

CSC435: Web Programming
Lecture 8: JavaScript: Syntax,
Functions, DOM object

Bei Xiao

American University

Feb 17, 2014

Overview

- Syntax review
- Capture mouse movement
- Conditions and looping
- Functions
- Variable scope
- Document objects

What we have talked about

- Importing JavaScript

```
<script type="text/javascript"> JavaScript  
here </script>
```

- Data types: numbers, strings, arrays, Booleans
- Variables: `var x = y; var string= "hello world";`
- Operators: `+, -, x++, ++x, %;`
- Document object model: `window.alert;`
`document.body; document.write`

Booleans

- True or false: `var success = false;`
- No “quote” for Boolean value.
- Return “true” or “false”, “`10<3`”, `isNaN(“string”);`
- `new Boolean(value)`

```
var b1 = new Boolean(0); //false
```

```
var b2 = new Boolean(1); // true
```

```
var b3 = new Boolean (“”); //false
```

```
var b4= new Boolean("false"); ??
```

Syntax

- Be a clean coder, it can lead to hazard if not careful.
If-else statement, use blocks

```
if (expression)
{
    //code
}
else
{
    //code
};
```

- End appropriate lines with **semicolons**.
- Use either single or double quote, don't mis-match.

```
var s = "string here";
```

```
var s = 'We can have "hello world" in the browser';
```

Capture mouse movement

- The onClick event handler.

```
<input type="button" onClick="alert ('Please submit') value="submit" />;
```

- onMouseOver and on MouseOut Event Handlers

```
;
```

Looping

- For loops

```
for (var i = 0, i < n; i++) {  
  //code  
}
```

- While loops

```
var i = 0  
while (i < n) {  
  //code  
}
```

- Do While loops

```
do  
{  
  code block to be executed  
}  
while (condition);
```

Conditionals

- If-else statement

```
If (condition){  
  //code  
}  
elseif (condition) {  
  //code  
}  
else {  
  //code  
}
```

- Switch statement

```
switch(variable){  
  case 1: //code  
  break;  
  case 2: //code  
  Break;  
  default: //code  
}
```


Function

- Functions are **objects**.

```
function add(a,b) {  
    return a+b;  
}
```

Function

- Functions are **objects**, two ways of declaring

```
function fun(x,y) { // will take 2 parameters  
  //code  
};
```

```
var function fun(a,b,c) { //uses 3 parameters  
  //code  
};
```

- Call a function

```
fun("string", obj1);
```

```
Fun(obj1,obj2); //these two calls the same function
```

Function: simple example

```
function times(a,b) {  
  alert (a*b);  
};
```

```
times(3,4); //alerts '12'
```

Function: calling by clicking

```
<html>
<head>
<script>
function myFunction(name,job)
{
alert("Welcome " + name + ", the " + job);
}
</script>
</head>
<body>

<p>Click the button to call a function with
arguments</p>
<button onclick="myFunction('Harry
Potter','Wizard')">Try it</button>

</body>
</html>
```

Function: returning values

- Functions return undefined by default. But can return any value:

```
function cube(x) {  
  Return x * x * x;  
};
```

```
var answer = cube(3);
```

Or

```
alert(cube(3));
```

Function: test prime numbers

```
function isPrime(n)
// Assumes: n > 0
// Returns: true if n is prime, else false
{
  if (n < 2) {
    return false;
  }
  else if (n == 2) {
    return true;
  }
  else {
    for (var i = 2; i <= Math.sqrt(n); i++) {
      if (n % i == 0) {
        return false;
      }
    }
    return true;
  }
}
```

Exercise 1: compute area of a circle

- Write a function that compute the area of a circle from the radius.
- Definition: $\text{area} = \text{Pi} * \text{r}^2$; `Math.PI`, `Math.POW(2)`
- Using `rad = prompt("Enter the radius");` to ask for user input.
- Looping until observer click "cancel".
- Write both the input and the result out to the browser.

Function: recursion

- A recursive function is a function that calls itself, either directly or indirectly.

```
function c = func(a,b){  
    var x = "hello";  
    func(x);  
}
```

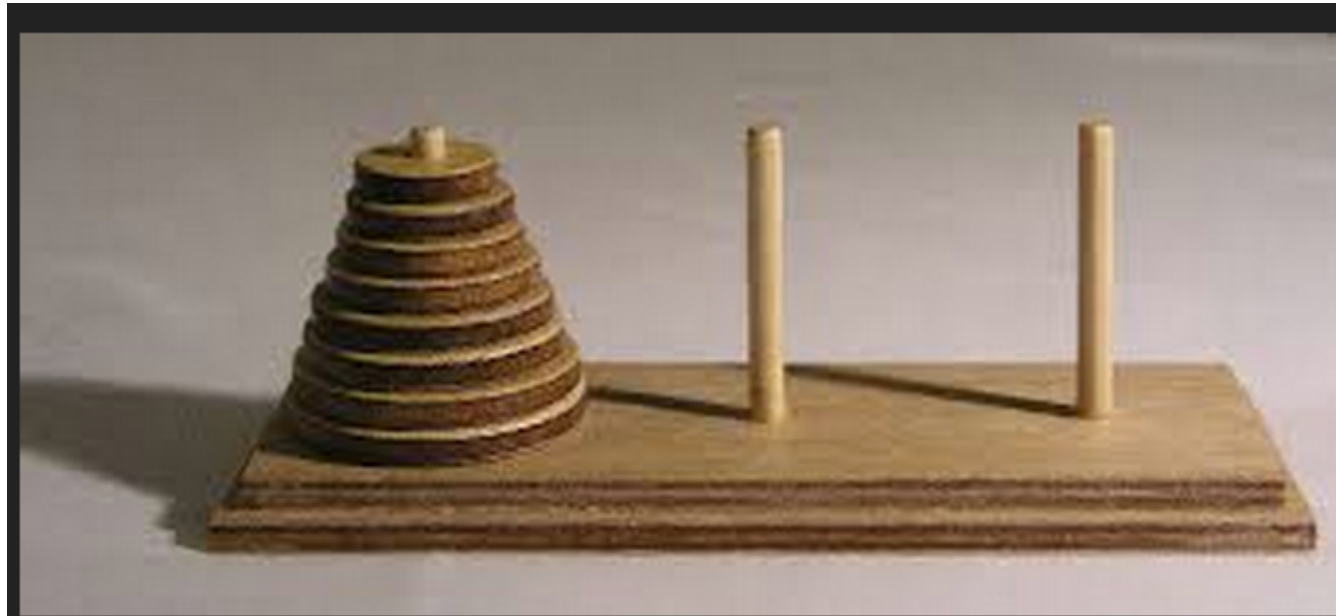

Function: recursion

- Demo: Tower of Hanoi.

source

middle

destination



Function: recursion

- Objective: to move the entire stack to another rod while obeying the following rules:
 1. Only one disk can be moved at a time.
 2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
 3. No larger disk may be placed on top of a smaller disk.

Function: recursion

- It is super simple: recurs the function for the $n-1$ disks
 1. Uncover the bottom disc by move the substack ($n-1$) above it from the **source** to the **middle** post.
 2. Move the bottom disc (n) from **source** to the **destination** post.
 3. Finally, move the substack ($n-1$) from **the middle** post to the **destination** post.

Function: recursion

- It is super simple:

```
function Hanoi(n, from, to , via)  
{  
  if (n==0) return;
```

```
  Hanoi(n-1, from, via , to);
```



Move n-1 from source
to middle tower via
destination

```
  moveDisk(from,to);
```



Move nth disk from
source to destination

```
  Hanoi(n-1, via, to , from);  
}
```



Move n-1 from middle
to destination

Function: Scope of Variables

- Variables declared inside the function only exists inside the function.

Local variables: declared inside the function.

Global variables: declared outside any function.

- Using “This”

```
var x = 13;
function scopeFun() {
    var x = 12;
    alert(x);      //value of 12
    alert(this.x); //value of 13
}
scopeFun();
```

Function Scope Example:

myCar.html : how passing a function change the variable's assigned value.

Quiz

At each of the step, yell out the value of variable “b”.

```
var foo = function(){  
  var a = 3; b = 5;
```

What is the value of b?

```
var bar = function(){  
  var b = 7, c = 11;
```

What is the value of b?

```
  a += b+c;
```

What is the value of b? What is the value of a?

```
};
```

```
bar();
```

What is the value of b? What is the value of a?

```
};
```

Exercise 2: maximum

- Write a function that return the maximum of two numbers.
- Call it like this:

```
alert("which is larger, 24 or 15?\nAh, "+ max(24,15) + " of course!");
```


Function: function calling function

```
Function myFunc(a){  
  //do something  
}
```

```
Function mySecondFunc (x) {  
  var y = myFunc(x);  
  return myFunc(y);  
}
```

```
alert("mySecondFunc(5));
```

Exercise 3

- Using the function you have written that compute maximum of two numbers. Write a script that compute the maximum of three numbers.
- Hint: to compute maximum of three numbers, you can compare the maximum of two numbers with the third one.

DOM objects

- Ways to interact with user:
- `Alert()`; e.g. `alert("this is my message!");`
alert doesn't not return a value
- `Confirm()`; returns true or false based on "OK" or "cancel".
- `Prompt()`; returns user input or null.

DOM method: getElementById

- Quickly access the value of an HTML element

```
<html>
<body>

<p id="intro">Hello World!</p>

<script>
var
txt=document.getElementById("intro").inne
rHTML;
document.write(txt);
</script>

</body>
</html>
```

DOM method: getElementById

- getElement byID is a function of the document object. Can only access using document.getElementById

The innerHTML property

- Each HTML element has an innerHTML property that defines both HTML code and the text between the opening and closing tag.
- By changing the element's innerHTML, you can make more interactive page.

innerHTML Demo

- Change the text based on user input

Other document methods

- getElementById
- getElementbyTagName
- getElementsByClassName

Demo: change text background

- ChangingBGColor.html

Exercise 4: `getDocumentById`

1. Write a couple of paragraphs onto the browser.
2. When user click a button, make the paragraphs to appear bold.

Exercise 5: Display random image

1. Download 5 images.
2. Display one of the image on the browser.
3. When the user click a button, the image change into any random image from the 5 images.

Homework & Readings

Sams teach you JavaScript in 24 hours, Hours 3,4,5,6.

JavaScript: The good parts, Chapter 4. Functions.

Code Academy, JavaScript (tracks 3-4).

<http://www.codecademy.com/tracks/javascript>