INT222 - Internet Fundamentals

Week 11: Math Object and built-in functions

Agenda

- Math Object
- built-in functions

JavaScript Built-in Objects

- JavaScript provides many predefined, built-in objects that enable you to work with Strings and Dates, perform mathematical operations, and etc.:
  - String
  - Array
  - Date
  - Math
  - Number
  - Boolean
  - RegExp

Math functions (common methods of the Math object)

- Math.max(ident_1, ident_2)
  - the maximum of n numbers
  - e.g. alert(Math.max(0.52, 1)); // 1
- Math.min(ident_1,ident_2)
  - the minimum of n numbers
  - e.g. alert(Math.min(0.52, 1)); // 0.52
- Math.pow(ident_1, ident2)
  - ident_1 to the power ident_2
  - e.g. alert(Math. pow(2, 8)); // 256
- Math.sqrt(ident_1)
  - square root of
  - e.g. alert(Math. sqrt(9)); // 3

Rounding floating-point

- Math.ceil(ident_1)
  - integer closest to and not less than
  - e.g.
    ```
    alert( Math.ceil(0.52) ); // 1
    alert( Math.ceil(0.49) ); // 1
    ```
- Math.floor(ident_1)
  - integer closest to and not greater than
  - e.g.
    ```
    alert( Math.floor(0.52) ); // 0
    ```
- Math.round(ident_1)
  - integer closest to
  - e.g.
    ```
    alert( Math.round(0.52) ); // 1
    alert( Math.round(0.49) ); // 0
    alert( Math.round(0.5) );   // 1
    ```

Generating Random Number

- Math.random() - pseudorandom number
  - Return a floating point number between 0 (inclusive) and 1 (exclusive)
  - e.g. alert(Math.random()); // 0.03517110995016992

- Generating number 1 to 10

```
Math.floor((Math.random()*10)+1);
```

JavaScript
Built-in Functions/Methods

escape(myString) function

- Used to encode string, same as encodeURI() .
- The function takes a non-alphanumeric string as its argument and returns the ASCII value (in hexadecimal form) for each character in the string preceded by a % sign.
- If the string includes alphanumeric characters or -+*/_.@, those characters are returned unchanged.
- Blank characters are returned as %20

Examples

- alert(escape("qwertyuiopfghjklzxcvbnm"));
  // qwertyuiopfghjklzxcvbnm
- alert(escape("QWERTYFGHJKLZXCVBNM"));
  // QWERTYFGHJKLZXCVBNM
- alert(escape("1234567890-+*/_.@"));
  // 1234567890-+*/_.@
- alert(escape(" ~`!#$%^&()=\|\\[]{};':\",<>?"));

//
%20%7E%60%21%23%24%25%5E%26%28%29%3D%7C
%5C%5B%5D%7B%7D%3B%27%3A%22%2C%3C%3E%3
F

eval(myString) function

- The eval() function uses one argument: a string.
  - If the string is an expression, eval() evaluates/executes the expression.
  - If the string is made up of JavaScript statements, eval() performs the statements.

Example

```
var x = 10;
var y = 20;
var a = eval("x*y") + "\n";
var b = eval("2+2") + "\n";
var c = eval("x+17") + "\n";

var res = a + b + c;
alert(res);
```

Result:

```
200
4
27
```

isNaN(myString) function

- The isNaN() function is used to determine if an argument is "NaN" (not a number).
- example

```
alert( isNaN("123") );  // false
alert( isNaN(123) );     // false
alert( isNaN("123 456 789") );  // true
alert( isNaN("+123") );   //  false
alert( isNaN("123+") );   //  true
alert( isNaN(" 123 ") );   //   false
```

parseFloat(myString) function

- The parseFloat() function parses a string and returns a floating point number.

- If a character other than a numeral, a sign (+ or -), or an exponent is found, the function returns the value up to that point.

- If the first character in the string cannot be converted to a number, the function returns "NaN".

Example

```
alert(  parseFloat("15.25")  );         // 15.25
alert(  parseFloat("0.000345")  );  // 0.000345
alert(  parseFloat("0.00159+E")  ); // 0.00159
alert(  parseFloat(" 1234")  );   // 1234
alert(  parseFloat("x 1234")  );  // NaN
alert(  parseFloat("1 2 3 4")  );   // 1
alert(  parseFloat("1234 x 123")  );  // 1234
```

parseInt(myString) function

- The parseInt() function parses its first argument (a string), and then
- tries to return an integer of the specified radix (or base).
- If a number in the string is beyond the base, parseInt() ignores the rest of the characters and returns an integer value up to that point.

Examples

- base 10 (decimal) examples

  parseInt('15', 10) returns 15
  parseInt('15') returns 15
  parseInt(15.99, 10) returns 15
  parseInt('15*3', 10) returns 15
  parseInt('Hello') returns NaN

- base 16 (hex) examples

  parseInt('F', 16) returns 15
  parseInt('FXX123', 16) returns 15

Examples

- base 8 (octal) example

  parseInt('17', 8) returns 15
  parseInt('18', 8) returns 1

- base 2 (binary) example

  parseInt('1111', 2) returns 15
  parseInt('**1**211', 2) returns 1

Note the following problems

parseInt('015',10)   with base 10 returns 15
parseInt('015',8)   with base 8 returns 13
parseInt('015',16)   with base 16 returns 21

parseInt('15')   with no base returns 15
- treated as decimal

parseInt('015')   with no base returns 15
- treated as octal

parseInt('0x15')   with no base returns 21
- treated as hex

parseInt(' 15') with a blank returns 15

unescape(myString) function

- The unescape() function is the exact opposite of the escape() function.
- Its argument is a string of ASCII values (in hexadecimal form), each preceded by a % sign. The function returns the character string.

```
var myString1 =
"%20%7E%60%21%23%24%25%5E%26%28%29%3
D%7C%5C%5B%5D%7B%7D%3B%27%3A%22";

unescape(myString1)
returns     ~`!@#$%^&()=|\[]{};':"
```

Number() function

- The Number() function returns the actual number value - when possible

```
var var1= new Boolean(true);
var var2= new Boolean(false);
var var3= new Date();
```

```
var1 = true
var2 = false
var3 = Mon Mar 17 2014 00:04:29 GMT-0400 (Eastern Daylight
Time)
Number(var1) = 1
Number(var2) = 0
Number(var3) = 1395029069226
```

Examples

```
var var4= new String("999");
var var5= new String("999 888");
var var6= "999";
var var7= "abc";
```

```
var4 = 999
var5 = 999 888
var6 = 999
var7 = abc
Number(var4) = 999
Number(var5) = NaN
Number(var6) = 999
Number(var7) = NaN
```

Parsing String to Number Without using functions

```
var str1 = "1234";
var num1 = str1 * 1;

alert(num1 + "\n" + typeof num1);

var str2 = "1234.5678";
var num2 = +str2;

alert(num2 + "\n" + typeof num2);
```

toFixed() Method

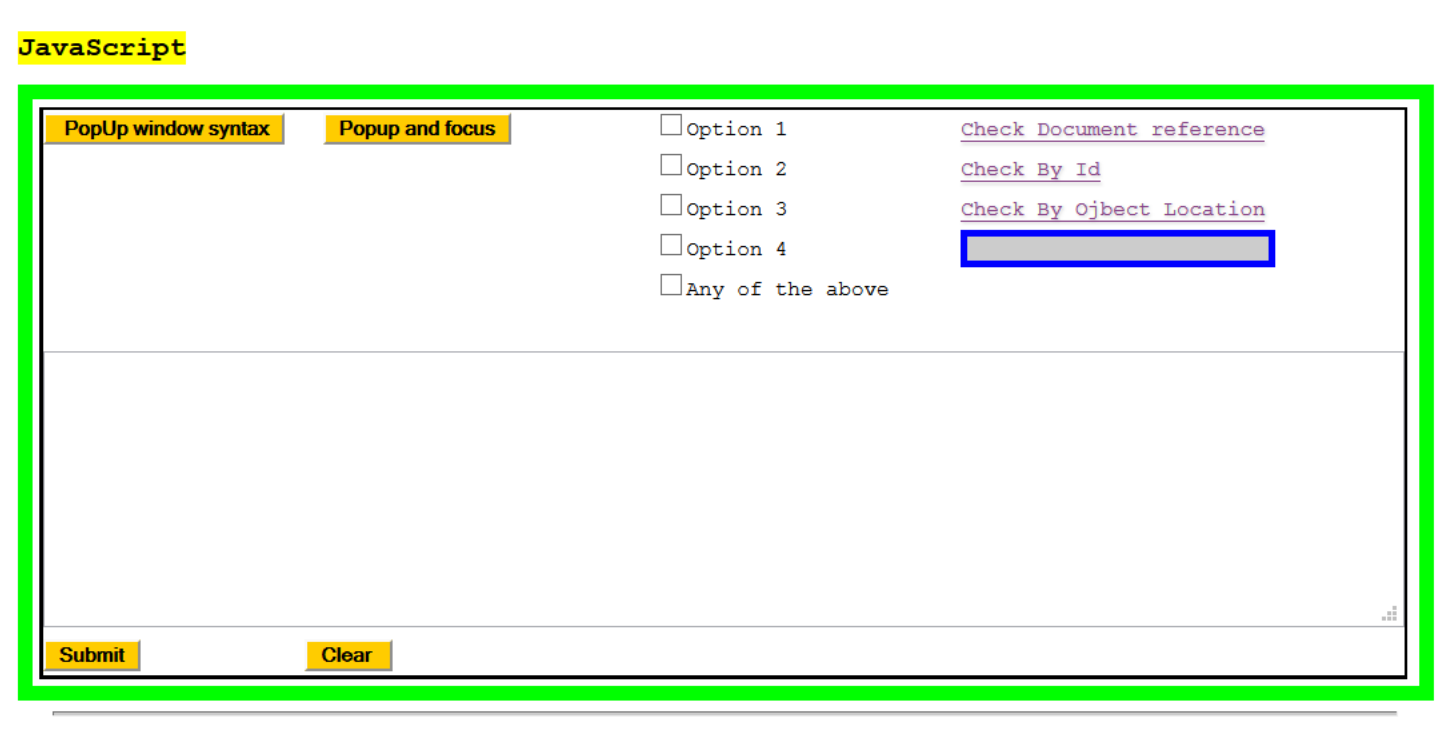
- toFixed() is the method of Number object.
- The toFixed() method formats a number to a specific number of digits to the right of the decimal.

```
    var amount = 165.25456;
```

```
amount.toFixed() is : 165
amount.toFixed(6) is : 165.254560
amount.toFixed(2) is : 165.25
```

Creating Popup window

- A demo:



Popup window syntax

- Generate a popup window - on the fly

```
messageWindow = window.open(' ','window target name',
                                             'window properties');
messageWindow.document.write(content);
messageWindow.document.close();
```

- Generate a popup window - Static

```
messageWindow=window.open('url','window target name',
                                             'window properties');
messageWindow.document.close();
```

Common sense logic

- Checkbox fields

```
<input type='checkbox' name='example4'  value='1'
          onclick='commonSense();' />Option 1<br />
<input type='checkbox' name='example4'  value='2'
          onclick='commonSense();' />Option 2<br />
<input type='checkbox' name='example4'  value='3'
          onclick='commonSense();' />Option 3<br />
<input type='checkbox' name='example4'  value='4'
           onclick='commonSense();' />Option 4<br />
<input type='checkbox' name='example4'  value='any'
          onclick='uncheckTheAbove();' />All of the above
```

Common sense logic

- JavaScript Code

```
 function commonSense() {
     document.formexample.example4[document.
              formexample.example4.length - 1].checked = false;
}

function uncheckTheAbove() {
    var numberOfCheckboxes2 =
              document.formexample.example4.length - 1;
    for (var j = 0; j < numberOfCheckboxes2 ; j++) {
        document.formexample.example4[j].checked = false;
    }
}
```

Thank You!