CS3320: Compilers - I

Mini-Assignment #2 - Lexical Conventions and Grammars of Languages: C vs. Python vs. Javascript

# Soumi Chakraborty

ES19BTECH11017

# 1 Lexical Aspects of C, Python, and Javascript

These three languages have many similarities in terms of their lexical aspects, such as the concept of tokens, identifiers, types of constants, etc. This short note aims to draw out the similarities and differences between the three languages.

## 1.1 Comparison between C and Python

C and Python are very similar in terms of their lexical rules about identifiers. The naming conventions of the identifiers are governed by the same rules in both the languages. The only difference is that in Python, starting or ending an identifier with underscores might have special meaning.

The two languages have both similarities and differences between their constants/literals.

Similarities:

* The same character escape codes/constants
* Have integers and floating points
* Allow octal/decimal numeric types

Differences:

* Python deals with string literals differently than C does (unicode)
  + Moreover, Python doesn’t differentiate between “ and ‘, whereas C does
* Python has provisions for complex numbers in their literals
* C has enums

C has provisions for both single-line and multiline comments, whereas Python officially only allows single-line comments. Since Python ignores string literals that don’t occur right after a function definition, etc., the ””” operator can be used for multi-line comments.

Decorators are a feature that’s unique to Python and allow you to wrap a function with something else to modify its behaviour.

## 1.2 Indentation

Python relies on indentation and whitespaces to mark blocks whereas C and JavaScript use brackets. Any statement that isn’t at the exactly right level of indentation can cause the entire code to crash. While the compulsory indentation ensures the auto formatting of the source code, debugging faulty code becomes a lot harder because indentation errors might be harder to catch.

## 1.3 JavaScript and its Differences

JavaScript has a fair share of similarities and differences with both the languages. For example, when it comes to comments and indentation, JavaScript is virtually the same as C. However, when it comes to something like literals, JavaScript is wildly different from the other two.

JavaScript uses different terms to describe identifiers (names in JS). JS also has reserved names, aka keywords and some special names which are not keywords but still hold some special meaning (NaN).

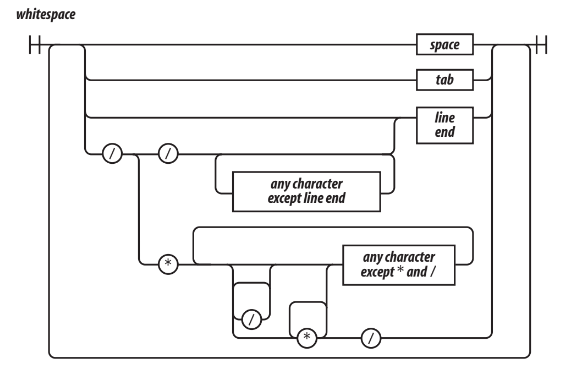
JavaScript literals are very different from C and Python. String literals in JavaScript are in unicode which is accepted in Python, but not in C. JavaScript allows functions to be treated like variables. It also has array, object and regular expression literals.

JavaScript also deals with true and false values a little differently, and makes use of the terms truthy and falsy. In JS, these values are not limited to just boolean values like they are in C and Python.

# 2 Transition Diagrams

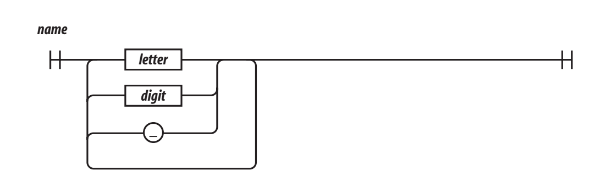
Upon examining the given railroad/transition diagrams in the book, it appears as though these diagrams are meant to represent various context free grammars in a visual manner. All the railroad diagrams give you a starting point, and show you the path(s) you can take to go to a legal JS syntax conforming to its grammar rules

## Examples 2.1 Whitespace



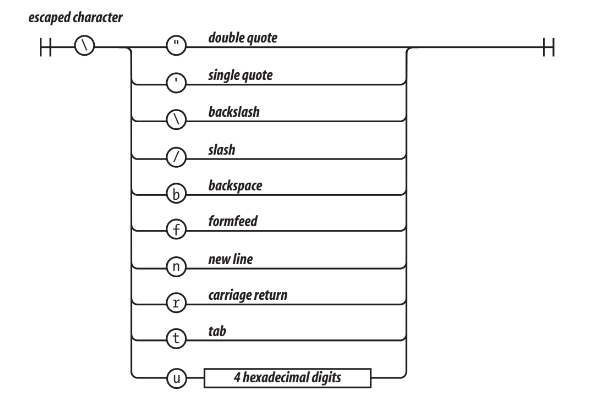
1. Start with space/tab/line end
2. Or turn into a // or a /\* \*/ comment

## 2.2 Names



1. Start with a letter, and follow it up with more letters, digits, or underscores

## 2.3 Escaped Character



1. Start with a ‘\’
2. Follow it with one of the listed letters/characters to create an escaped character (no space after the ‘\’)

These were a few examples and interpretations of railroad/transition diagrams. The remaining diagrams can be interpreted in a similar fashion.