Javascript Description

- Javascript is a programming language. You write programs in it to make the computer do something you want
  - Javascript is widely used to program web pages
- Javascript code (embedded or in external files) is a sequence of statements
- Statements come in two forms
  - Function definitions
  - Ordinary statements

Javascript Function definitions

- Function definitions
  - Functions are identified by the keyword function:
  - function MyClick() { //some code… }

Javascript Statements

- Functions contain ordinary statements, including
  - Variable definitions
  - Operations on variables
  - Conditional statements
  - Loops
  - Calls to other functions

Variable definition

- Variable definitions look like
  - Var x;
- You must have a ‘;’ at the end of the variable definition

Operations on Variables

Mathematic

- Z = x + y;
- Z = x - y
- Z = x * y;
- Z = x / y;
- Z = x % y; //remainder when x is divided by y
- ++z; //add one to z

- String concatenation – tie two strings together
  - S = “my hovercraft “ + “is full of eels!”
- Function invocation
  - Z = Math.floor(Math.random() * 11); // z is a random number between 0 and 10

Conditional Statements

- Conditional statements look like:
  - If (*condition) { statement;…}*
    else { *statement; }*
- *Condition* is some ‘test’ on a variable or value
  - If (x > 3)
  - If (y == x)
  - If (y != x)
  - …

Loops

- Loops are sets of statements that run repeatedly until ‘something’ makes the loop stop running
- Two types of loops
  - **for** loops
  - **while** loops
  - **Do-while** loop

for loops

- **for** loop
  - **for** (*expression-1; condition; expression-2) {statement-1…statement-n}*
    - *Expression-1* is executed.
    - *Condition* is tested.
    - If *condition* is true, statements-1…n are executed
    - *Expression-2* is executed, and we loop again (starting with testing *condition*)
- Example:
  - For (var I = 0; I < 10; i++) { … } //does something 10 times

While loops

- **while** loops
  - while (*condition); { statement-1; statement-2; … }*
  - Loop repeats until condition is false:
    - While (x < 3) { x = x + 1; }
- **Do-while loops**
  - **Do {** *statement-1…statement-n}* **while** (*expression);*
  - Loop runs once, then evaluates *expression*.
  - If *expression* is true, the loop runs again, *expression* is evaluated and this ‘loop’ repeats, until *expression* is false
    - Example:
      - Var I = 0;
      - **do** { I = I + 1; } **while** (I < 10);