Sep 2007 15:47:22 UTC
toString()	Returns a string representation of the date and time in a form specific to the locale of the computer
valueOf()	The time in number of milliseconds since midnight, Jan 1, 1970



Boolean Object



- Boolean is object wrapper for boolean true/false values
- When a boolean value is required, JavaScript automatically creates a Boolean object to store the value
- JavaScript programmers can create Boolean objects explicitly

```
var b = new Boolean( booleanValue );
```

toString()	Returns the string "true" if the value of the Boolean object is true; otherwise, returns the string "false"
valueOf()	Returns the value true if the Boolean object is true; otherwise, returns false.



Number object



- JS automatically creates Number objects to store numeric values
- Can create a Number object with the statement

```
var n = new Number( numericValue );
```

- Although you can explicitly create Number objects, normally they are created when needed by the JavaScript interpreter



Number object



toString(<i>radix</i>)	Returns the string representation of the number
valueOf()	Returns the numeric value.
Number.MAX_VALUE	This property represents the largest value that can be stored in a JavaScript program—approximately 1.79E+308.
Number.MIN_VALUE	This property represents the smallest value that can be stored in a JavaScript program—approximately 5.00E−324.
Number.NaN	This property represents <i>not a number</i> —a value returned from an arithmetic expression that does not result in a number
Number.NEGATIVE_INFINITY	This property represents a value less than -Number.MAX_VALUE
Number.POSITIVE_INFINITY	This property represents a value greater than Number.MAX_VALUE



Document Object



- For manipulating the document that is currently visible in the browser window

getElementById(id)	Returns the DOM node representing the XHTML element whose id attribute matches id.
write(string)	Writes the string to the XHTML document as XHTML code.
writeln(string)	Writes the string to the XHTML document as XHTML code and adds a newline character at the end.
cookie	A string containing the values of all the cookies stored on the user's computer for the current document. See Section 11.9, Using Cookies.
lastModified	The date and time that this document was last modified.



Window object



- Provides methods for manipulating browser windows

Open(url, name, options)	Creates a new window with the URL of the window set to url
Prompt(prompt, default)	Displays a dialog box asking the user for input
close()	Closes the current window and deletes its object from memory
focus()	This method gives focus to the window
blur()	This method takes focus away from the window
window.document	An document object represents document currently inside the window
window.closed	return boolean value that is set to true if the window is closed
window.opener	Contains window object that opened the current window, if it exists.



Define object



- Object can be defined by object literal
- Property can be accessed by ***objectname.propertyname***

```
var car = {  
  make:"Maruthi", type:"SUV", model:"Breeza", color:"white"  
  Name: check()  
    {  
      return this.make+" "+this.model;  
    }  
};
```

this keyword refers current object



Create object using new



- Variable is declared with the keyword "new", the variable is created as an object (called, instance)

```
<script>  
    var x = new employee();  
    emp.code="X101";  
    emp.name="Xxxx";  
    emp.salary=10000;  
    document.write(emp.code+" -" emp.name" + " - "+  
emp.salary);  
</script>
```



Create object using object constructor



- Create function with arguments
- Each argument value can be assigned in the current object by using this keyword

```
<script>  
    function emp(code, name, salary)  
    {  
        emp.code=code;  
        emp.name=name;  
        emp.salary=salary;  
    }  
var x= new emp("X102, "YY", 21000);  
document.write(x.code+" –" +x.name + " – "+ x.salary);  
</script>
```



Define methods



```
<script>
  function emp(code, name, salary)
  {
    emp.code=code; emp.name=name; emp.salary=salary;
    this.changeSal=changeSal;
    function changeSalary(newSalary)
    {
      this.salary=newSalary;
    }
    var x= new emp("X102, "YY", 21000);
    document.write(x.code+" –" +x.name + " – "+ x.salary);
    x.changeSalary(32000);
    document.write(x.code+" –" +x.name + " – "+ x.salary);
  }
</script>
```



String



- **string** is an object that represents a sequence of characters
- There are 2 ways to create string in JavaScript
 1. By string literal
 2. By string object (using new keyword)

```
var x= 20  
var fname="myname";
```




Examples

