

LBSC 690 Session #10  
Programming, JavaScript

Jimmy Lin  
The iSchool  
University of Maryland

Wednesday, November 5, 2008



This work is licensed under a Creative Commons Attribution-NonCommercial-Share Alike 3.0 United States license. See <http://creativecommons.org/licenses/by-nc-sa/3.0/us/> for details

## Take-Away Messages

- o Programming is a lot like cooking
- o Although after this session, you may never want to do it again...



## You will learn about...

- o Different types of programming languages
- o Basic programming constructs
- o Controlling execution of instructions



Source: Wikipedia



Source: Wikipedia

Software  
Hardware  
"It's just sand!"

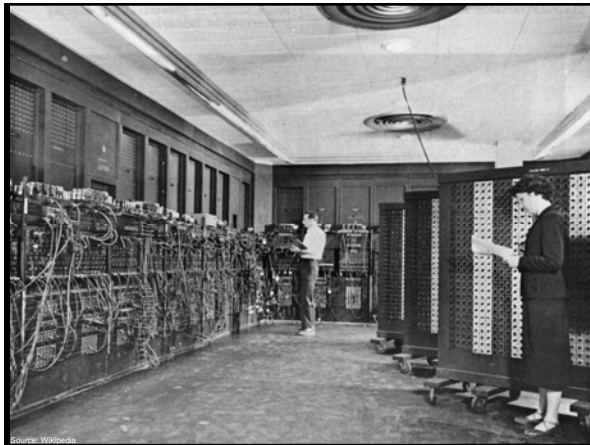
## Types of Software

- Application programs (e.g., PowerPoint)
  - What you normally think of as a "software"
- Operating system (e.g., Windows XP)
  - Software that manages your computing resources
- Compilers and interpreters
  - Software used to write other software
- Embedded software (e.g., TiVO)
  - Programs permanently embedded inside some physical device

The iSchool  
University of Maryland



# Programming



Source: Wikipedia

## Programming Languages

- Software "does something"
  - Instructions for telling the machine "what to do" are expressed in a programming language
- Special purpose: geared towards specific tasks
  - Spreadsheets (e.g., Excel)
  - Databases (e.g., SQL)
  - Complex math (e.g., Matlab)
- General purpose: able to accomplish anything
  - Examples: Java, JavaScript, C, C++, Perl, Python ...

The iSchool  
University of Maryland



## Types of Programming

- Low-level languages
  - Directly specifies actions of the machine
  - Example: assembly language
- High-level languages
  - Specifies machine instructions at a more abstract level
  - Compiler/interpreter translates instructions into machine actions
  - Example: JavaScript

The iSchool  
University of Maryland



# Programming

... it's a lot like cooking



Source: Iron Chef America

Ingredients  
Containers  
Instructions

Data types  
Variables  
Instructions

## Constructs for controlling execution of instructions

Sequence  
Condition  
Repetition

## Data Types and Variables

- Data types = things that you can operate on
  - Boolean: true, false
  - Number: 5, 9, 3.1415926
  - String: "Hello World"
- Variables hold values of a particular data type
  - Represented as symbols (e.g.,  $x$ )
- In JavaScript, *var* declares a variable
  - `var b = true;`      create a boolean  $b$  and set it to true
  - `var n = 1;`            create a number  $n$  and set it to 1
  - `var s = "hello";`      create a string  $s$  and set it to "hello"

## Instructions

- Things that you can do:
  - `-x`                    reverse the sign of  $x$  (negation)
  - `6+5`                    Add 6 and 5 (numeric)
  - `"Hello" + "World"`    Concatenate two strings
  - `2.1 * 3`                Multiply two values
- Storing results:
  - `x = 5`                    set the value of  $x$  to be 5
  - `x += y`                   $x = x + y$
  - `x *= 5`                   $x = x * 5$
  - `x++`                    increase value of  $x$  by 1
- In JavaScript, all instructions end with a semicolon (;)

## Controlling Execution

- Sequence
- Condition
- Repetition

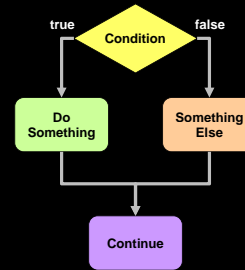
## Sequence



```
var a = 2;  
var b = 3;  
var c = a * b;
```



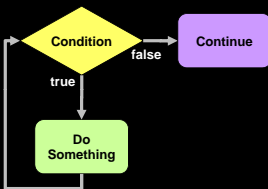
## Condition



```
if (gender == "male") {  
  greeting = "Hello, Sir";  
} else {  
  greeting = "Hello, Madam";  
}
```



## Repetition



```
n = 1;  
while (n <= 10) {  
  document.writeln(n);  
  n++;  
}  
  
for (n = 1; n <= 10; n++) {  
  document.writeln(n);  
}
```



## Test Conditions

- o `x == y` true if `x` and `y` are equal
- o `x != y` true if `x` and `y` are not equal
- o `x > y` true if `x` is greater than `y`
- o `x <= y` true if `x` is smaller than or equal to `y`
- o `x && y` true if both `x` and `y` are true
- o `x || y` true if either `x` or `y` is true
- o `!x` true if `x` is false



## Arrays

- o A set of elements grouped together
  - For example, the number of days in each month
- o Each element is assigned an index
  - A number is used to refer to that element
  - For example, `x[4]` is the fifth element (count from zero!)
  - Arrays and repetitions work naturally together



## Functions

- o Reusable code for doing a single task
- o A function takes in one or more parameters and returns one value

```
function convertToCelsius(f) {  
  var celsius = 5/9 * (f-32);  
  return celsius;  
}
```

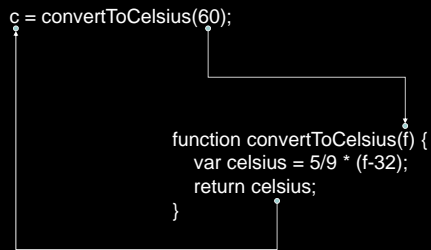
```
function weirdAddition(a, b) {  
  var result = a + b - 0.5;  
  return result;  
}
```



## Calling Functions

- When you "call" a function, you invoke the set of instructions it represents

```
c = convertToCelsius(60);  
  
function convertToCelsius(f) {  
  var celsius = 5/9 * (f-32);  
  return celsius;  
}
```



The iSchool  
University of Maryland



## More Examples

```
var f = 60;  
c = convertToCelsius(f);  
  
r = weirdAddition(2, 4);  
  
var a = 2;  
var b = 3;  
r = weirdAddition(a, b);
```

The iSchool  
University of Maryland



## Algorithms

- Derived from the name of the Persian mathematician Al-Khwarizmi
- A sequence of well-defined instructions designed to accomplish a certain task

The iSchool  
University of Maryland



## Programming for the Web

- Common Gateway Interface (CGI) [Server-side]
  - User inputs information into a form
  - Form values passed to the server via CGI
  - Program on the server generates a Web page as a response
- JavaScript [Client-side]
  - Human-readable "source code" sent to the browser
  - Web browser runs the program

The iSchool  
University of Maryland



## Where is the JavaScript?

- JavaScript is usually kept in the <head> section of an HTML document

```
...  
<head>  
<script language="JavaScript" type="text/javascript">  
<!--  
function calculate() {  
  var num = eval(document.input.number.value);  
  ...  
  document.output.number.value = total;  
}  
//-->  
</script>  
</head>  
...
```

The iSchool  
University of Maryland



## Handling Events

- When does code actually get executed?
- Events:
  - User actions trigger "events"
  - Embedded in all modern GUIs
- Event handlers are used to respond to events
  - Examples of event handlers in JavaScript
    - onMouseover: the mouse moved over an object
    - onMouseout: the mouse moved off an object
    - onClick: the user clicked on an object

The iSchool  
University of Maryland



## Input and Output

- How do you get information to/from the user?
  - Forms provide a method for accepting input and displaying output

### In HTML

```
<form name="input" action="">
Please enter a number:
<input size="10" value=" " name="number"/>
</form>
<form name="output" action="">
The sum of all numbers up to the number above is
<input size="10" value=" " name="number" readonly="true"/>
</form>
```

### JavaScript code

```
var num = eval(document.input.number.value);
document.output.number.value = 10;
```

Reads in a value  
eval function turns it into a number

Changes the value in the textbox

The iSchool  
University of Maryland



## JavaScript Resources

- Google "javascript"
  - Tutorials: to learn to write programs
  - Code: to do things you want to do ("borrow")
- Books

The iSchool  
University of Maryland



## Programming Tips

- Details are everything!
  - Careful where you place that comma, semi-colon, etc.
- Write a little bit of code at a time
  - Add a small new functionality, make sure it works, then move on
  - Don't try to write a large program all at once
- Debug by outputting the state of the program
  - Print out the value of variables using document.write
  - Is the value what you expected?

The iSchool  
University of Maryland



## You have learned about...

- Different types of programming languages
- Basic programming constructs
- Controlling execution of instructions

The iSchool  
University of Maryland

