Intro to JS Cheat Sheet

Here are some notes on what's been covered in this chapter. Feel free to copy this information and extend it to make your own cheat sheet.

Pseudocode

- A way to plan out your script before coding it.
- It is a detailed, step-by-step description of what a computer must do, expressed in plain English rather than in a programming language.

Variables

- Variables are used to store data into the memory of the computer so that it can be referenced later.
- A variable is assigned a value using the `=` operator. First, the expression to the right of the `=` is evaluated. Then, this value is assigned to the variable to the left of the `=`.

```
var favoriteMovie = "Jaws";

favoriteMovie;
// => "Jaws"
```

- To use the value that a variable is storing, simply include that variable in an expression. An expression containing variables will evaluate just like one without variables, except that the variables will themselves be evaluated as part of the expression.
- When a variable is reassigned, it retains **no knowledge** of any prior values it may have held.

```
var winner = "Sarah";
winner = "Jeff";

winner;
// => "Jeff"
```

- A variable may be reassigned "in place" using an expression like `x = x + 1` (or its shorthand, `x += 1`).
- An expression like `x = y` only means that the value that `y` had been holding is now *also* held in `x`. **It does not imply any lasting relationship between x and y**.

Assignment Operators

- The `=` operator assigns a value to a variable.

- The `+=` operator adds a value to an existing variable.
- The `-=` operator subtracts a value from an existing variable.

Data Types

- Some common data types are numbers, strings, and Booleans.
- Strings should be surrounded by quotes: `"Hello"`.
- Numbers should *not* be surrounded by quotes: `7`.
- To check to see what type of data a variable holds, the `typeof`operator can be used.
    - Example:

```
var myPet = "Arthur";
typeof myPet;
// => "string"
```

Expressions

- An expression is a collection of values (like `1` and `2`) and operators (like `+` or `*`).
- The process of reducing this expression down to a single value is called **evaluation**.
- An operator takes in a number of inputs but outputs/evaluates to a single value.
- Common arithmetic operators include:
    - `+` - Addition
    - `-` - Subtraction
    - `*` - Multiplication
    - `/` - Division
    - `%` - Modulus
- Using the `+` operator with two strings literally puts them next to each other, instead of evaluating their total. This is called **string concatenation**.

Pseudocode

```
// Get the patron's age

// If age is greater than or equal to 65
    // Display message "Ticket price: $6.00"

// Otherwise if age is less than or equal to 25
    // Display message "Ticket price: $8.00"

// Otherwise
    // Display message "Ticket price: $10.00"
```

Data Types and Variables

1. Create a variable `petName`. Assign (give) it the value `"Rover"`.

   ```
   var petName = "Rover";
   ```

2. Create a variable `age`. Assign it the value `8`.

   ```
   var age = 8;
   ```

3. Create a variable `favoriteToy`. Assign it the value `"ball"`.

   ```
   var favoriteToy = "ball";
   ```

4. Hit the "run" button in the "Console" panel and then check the values of the three variables you created by typing each variable name into the "Console" tab and hitting enter/return.

   ```
   petName;
   age;
   favoriteToy;
   ```

5. Update `petName`. The new value should be `"Arthur"`.

   ```
   petName = "Arthur";
   ```

6. Update `age`. The new age should be `5`.

```
age = 5;
```

7. Update `favoriteToy`. The new favorite toy should be `"yarn"`.

```
favoriteToy = "yarn";
```

8. Hit the "Run" button in the "Console" panel and then check the values of the three variables you created by typing each variable name into the "Console" tab and hitting enter/return.

```
petName;
age;
favoriteToy;
```

9. In the "Console" panel, use the `typeof` command to find the type of data stored in each variable.

```
typeof petName;
typeof age;
typeof favoriteToy;
```

Expressions and Evaluations

1. In the "JavaScript" panel, declare (create) a variable `myNumber`. Assign it the value `30`.

```
var myNumber = 30;
myNumber; // Check the value of myNumber in the "Console" panel
```

2. Reassign (update) the `myNumber` variable to `20`.

```
myNumber = 20;
myNumber; // Check the value of myNumber in the "Console" panel
```

3. Use the `+=` operator to add `5` to the current value of `myNumber`.

```
var myNumber += 5;
myNumber; // Check the value of myNumber in the "Console" panel
```

4. Now create a second variable `greeting` and assign (give) it the value `"Hello "`.

```
var greeting = "Hello ";
```

```
greeting; // Check the value of greeting in the "Console" panel
```

5. Create a third variable `name` and assign it the value `"Margaret"`.

```
var name = "Margaret";
name; // Check the value of name in the "Console" panel
```

6. Create a fourth variable `sayHello`. The final value of the variable should be `"Hello Margaret"`. Use the variables `greeting` and `name`along with the `+` to create this value (string concatenation).

```
var sayHello = greeting + " " + name;
sayHello; // Check the value of sayHello in the "Console" panel
```