Spartie Syntax Overview

Basics

- The language is interpreted
- The language is dynamically typed (we will get into this later)
- There is automatic garbage collection, so we don't have to worry about memory allocation or release (we will use the JVM)

Identifiers

- Identifiers for variables and functions can be any mix of letters, but not include numbers
- For example, the following are valid identifiers:
    - `count`
    - `averageScore`
    - `HELLOWORLD`
- The following are not valid:
    - `count1`
    - `1count`
    - `count_one`

Variables

- Variables are declared using the var keyword:

```
var x = 5;
var greeting = "hello";
```

Types and Possible Values

- The types are pretty simple: boolean, string, and number
- A floating point number needs to have a number before the decimal point. For example, `.2` would be invalid, but `0.2` would be invalid.
- If there is no value, the value is `null`

Type	Possible Values
Boolean	true, false

String	"Hello!", "3.14"
Number	3.14, 30

Expressions

- Simple arithmetic expressions are supported:

```
x + y
x - y
x / y
x * y
-x
```

- Parenthesis are supported:

```
(x + y) * y
```

- Comparison is supported:

```
x < y
x <= y
x > y
x >= y
x == y
x != y
```

Logical Operators

- You can use `!` as a not operator as well as `&` for and `|` for or, respectively
  - Please note, there is no bitwise **and** or **or**

Statement

- Like C and Java, each statement ends with a `;`

Built Ins

- The `print` statement is built in. It is not a function that you create. Below is an example:

```
print "Hello CSDS 345!";
```

Control

- If statements, while statements, and for are very much like C. Except, we do not support else if.

```
if (i > 5) {
  // Do something
}
else {
  // Do something else
}

while (i < 10) {
  // Do something
}

for(var i = 0; i < 10; i = i + 1) {
  // Do something
}
```

Functions

- Like Kotlin, functions are declared using fun!
- But, because it is dynamically typed, we don't declare the return type or parameter types

```
fun storeGrade(score, name) {
  // Do something
}
```

Strings

- Strings use double quotes: `"string"`
- Strings exist on a single line

```
var someString = "Stay on one line";
```

Comments

- Comments use two forward slashes `//`