

JavaScript

May 18, 2007
CPSC 101
KangKang Yin

Why JavaScript

- ▶ Interpreted scripting language
- ▶ Default interface: Web browsers
- ▶ Loosely typed
 - primitive: number, string, Boolean
 - composite/collection: object, array
- ▶ Make static web content interactive
 - have access to the content of HTML documents
 - change the content
 - respond to user input

Why JavaScript

- ▶ Demos
- ▶ Myth
- ▶ Project Idea
- ▶ How to learn a programming language
 - learn the basics
 - program
 - debug
 - keep a reference book around

Declaring Variables

- ▶ Number `var price=10.0;`
- ▶ String `var hairColor="black";`
- ▶ Boolean `var indexFound=false;`
- ▶ Object `var point={
 x:0.0,
 y:0.0
 };`
- ▶ Array `var primeNumbers=new Array(2,3,5);`
`var lookup_table=new Array(1024);`

Operators

- ▶ Arithmetic `%`
- ▶ Relational `!=`
- ▶ Logical `&&` `||` `!`
- ▶ Bitwise `&` `|`

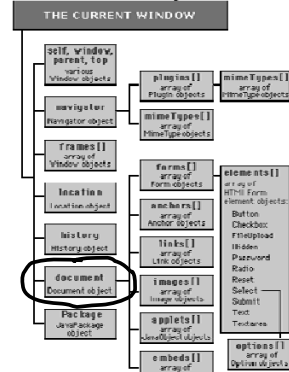
Statement

- ▶ `if(expression) statement1 else statement2`
- ▶ `if() {} else if() {} else {}`
- ▶ `while(expression) statement`
- ▶ `for(initialize; test; increment) statement`
- ▶ `break`
- ▶ `return expression`

Function

```
function <name> (<parameter list>)
{
    <statements>
    ....
    return <expression>
}
```

Client-Side Hierarchy and the DOM



Event-Driven Programming Model

batch mode =>

limited interactivity =>

dynamic, interactive: made possible by graphical display, GUI, pointing devices

event-driven: web browser notifies programs of user input by generating *events*.

event handler: function to respond to the event

Event-Driven Programming Model

	computational	event-driven
code	single, monolithic block	A number of event handlers
flow of control	static, from beginning to end	unpredictable, asynchronous
invocation	user	user input triggers, system invokes handlers