Fofanova N, CIS 236, UH Lecture for Week 13: JavaScript

JavaScript Introduction

JavaScript scripting Language

What is JavaScript?

JavaScript is Programming Language designed for web page. There is many languages like this, therefore, we need to specify in the code that we are using JavaScript by using TAGs

<script> type="text/javascript"> </script>

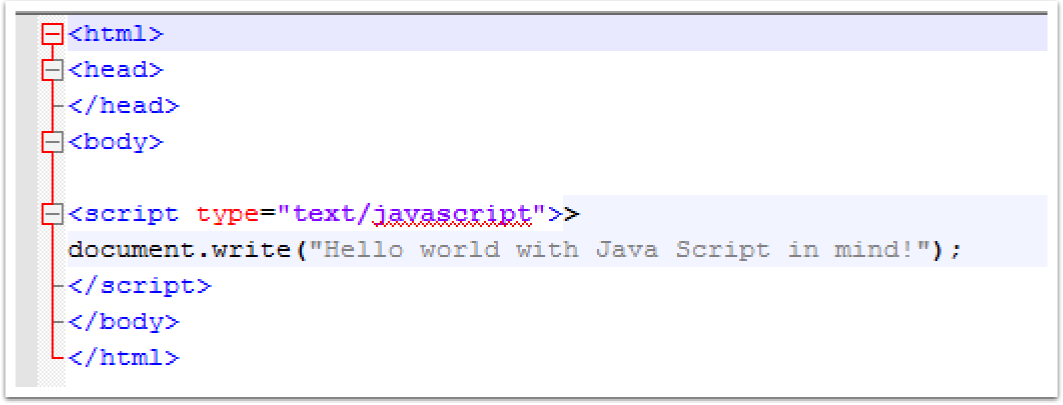
JavaScript runs in client software. Client-side scripting enhances functionality and appearance

- Makes pages more dynamic and interactive
- Pages can produce immediate response without contacting a server
- Customization is possible on the basis of users' explicit and implicit input
- Browser has to have a built-in (JavaScript) interpreter

Checking if JavaScrip set up in your Browser

Create simple program in NotePad++, save as .html and run it in IE, FireFox and all other browsers you have. If you will not see Hello message - you have old browser and need to download new one. For checking use ONLY super simple program. If you have wrong tag or even error in syntax - there will be no warning. Browser will show up as empty page.

Create JScriptPracticeFolder and transfer HelloWorld Program in it. Check if program running again



```
<html>
<head>
</head>
<body>

<script type="text/javascript">
document.write("Hello world with Java Script in mind!");
</script>
</body>
</html>
```

Comments and Statements

Browser will ignore comments. Comments needed for the same reason as you need them in other program - for easy readability of your code. There is way to write comments

Practice

1.// Comments goes here

2. /*

Comments goes here

Best for multiple line comments

*/

Statements (or instructions) are written one per line and ended by semicolon

document.write("Hello world with Java Script in mind!");
Keep in mind JavaScript seriously case sensitive.

Variables and their types.

var will declare variable. For example x=17;
Variables uses letters, numbers and underscores only. Variable name cannot start with number and have spaces

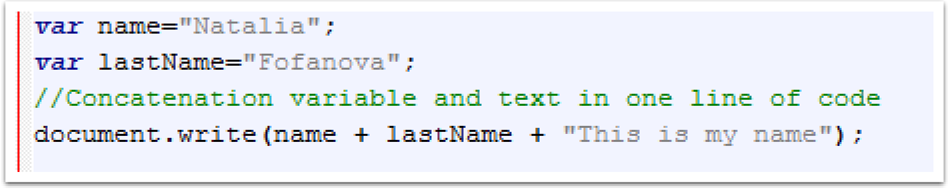
JavaScript can handle whole numbers, decimal, negative

Text should be included to quotation marks. Example "Love to learn!", when variable is not.

You can concatenate variable and text in one printout line

Practice: Hello I am (your first and last name) taking CIS 2336 and demonstrating Java Scrips by running simple programm.

Make first and last name as variables



```
var name="Natalia";
var lastName="Fofanova";
//Concatenation variable and text in one line of code
document.write(name + lastName + "This is my name");
```

Functions (mini program). Average function

Function with no parameters listed
Function could be written by you or someone else and exist as separate block of code.

function nameOfFunction(){
body of function as one or multiple statements separated by semicolon} //Note: no semicolon after curly braces

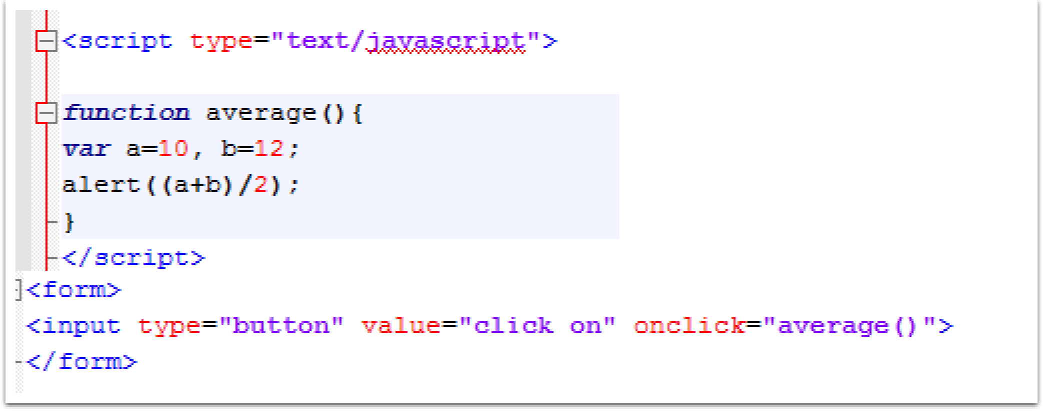
To use function, you need to call it by name.

So lets create function Average with 2 variables a=10 and b=12. You will need new special word ***alert***
"alert" will create a message box with output.
Then create Form - use HTML code outside of JavaScript

Use "button" as ***input***, create ***value*** (text you want to see on button); use action "***onclick***"

Run and enjoy results

Class assignment: Do not use form, just call function by name. You still have to get result in message box.



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

function average(){
var a=10, b=12;
alert((a+b)/2);
}
</script>
<form>
<input type="button" value="click on" onclick="average()">
</form>
```

Function with parameters: single parameter and multiple parameters

Function with one parameter

Parameters of function is additional information for function. Example below will display text:

```
function name(n)
    {
      alert("My name is " + n);
    }
name("Natalia");
```

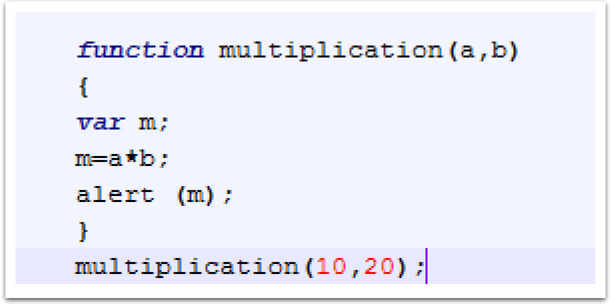
If you pass another name and then other and so on, program will complete all line of codes one by one.

```
name("Fofanova");
name("N Fofanova");
```

Function with more than one parameter.

Multiplication of two numbers: Make sure if you have two parameters in initial function, you call function and have to list two values in parentheses.

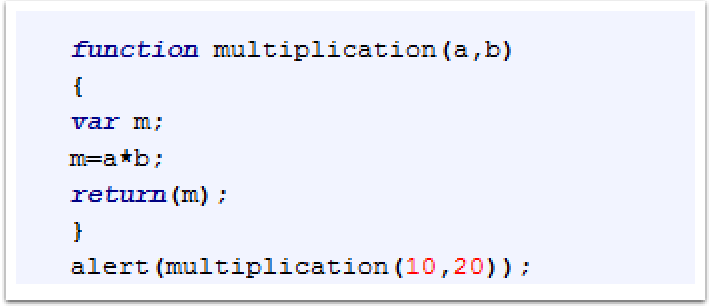
You can call functions with different parameters (values) as many times as you want - it would be still same function and you do not have to rewrite code over and over. That is advantage of using functions.



```
function multiplication(a,b)
{
var m;
m=a*b;
alert (m);
}
multiplication(10,20);
```

Returning value from Function

We are going to use ***return*** keyword for returning value from functions. Returning value, means complete calculation or other action inside the function and make it available for future use.
Lets rewrite multiplication function, using ***return*** statement



```
function multiplication(a,b)
{
var m;
m=a*b;
return(m);
}
alert(multiplication(10,20));
```

Assignment Operators

Increment by 1
a=a+1; is same as a++;

Decrement by 1
a=a-1; is same as a--;

Assignment Operator
a+=b; is same as a=a+b. Same with all other math signs: minus (-), multiply (*), divide (/)

Example:

var a=12;

a+=15, means a=12+15

Decision statement: Basic IF Statement

Basic IF statement: we are checking conditions. If conditions is true or match, we are completing following code. If not we are going to end of IF statement (nothing going to happens). Note: equal (compare) sign in conditions use (==) not (=); != means not equal.
General form:

```
if (condition)
{
complete statement/s;
 }
```

Complex IF statement: You can use IF as many time as you need it in the program.
Advanced way to check more than one conditions

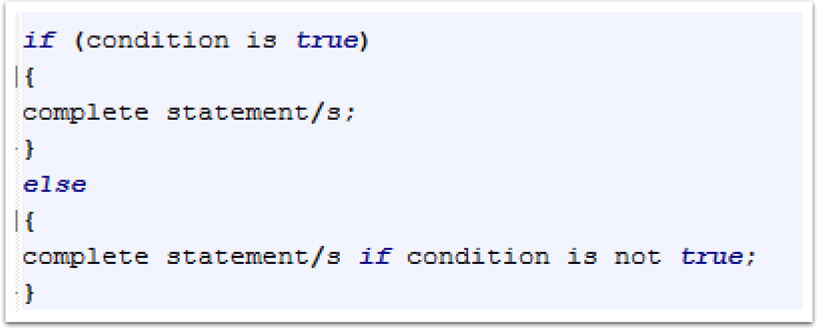
- **if((first condition)&&(second condition)){}.** If both condition is true, complete the code; if not go to the end. If one condition is false - go to the end

- **if((first condition)||(second condition)){}.** If one or another condition is true - complete the following code. If both true - complete the code. If both false - go to the end

Decision statement: IF/ELSE statement

We are checking conditions: if condition is true, complete following to IF statement. If not check other conditions or complete statements inside ELSE statement

General form

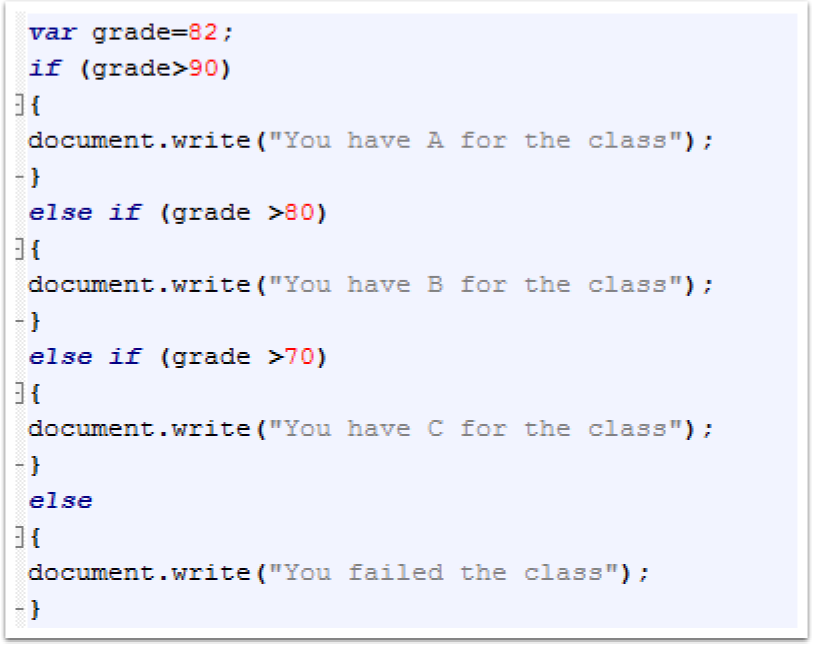


```
if (condition is true)
{
complete statement/s;
}
else
{
complete statement/s if condition is not true;
}
```

Nesting IF statements. Nesting IF/ELSE statements

Nesting means IF statement inside another IF statement

Below syntax for simple IF/ELSE statement

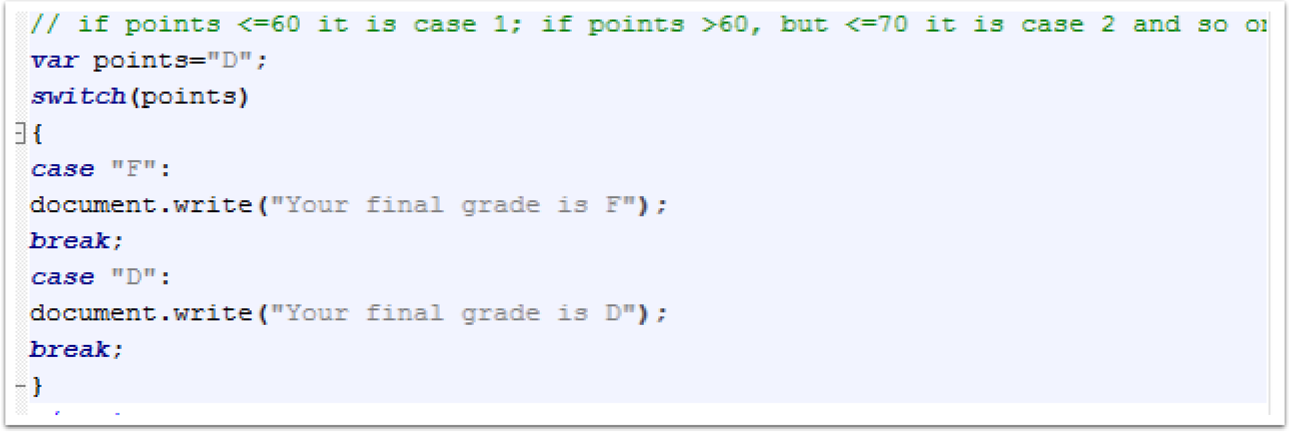


```
var grade=82;
if (grade>90)
{
document.write("You have A for the class");
}
else if (grade >80)
{
document.write("You have B for the class");
}
else if (grade >70)
{
document.write("You have C for the class");
}
else
{
document.write("You failed the class");
}
```

Switch

What if we need to create the program that need to make a choice during execution. For example assign letter grade according number of points: 0-60 -F; 60.1-70 -D; 70.1-80 C; 80.1-90 B; 90.1-100 A. Definitely, you can use nested IF/ELSE statement. But there is easier way: SWITCH

SWITCH statement take one variable and check for all possible scenarios, which we will be calling CASE. Break needed if correct case found - there is no point to test all other conditions.



```
// if points <=60 it is case 1; if points >60, but <=70 it is case 2 and so o
var points="D";
switch(points)
{
case "F":
document.write("Your final grade is F");
break;
case "D":
document.write("Your final grade is D");
break;
}
```

Take home assignment: Calling function from another subprogram or function

Complete at home all JavaScript on codecademy.com (entire 100%)

JavaScript from http://www.w3schools.com/js/default.asp is optional, but can help to complete Quiz in the class
Homework: lets solve quadratic equation ax^2+bx+c=0. If you do not remember formula, Google it

Do not Google solving quadratic equation, because I need you to complete it my way.

2x^2+4x-4=0. Find x1 and x2

1. Create function Discriminant

2. Create function unknownX1, return X1. Call Discriminant function from here

3. Create function unknownX2, return X2. Call Discriminant function from here

4. Display results by calling functions from step 2 and 3