Token in python

Token in python

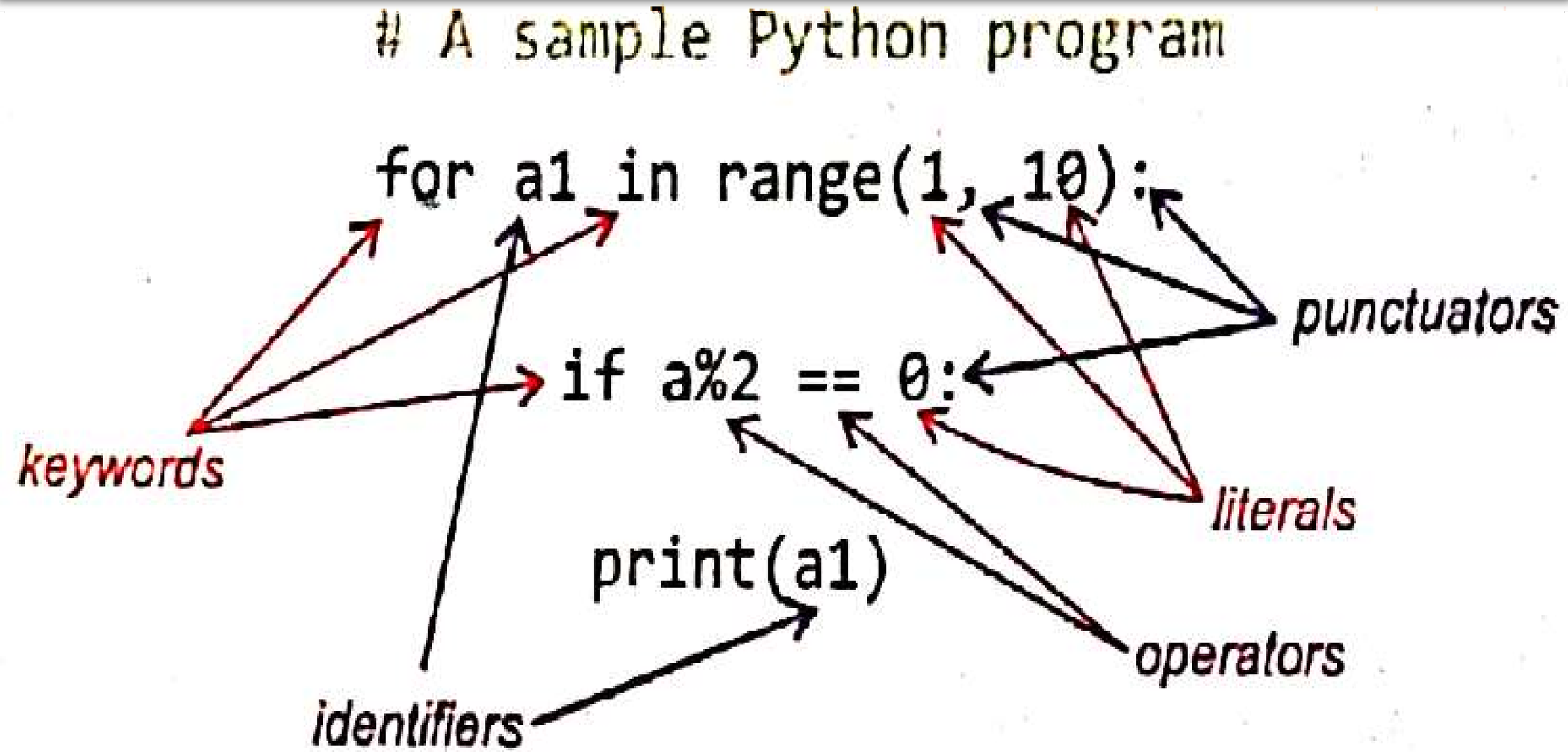
The smallest individual unit in a program is known as a token Or a lexical unit.

Python has following tokens:

- **Keywords**
- **Identifiers (Name)**
- **Literals**
- **Operators**
- **Punctuators**

Viva Technologies

Token in python



Keywords

Keywords

- A Keyword is a word having Special meaning reserved by python.
- These are reserved for special purpose and most not be used as normal identifier names.

False	**assert**	**del**	**for**	**in**	**or**	**while**
None	**Break**	**elif**	**from**	**is**	**pass**	**with**
True	**class**	**Else**	**Global**	**lambda**	**raise**	**yield**
and	**continue**	**Except**	**if**	**nonlocal**	**return**	
as	**def**	**finally**	**import**	**not**	**try**	

Identifiers (Name)

Identifiers (Name)

Identifiers are the names given to different parts of the program

The naming rules for python identifiers can be summarized as follows:

- Variable names most only be in non-keyword word with no spaces in between.
- Variable names most be made up of only letters, numbers, and underscore (_).
- Variable names cannot begin with a number, although they can contain numbers

NOTE : Python is case sensitive as it treats upper and lower-case characters differently

Identifiers (Name)

Valid identifiers	
Myfile	Date_7_77
MYFILE	_DS
_CHK	FILE13
Z2T0Z9	_HJI3_JK

Invalid identifiers	
DATA-REC	Contain special character – (hyphen) (other than A-Z, a-z and _(underscore)
29CLCT	starting with a digit
Break	reserved keyword
My.file	contains special character dot (.)

Literals / Values

Literals / Values

(i) String literals

(ii) Numeric literals

- (a) Int (signed integers)
  - Decimal
  - Octal
  - Hexadecimal
- (b) Floating point literals.
- (c) Complex number literals

(iii) Boolean Literals

(iv) Special Literal - None

Literals/value

(i) String literals

- A string literal is sequence of chapters surrounded by quotes single or double or triple quotes.
- String literal can ether be single line string or multi line strings.

```
>>> Text1 = "Hello World"
```

Single line string

```
>>> Text2 = "Hello\
    World "
```

Multi-line string

```
Text3 = '''Hello
    World'''
```

No backslash needed

Escape Sequence	What it does (non-graphic character)	Escape sequence	What it does (non-graphic character)
\\	Backlash (\)	\r	Carriage Return (CR)
\’	Single Quote (‘)	\t	Horizontal Tab (TAB)
\”	Double Quote (“)	\uxxxx	Character with 16-bit hex value xxxx (Unicode only)
\a	ASCII Bell (BEL)	\Uxxxxxxxx	Character with 32-bit hex value xxxxxxxx (Unicode only)
\b	ASCII Backspace (BS)	\v	ASCII Vertical Tab (VT)
\f	ASCII Form feed (FF)	\000	Character with octal value 000
\n	New line character	\xhh	Character with hex value hh
\N{name}	Character named name in the unicode,		

Literals/value

(ii) Numeric literals

- Numeric literals are numeric values and these can we one of the following types:

(a) Int (signed integers) often called just integers or ints, are positive or negative whole numbers with no decimal point.

The integer literals can be written in:

- Decimal form: an integer beginning digits 1–9.
- Octal form: an integer beginning with 0o (zero followed by letter o)
- Hexadecimal form : an integer beginning with 0x (zero followed by letter X)

Literals/value

(b) Floating point literals.

- Floating point literals floats represent real numbers and are written with a decimal point dividing the integer.

(c) Complex number literals

- Complex number literals are of the form a+bJ, where a and b are floats and J (or j) represents √–1, which is an imaginary number).
- a is the real part of the number, and b is the imaginary part.

Literals/value

(iii) Boolean Literals

- A Boolean Literal in python is used to represent one of the Boolean values i.e., true (Boolean true) or false (Boolean false).
- A Boolean Literal con either have value as true or as false.

(iv) Special literal -None

- Python has one special literal, which is none. The none literal is used to indicate absence of value.
- Python can also Store literal collections, in the form of tuples and lists etc.

Operators

Operators

- Operators are tokens that trigger some computation / action when applied to variables and other objects in an expression.
- The operators can be:
  - Arithmetic operators (+,–,*,/,%,**,//),
  - Bitwise operators (&,^,|,),
  - Shift operators (<<,>>),
  - Identify operators (is, is not),
  - Relational operators (>,<,>=,<=,==,!=),
  - Compound assignment operators (/=,+=,–=,*/,%=,**=,//=).

Punctuators

Punctuators

- Punctuators are symbols that are use in programming languages to organize sentence structures, and indicate the rhythm and emphasis of expression, statements, and program structure.
- Most common punctuators python programming language are :

‘	“	#	\	( )	[ ]	{ }	@	,	:	.	=

Viva Technologies

PYTHON TEST – 1.2
TOKENS IN PYTHON