Chapter 2

By-
Neha Tyagi
PGT (CS)
KV 5 Jaipur (II Shift)
Jaipur Region
In order to provide an input, process it, and to receive output, we need to write a program.

Program, is a group of instructions which controls processing.

In other words, base for processing is ‘the Program’.

In this chapter we will come to know about various element of processing like – character set, token, expressions, statements, input.
Python Character Set

- Character Set is a group of letters or signs which are specific to a language.
- Character set includes letter, sign, number, symbol.
  - Letters: A-Z, a-z
  - Digits: 0-9
  - Special Symbols: _, +, -, *, /, (, ), {, } . . . Etc.
  - White Spaces: blank space, tab, carriage return, newline, formfeed etc.
  - Other characters: Python can process all characters of ASCII and UNICODE.
Tokens

- Token- is the smallest unit of any programming language. It is also known as Lexical Unit.

Types of token are:

i. Keywords
ii. Identifiers (Names)
iii. Literals
iv. Operators
v. Punctuators
Keywords

- Keywords are those words which provide a special meaning to the interpreter.
- These are reserved for specific functioning.
- These cannot be used as identifiers, variable names, or any other purpose.
- Available keywords in Python are:

<table>
<thead>
<tr>
<th>False</th>
<th>class</th>
<th>finally</th>
<th>is</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>continue</td>
<td>for</td>
<td>lambda</td>
<td>try</td>
</tr>
<tr>
<td>True</td>
<td>def</td>
<td>from</td>
<td>nonlocal</td>
<td>while</td>
</tr>
<tr>
<td>and</td>
<td>del</td>
<td>global</td>
<td>not</td>
<td>with</td>
</tr>
<tr>
<td>as</td>
<td>elif</td>
<td>if</td>
<td>or</td>
<td>yield</td>
</tr>
<tr>
<td>assert</td>
<td>else</td>
<td>import</td>
<td>pass</td>
<td></td>
</tr>
<tr>
<td>break</td>
<td>except</td>
<td>in</td>
<td>raise</td>
<td></td>
</tr>
</tbody>
</table>

Neha Tyagi, KV 5 Jaipur
Identifiers

• These are building blocks of a program and are used to give names to different parts/blocks of a program like - variable, objects, classes, functions.
• An identifier may be a combination of letters and numbers.
• An identifier must begin with an alphabet or an underscore(_). Subsequent letters may be numbers(0-9).
• Python is case sensitive. Uppercase characters are distinct from lowercase characters (P and p are different for interpreter).
• Length of an Identifier is unlimited.
• Keywords can not be used as an identifier.
• Space and special symbols are not permitted in an identifier name except an underscore( _) sign.
• Some valid identifiers are –
  – Myfile, Date9_7_17, Z2T0Z9, _DS, _CHK FILE13.
• Some invalid identifiers are –
Literals / Values

• Literals are often called Constant Values.

• Python permits following types of literals -
  – String literals - “Pankaj”
  – Numeric literals – 10, 13.5, 3+5i
  – Boolean literals – True or False
  – Special Literal None
  – Literal collections

Neha Tyagi, KV 5 Jaipur
String Literals

• String Literal is a sequence of characters that can be a combination of letters, numbers and special symbols, enclosed in quotation marks, single, double or triple (" " or ‘ ‘ or “ ”).

• In python, string is of 2 types-
  – Single line string
    • Text = “Hello World” or Text = ‘Hello World’
  – Multi line string
    • Text = ‘hello\n    world’ or Text = “”hello world “”
Numeric Literals

• Numeric values can be of three types -
  
  – int (signed integers)
    • Decimal Integer Literals – 10, 17, 210 etc.
    • Octal Integer Literals - 0o17, 0o217 etc.
    • Hexadecimal Integer Literals – 0x14, 0x2A4, 0xABD etc.

  – float (floating point real value)
    • Fractional Form – 2.0, 17.5 -13.5, -.00015 etc.
    • Exponent Form - -1.7E+8, .25E-4 etc.

  – complex (complex numbers)
    • 3+5i etc.
Boolean Literals

- It can contain either of only two values – True or False
  - A = True
  - B = False

Special Literals

- None, which means nothing (no value).
  - X = None
Operators

• An Operator is a symbol that trigger some action when applied to identifier (s)/ operand (s)
• Therefore, an operator requires operand (s) to compute upon.

example:

\[ c = a + b \]

Here, a, b, c are operands and operators are = and + which are performing differently.
Types of Operators

• Python supports following types of operators -
  – Unary Operator
    • Unary plus (+)
    • Unary Minus (-)
    • Bitwise complement (~)
    • Logical Negation (not)
  – Binary Operator
    • Arithmetic operator (+, -, *, /, %, **, //)
    • Relational Operator(<, >, <=, >=, ==, !=)
    • Logical Operator (and, or)
    • Assignment Operator (=, /=, +=, -=, *=, %=, **=, //=)
    • Bitwise Operator (& bitwise and, ^ bitwise xor, | bitwise or)
    • Shift operator (<< shift left, >> shift right)
    • Identity Operator (is, is not)
    • Membership Operator (in, not in)
Punctuators

- In Python, punctuators are used to construct the program and to make balance between instructions and statements. Punctuators have their own syntactic and semantic significance.

- Python has following Punctuators -
  ‘, ”, #, \, (, ), [, ], {, }, @. ,, :, .. ``, =
A Python Program Structure

```python
# programs components
# function definition hello()

def hello():
    print("Hello how are you")

#main program code
a=15
b=20
print(a+b)
if a>b:
    print("Greater is : ",a)
else:
    print("Greater is : ",b)

hello()"
```

- **Comments**
- **Function**
- **Statements**
- **Block**
- **Indentation**
- **Function Calling**
A Python Program Structure

• As we have seen in previous slides, a program contains following components -
  – Expressions like a+b, a>b etc.
  – Statements like a=10, c=a+b etc.
  – Comments, lines starting with #.
  – Function, block starting with def keyword
  – Blocks and indentation like if and else blocks

*These will be explained in detailed further.
Thank you

Please follow us on our blog-

www.pythontrends.wordpress.com