

PyJac

Language Specification

To define the language, we have to give the details about the following:

I. Words

a. Keywords

- **if**
- **else**
- **do**
- **while**
- **for**
- **switch**
- **case**
- **default**
- **Loop Control:**
 - **continue**
 - **break**
- **return**
- **void**
- **class**
- **condensed** - ***abstract***
- **symbol** - ***interface***
- **concrete** - ***final***
- **static**
- **function** - ***def (in Python)***
- **Access Modifiers:**
 - **public**
 - **private**
 - **preserved** - ***protected***
- **expands** - ***extends***
- **applies** - ***implements***
- **try**
- **catch**
- **finally**
- **throw**
- **Chaining:**
 - **self** - ***this***
 - **parent** - ***super***

b. Comments

- \$ Comment - Start of a single line comment
- \$# Comments #\$ - Start and ending of a multiline comment

c. Datatypes

- int - Integer
- float - Float
- char - Character
- String - Strings
- bool - Boolean
- exception - Exception

II. Punctuators

- ; (Semicolon) - To terminate lines
- , (Comma) - Separator
- . (Dot) - To access attributes and methods
- () (Parenthesis) - To declare functions
- { } (Braces) - Scope of methods
- [] (Brackets) - Index no# in dimensional array
- // : (Colon)

III. Operators

a. Arithmetic

+, -, *, /, %

b. Relational

<, <=, >, >=, !=, ==

c. Logical

&&, ||, !

d. Assignment

=

e. Increment Decrement

++, --

f. Compound Assignment

+=, -=, *=, /=, %=

IV. Identifiers (Rule – Regex | RE)

- a. Name should start with an alphabet
- b. Followed by any alphanumeric character

Alphabet => A-Z, a-z Alphanumeric Character => A-Z, a-z, 0-9, _

Regex: `[A-Za-z] /w*`

V. Constants

a. Integer

Regex: `[+]?[0-9]+`

Integer numbers

b. Float

Regex: `[+]?[0-9]*[.][0-9]+`

Number having single . (decimal) may or may not have numbers following decimal

c. Boolean

true

false

d. Character

Regex: `\[\\sA-Za-z0-9-!@#$%^&*()+=;<>,.?/{}[]|\'`

Single character or any combination of defined double characters which Starts and Ends with single quotes

e. String

Regex: `\"(.*)\"`

Any combination of characters Starts and Ends with double quotes