

# Javascript continued

## Review types & conversions

- **each variable holds information of a specific type**
  - really means that bits are to be interpreted as info of that type
    - 3 integer
    - 3.00 floating point - decimal point
    - "3.00" character string - quotes
- **Javascript usually infers types from context, does conversions automatically**
  - "Sum = " + sum
  - sum \* sum
- **Special value null**
  - Compare "" - character string of length 0 NOT SAME AS null
  - Cancel received from prompt yields null
  - OK with no input from prompt yields ""

## Review control statements

- **If (condition is true) { do these statements} else {do these statements instead}**
- **While (condition is true) {do these statements}**
- **for (value = startvalue; value <= lastvalue; value++) {do these statements} Foxtrot cartoon**
  - value is an integer variable
  - value++ means add one to value
  - Equivalent to "counting" (index) loop with WHILE last time
- **Function call and return**
- 

## Data structures and "complex types"

- **Arrays in Javascript**

### pop up a level of abstraction:

- **arrays - get at by "index" - number - i<sup>th</sup> item**
- **sets - get at by name - like database record**
- **Critical part of algorithm design**

## A working sort example

```
var name, i = 0, j, temp
var names = new Array()

// fill the array with names
name = prompt("Enter new name, or OK to end")
while (name != "") {
    names[names.length] = name
    name = prompt("Enter new name, or OK to end")
}
// insertion sort
for (i = 0; i < names.length-1; i++) {
    for (j = i+1; j < names.length; j++) {
        if (names[i] > names[j]) {
            temp = names[i]
            names[i] = names[j]
            names[j] = temp
        }
    }
}
// print names
for (i = 0; i < names.length; i++) {
    document.write("<br> " + names[i])
}
```