Variables, Conditionals, and Functions

DMS 102: Programming Digital Media

Lecture 3

Variables

- "A hole in computer memory" …with a name
- Objects, lists, other variables
- Changeable ("variable")

JavaScript variables

- keyword: `var` followed by a name
- upper or lower case letters, numbers
- no spaces
- first character must not be a number
- must not be a *keyword*

examples:

```
var year;
var firstName;
```

Note:

$$\texttt{first}\underset{\uparrow}{\texttt{N}}\texttt{ame} \neq \texttt{first}\underset{\uparrow}{\texttt{n}}\texttt{ame}$$

The Assignment Operator

- Variables, typically used with the *assignment operator*

```
var firstName = "Bubs";
```

- Whatever on the right → whatever on the left
- "=" …does NOT mean equals!
- This is okay…

```
var x = 1;
x = x + 1;
```

…results in `x` stores the number 2

Data Types

- Variables:
  - integers
  - floating point number
  - a character or a string
  - a Boolean
  - others
- Strongly-typed vs Weakly-typed

```
var myVariable;
myVariable = 200;
myVariable = 12.1;
myVariable = "Hello World!";
myVariable = true;
alert(myVariable);
```

Operators

- assignment: `=`
- concatenation: `+`
- arithmetic: `+  -  *  /`
- order of operations:
  - `( )` evaluated first
  - `*` and `/` left to right
  - `+` and `-` left to right
- shorthand:

```
+=      ++
-=      --
*=
/=
```

```
var a = 100;
var b = 50;
var result = a + b;
```

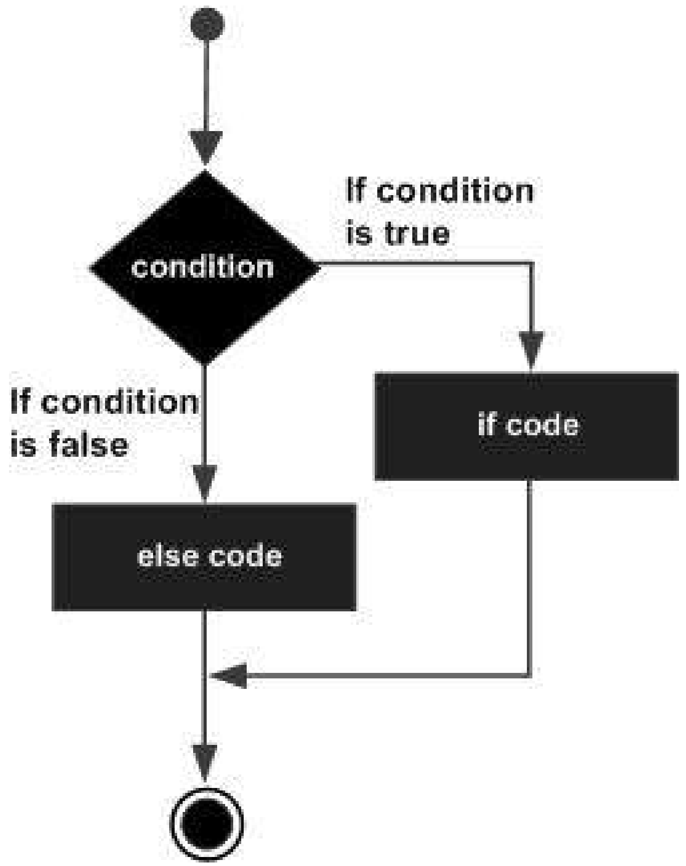
```
score = score + 1;
score += 1;
score++;
```

Conditionals - the IF statement

- ALL programming languages: conditions (e.g. "if")
- Condition examples:

```
bankBalance < 100
age > 18
```

```
if ( condition ) {
      // JavaScript statements go here...
}
```



Conditions

```
if ( condition ) {
    // JavaScript statements go here...
}
```

- Any condition MUST evaluate as either TRUE or FALSE
- Comparison operators:

==	…is equal to	===	…is strictly equal to	!=	…is not equal to
>	…is greater than	>=	…is greater than or equal to		
<	…is less than	<=	…is less than or equal to		

Complex Conditions

- You can use logical operators AND and OR
  - And: && (double ampersand)
  - Or: || (double pipe)

```
if ( (a == b) && (c == d) ) {...
```

a is equal to *b* AND *c* is equal to *d*

…both must be true for the overall condition to be true

```
if ( a == b || c == d ) {...
```

a is equal to *b* OR *c* is equal to *d*

…either one must be true for the overall condition to be true

Complex IF statements

- Two IF statements: suboptimal
- IF/ELSE

```
if ( condition ) {
    // JavaScript statements go here...
} else {
    // Alternative JavaScript statements here...
}
```

Conditionals - Switch Statements

- A different kind of IF: "switch"
- Uses these keywords:

  `switch` ...instead of IF

  `case` ...potential values

  `break` ...prevent "fall through"

  `default` ...catch-all

```
var price;
var grade = "Premium";
switch ( grade ) {
  case "Regular" :
    price = 3.15;
    break;
  case "Premium" :
    price = 3.35;
    break;
  case "Diesel" :
    price = 3.47;
    break;
  default: alert("That's not
    a valid grade");
}
```

Conditionals - the WHILE statement

- Just like the IF statement, except re-checks the condition; did it change?
- Something needs to change the condition - typical: an *incrementer*

something to do with `i`
e.g. `( i < 10 )`

```
var i = 0;
while ( condition ) {
    // JavaScript statements go here...
    i++;
}
```

Functions

- A block of code
- Unique name, verb/object (typically)
- Parenthesis for passing data
- "Call" the function by its name
  - Note: doesn't run unless called
- Functions can contain other functions
- JS: functions and calls can come in any order (in one file)

```
function calculateScore() {
   // JavaScript statements go here...

}


// "call" the function...
calculateScore();
```