Fundamentals of Web Programming

Lecture 2: flow of control

Devin Balkcom
devin@cs.dartmouth.edu
office: Sudikoff 206

http://www.cs.dartmouth.edu/~fwp
Variables store data

Declaration:

```javascript
var temperature;
```

Assignment operator:

```javascript
temperature = 42;
```

"Copy value on right into variable on left."

**will not work:**

```javascript
42 = temperature;
```

Combined declaration and assignment:

```javascript
var temperature = 42;
```
= does not mean equals!

Bad math (statements of fact):

\[ \begin{align*}
    x &= 0 \\
    x &= 1
\end{align*} \]

Legal Javascript (instructions):

\[
\begin{align*}
    x &= 0; \\
    x &= 1; \\
    \text{print}(x);
\end{align*}
\]

"Assign the value 1 to variable x", not "x equals 1".
Values of variables affect code behavior

```
1 // coordinates of the center of the face:
2 var x = 100;
3 var y = 100;
4
5 // draw the outline of the face
6 fill(255, 255, 0); // set fill color to yellow
7 ellipse(x, y, 100, 100);
8
9 // draw the eyes
10 fill(0, 0, 0); // set fill color to black
11 ellipse(x - 20, y - 10, 10, 10);
12 ellipse(x + 20, y - 10, 10, 10);
13
14 // draw the mouth
15 fill(255, 255, 0); // yellow
16 arc(x, y + 10, 50, 40, 0, PI);
```

You can use a **variable** anywhere a value is needed.
Types of values

Number type:

```javascript
meaningOfLife = 42;

// number of molecules in an avocado
avocadoNumber = 6.02e23;

weightOfAntInGrams = .004;
```

String type (quotes indicate string value rather than variable):

```javascript
lukesFather = "Darth";
```
Types of values

Boolean type:

iLoveFWP = True;

Function type:

```javascript
var sayHello = function() {
    print( "Hello!" );
};
var s = sayHello;
s();
sayHello();
```
Expressions

You can use an **expression** anywhere a value is required.

- Number operators: +, -, /, *, %
- String operator: +
- Conditional operators: >, <, >=, <=, ===, !=
- Boolean operators: &&, ||, !

---

**Exercise**: computing with conditional and boolean operators
Exercise: Computing with conditional and boolean operators
while-loops

The Shining (3/7) Movie CLIP - All Work and No Play (1980) HD

```javascript
var condition = function() {
};
print("All work and no play makes Jack a dull boy.");
print("All work and no play makes Jack a dull boy.");
```
while-loops: repeated patterns

```javascript
1   var robotSim = function() { ()
63  var condition = function() { }
79
80  drive(100);
81  turnRight();
82
83  drive(100);
84  turnRight();
85
86  drive(100);
87  turnRight();
88
89  drive(100);
90  turnRight();
```
while-loops: how they work

1. Program counter reaches while loop header
2. If condition is false, skip the body
3. If condition is true, execute body and go back to header

Jack example, with `false` vs. `true` conditions:

We need a condition that is initially true, but becomes false.
Counter loops

What is the same about these lines of code? What is different?

```c
1  print( 1 );
2  print( 2 );
3  print( 3 );
4  print( 4 );
5  print( 5 );
```

We want to use a variable to **unify** the lines of code.

---

Exercise: String art
Exercise: String art
Anatomy of a counter while-loop

There is a common pattern for while loops used for counting:

```javascript
var i = 1;
while ( i < 6 ) {
    print( i );
    i = i + 1;
}
```

1. Initialize a counter variable
2. Test a condition in the header
3. Modify the counter variable
for-loops

Could we write a counter loop with less code?

```javascript
var i = 1;
while ( i < 6 ) {
    print( i );
    i = i + 1;
}
```

A for-loop is shorthand for a type of while-loop:

```javascript
for ( var i = 1; i < 6; i = i + 1; ) {
    print( i );
}
```

Exercise: Robots and for loops
Exercise: Robots and for loops
if-statements

if-statements, like while-loops, check if a condition is true. If true, the body runs once.

Let's write a program to check if it's 32 degrees or colder, and if so, print out "It's freezing."
else statements

A else-statement can be paired with an if-statement to provide a default:

```javascript
var temperature = 35;
if (temperature <= 32) {
    print("It's freezing.");
} else {
    print("It's not so cold.");
}
```

A sequence of if-statements, possibly followed by an else, is called an if-ladder.
else if

You can combine else and if to get an "else if" statement

```javascript
var temperature = 35;
if (temperature <= 32) {
  print("It's freezing.");
} else {
  print("It's not so cold.");
}
```
Nesting while and if

Let's find the integer factors of a number. We'll use a strategy called generate and test.

1. **Generate** a candidate factor.
2. **Test** if that candidate is really an integer factor and output if so.