PYTHON FUNDAMENTALS

As per CBSE curriculum
Class XI

INTRODUCTION

- 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-
  - Keywords
  - Identifiers (Names)
  - Literals
  - Operators
  - Punctuators

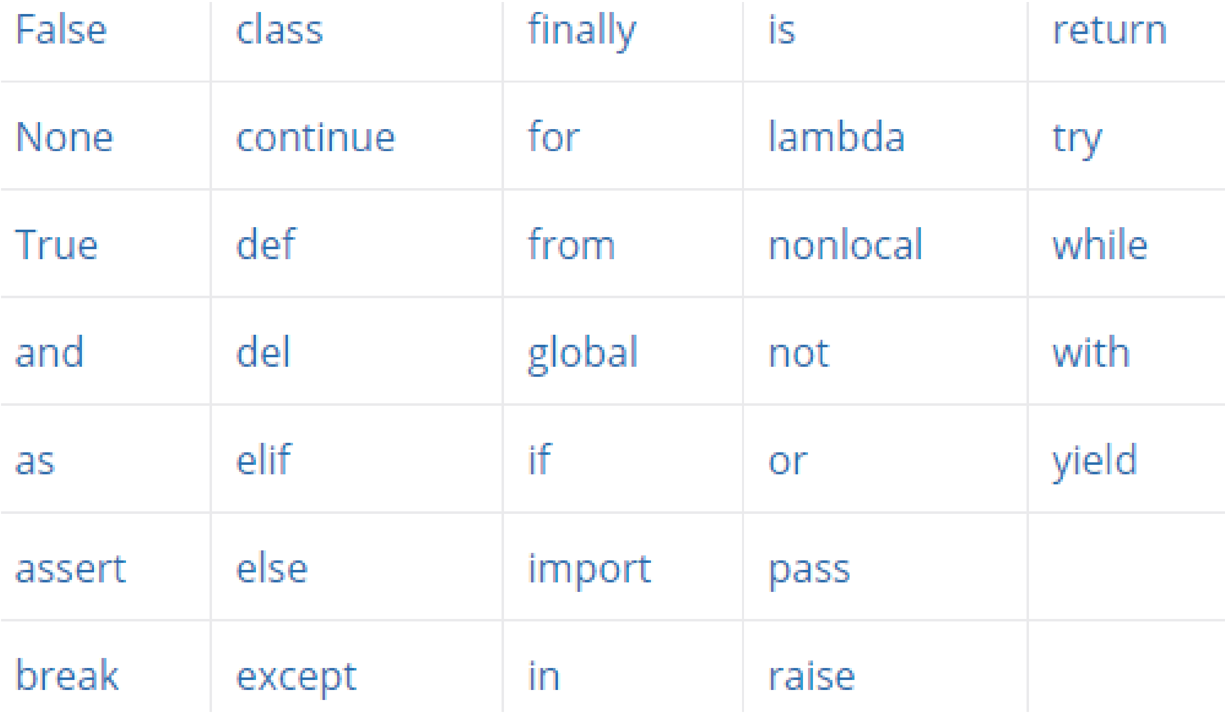
KEYWORDS

Keywords are those words which provides a special meaning to interpreter.

These are reserved for specific functioning.

These can not be used as identifiers, variable name or any other purpose.

Available keywords in Python are-



False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

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 invald identifiers are –
- DATA-REC, 29COLOR, break, My.File.

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

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\
  world’ or Text = ‘’’hello
  word ‘’’

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

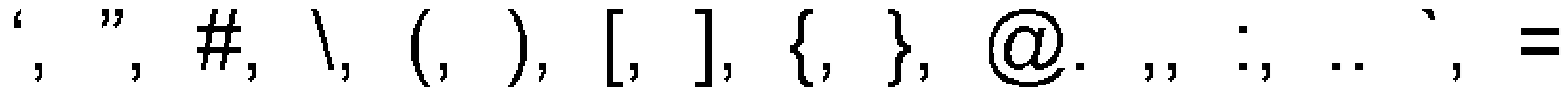
Python supports following types of operators -

1. Unary Operator
   - Unary plus (+)
   - Unary Minus (-)
   - Bitwise complement (~)
   - Logical Negation (not)

2. Binary Operator
   - Arithmetic operator (+, -, *, /, %, **, //)
   - Relational Operator(<, >, <=, >=, ==, !=)
   - Logical Operator (and, or)
   - Assigment 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)

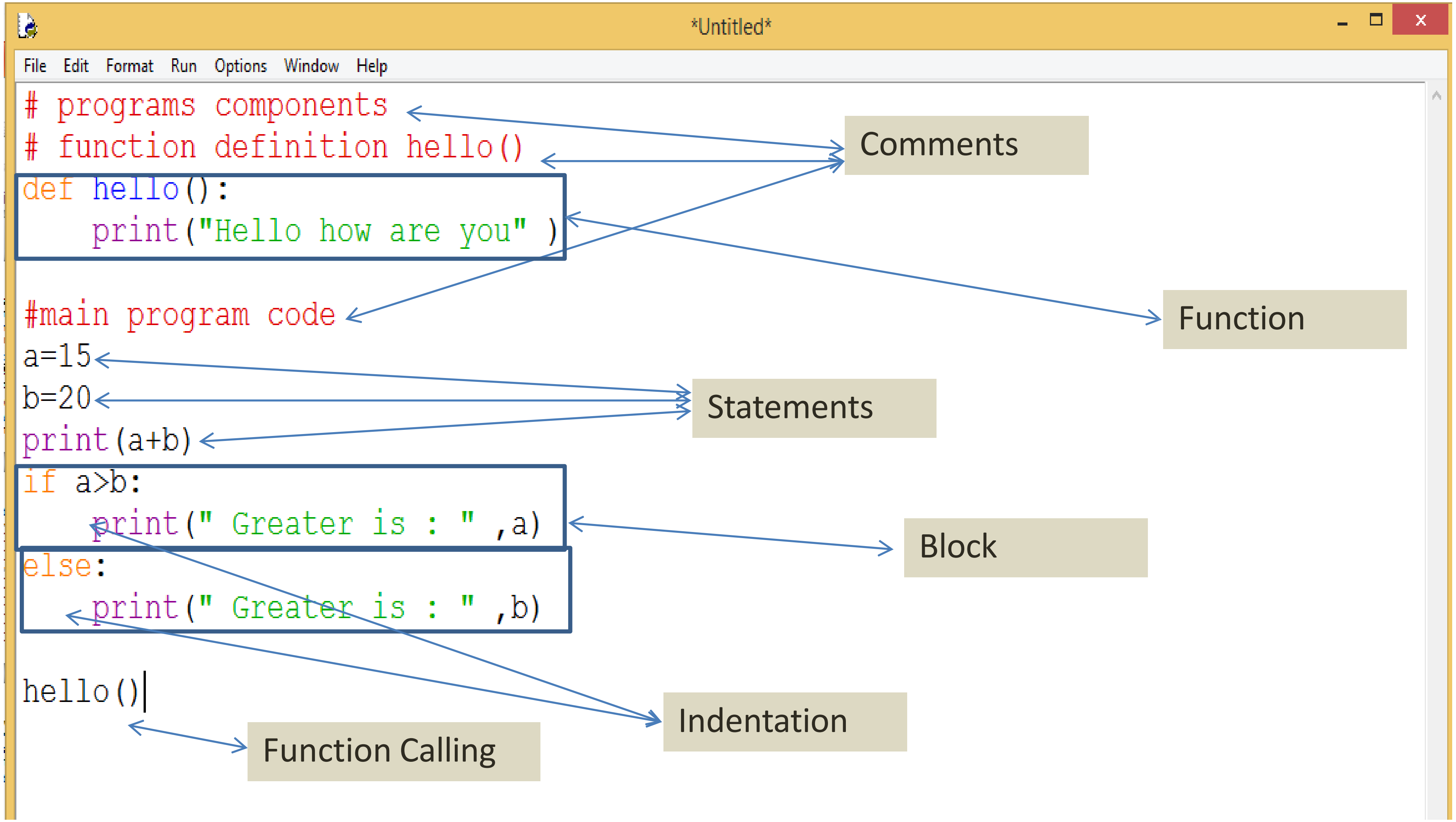
PUNCTUATORS

- 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



```
# 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()
```

A PYTHON PROGRAM STRUCTURE

- As we have seen in previous slides, a program contains following compnents-
- 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