Lesson 3: Expressions, Statements, and Booleans

If you haven't already, unzip the folder for this class and drop it in your desktop (remember to cd into your desktop before trying to run files!)

The Expression

- Like a one-sided equation
  - Gives the computer (or us) something to evaluate
  - Evaluate = Compute
  - We already saw these last lesson
- Serves as the basic building block in most code structures - not useful by themselves!
- Often used as conditionals (more later)

```
2 + 2 * 65536

speed > 55.0

regularPrice * (1.0 - salePercentOff)
```

Arithmetic Expression

- Throwback to math class
  - Remember order of operations when evaluating/writing expressions!
  - Don't worry too much if you're fuzzy on the order of operations!
- If nothing else, parentheses first
- All inputs are numbers
- Just apply general math rules to find the answer!

```
3 * 3 +5
// 14

3 * (3+ 5)
// 24
```

Boolean Values

```
let positive = true;
let negative = false;
console.log(typeof positive);
// prints "boolean"
```

- Most basic data type in computer science
- Only two possible values:
  - True
  - False
- Translates to simple “on” or “off” switches in your computer
- Can be directly assigned through “true” or “false” keywords
- Can also be assigned through a conditional

Boolean Expressions

- Result will only ever be “True” or “False”
- For our purposes, most of the time we’ll be dealing with conditionals
  - 10> 5 // true
  - 5==4 //false
- Very important when we only want our code to do something “if” something is true

Task #1

- Open up main.js in Atom
- Read the instructions for Task #1
  - Guess what each expression evaluates to before logging the result to the console and checking
  - Be sure to pay attention to the type of expression.
- Take 6 minutes, then we'll go over the answers

Statements

- Statements are like equations
  - Need two sides, not just one
- Always end a statement with a ;, like a period in english
  - Shows the program where lines are separated
- Most commonly used to assign a value to a variable (see right)
- This example had a lot, so don't panic!

```
// at the Grocery

salesTaxRate = 0.06;

totalGroceries = 38.99;
salesTax = totalGroceries * salesTaxRate;

chargeToCard = totalGroceries + salesTax;
```

Multi-Line Statements

```
k = h * kph - (rest / 60);

kilometersCycled = numberOfHoursPedalled *
kilometersPerHour - (totalMinutesOfRest / 60);
```

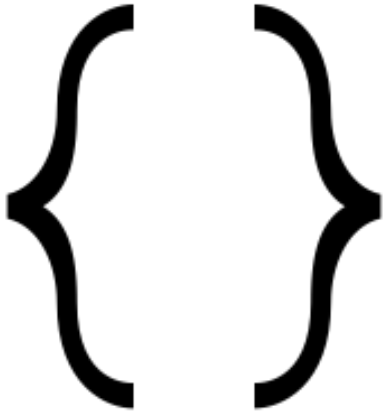
- Importance of semicolons!
- Line doesn't end until there's a semicolon
  - Javascript doesn't care how many physical lines the "line" goes on for
  - Always look for an ending semicolon and work backwards from there
  - Good code generally doesn't stretch onto multiple lines!
- Side note: These are all terrible variable names!

If Statements

- If statements are made up of two parts:
  - Condition - enclosed with ()
  - Block - enclosed with {}
- Think of the condition like a gateway
  - Always contains a boolean expression!
- The block is the meat of the structure
  - Made up of statements
  - Only executed if the conditional is true!

```
if (magePower > 120.0) {
    maxMagic = 500.0;
    lifeSpan = 800.0;
    maxWeapons = magePower / maxPowerPerWeapon;
}

// some more code
```



What is a block?

- Blocks can contain any type of code, but are useful because they section this code off
- This code can be reused when appropriate (which saves you time!)
- We'll go over all these in detail later on, but examples include
  - If/else statements
  - for/while loops
  - Functions and classes
- Keep track of your brackets!
  - It can be easy to forget which bracket lines pairs with another - very annoying!

```
$("#email3").change (function (e) {
    console.log ("email address: " + $("#email3").val());
    $.ajax({
      type: "GET",
      url: "/cryptex/_lookup/?email=" + $("#email3").val(),
      success: function (data) {
        if (data.totp == false || data.new == true) {
          $("#pass1").removeAttr ("style");
          $("#pass1").attr ("required", true);
          $("#pass1").focus ();
        } else {
          $("#api_code").removeAttr ("style");
          $("#api_code").attr ("required", true);
          $("#api_code").focus ();
        }
        $('#alert_placeholder').css('display','none');
        have_valid_email = true;
      },
      error: function (data) {
          makeAlert ("invalid email address");
          console.log ("invalid email entered");
      }
    });
});
```

Task #2

- Go to task #2 in main.js for this lesson
- Read the instructions carefully, and begin
- Shoot a message in the chat if you have a question.
- Take 15 minutes, we'll go over the solution afterwards.

Review

Today we learned:

- Expressions → how to evaluate them
- Boolean Values
- Statements → often comprised of expressions
  - Builds upon variable assignment
- If statements
  - Conditionals
  - Blocks

First Week... Done!

Next time:

- Arithmetic and Boolean Operators!
  - More ways to do math!
  - Manipulate our variables

Wrap Up - Enjoy the weekend!

Email questions to info.codedelaware@gmail.com

Next class: Monday 1/11