Compilers-I

CS 3510 Spring 2019

Assignment 2:

Lexical Conventions and Grammars of Languages: C vs. Python vs. Javascript

INSTRUCTOR: DR. Ramakrishna Upadrasta

By:
Vijay Tadikamalla
CS17BTECH11040

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Lexical Conventions of C

Tokens

There are six classes of tokens: identifiers, keywords, constants, string literals, operators, and other separators.

Comments

// are used for single line comments
/* Multiline comment */

Identifiers

An Identifier can only have alphanumeric characters(a-z , A-Z , 0-9) and underscore(_). The first character of an identifier can only contain alphabet(a-z, A-Z) or underscore (_). Identifiers are also case sensitive in C. For example, name and Name are two different identifiers in C.

Keywords

Some Identifiers are reserved for the use as keywords like int, for, if, typedef, etc.

Literals/constants

- Integer-constant
- Character-constant
- Floating-constant
- Enumeration-constant
- String Literals

Lexical Conventions of Python

Line structure

A Python program is divided into a number of logical lines. The end of a logical line is represented by the token NEWLINE.

Tokens

NEWLINE, INDENT, and DEDENT are some new type of tokens apart from the usual tokens like identifiers, keywords, literals, operators, and delimiters.

Comments

# is used for single line comments
‘’’ Multiline comment’’’ - Make sure to indent the leading ''' appropriately to avoid an IndentationError

Identifiers

It follows the same convention as C.

Keywords

The list of keywords consists of identifiers like elif, return, etc.

Literals/constants

- Booleans
- Integers
- Long integers
- Floating-point numbers
- Complex numbers
- String Literals

Lexical Conventions of Javascript

Tokens

There are six classes of tokens: identifiers, keywords, constants, operators, and other separators.

Comments

// are used for single line comments
/* */ are used multiline comment

Identifiers

It follows the same convention as C.

Keywords

Some Identifiers are reserved for the use as keywords like finally, function, etc.
language. The list does not include some words that should have been reserved but was not, such as undefined, NaN, and Infinity.

Literals/constants

- Javascript has only a single number type i.e 64-bit floating point(double).
- String Literal

2. In the Javascript book, the transition diagrams are discussed (till pages 9 and 10). Go through them carefully. Go through them and write a short note.

- JavaScript offers two forms of comments, block comments formed with /* */ and line-ending comments starting with //. They also are represented in form of the regular expressions. However, we should be careful while using them because */ or /* can occur in our code.
- Javascript has only a single number type i.e 64-bit floating point(double). This number literal has three parts:
  - Integer
  - Fraction
  - Exponent
- JavaScript has string literals which can be wrapped in single quotes and double quotes. It does not have char type, so to represent a char, it just makes a string with just one character.
- In strings, we may want to add some symbols like double quotes,single quotes but unfortunately, the behaviour will be different from what is required. So, JavaScript provides us with escape character \. Ex \n is a new line, \t is a tab.
- Unlike other languages, Javascript has a bit different notation of equality. It uses
  - = :as an assignment operator
  - == :checking equality(not strict)
  - === :checking equality(Strict)
- A computer program is a list of "instructions" to be "executed" by a computer. A JavaScript program is a list of programming statements.
- JavaScript statements are composed of Values, Operators, Expressions, Keywords, and Comments.
- Statements tend to execute in order from top to bottom. Its sequence of execution can be altered by flow control statements like the conditional statements (if and switch), by the looping statements

(while, for, and do) or by the disruptive statements (break, return, and throw), and by function invocation.

- Each control statement has a specify syntax associated with it.
- An expression is any valid set of literals, variables, operators, and expressions that evaluates to a single value. The value may be a number, a string, or a logical value. Conceptually, there are two types of expressions: those that assign a value to a variable, and those that simply have a value.
- JavaScript also many operators like +,-, * /, % ,=, !==,==,=== etc
- JavaScript also has well-defined operator precedence.
- All the expression are evaluated using the operator precedence rules