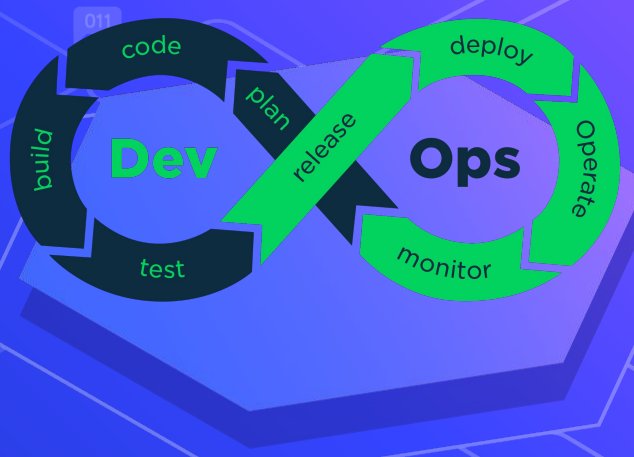
DevOps Beginner Series

DevOps Malayalam



Rajesh Gopinathan
Corporate Technical Trainer

DevOps Malayalam

Program Overview

Targeted Audience

- Any one who has an urge to learn DevOps practice.

Prerequisites

- Basic understanding of managing a computer.

Course Goal

- The goal of this free seminar/workshop is to help new aspirants to have a general understanding of DevOps practices from an industry point of view.

DevOps Malayalam

Course Design

Module 1

- Linux for DevOps
- Bash Scripting
- Python for DevOps
- Golang
- GIt & GitHub

Module 2

- Virtualization & Cloud
- Major public cloud providers
- DevOps Methodology
- Software Architecting

Module 3

- Docker
- Jenkins & Maven
- Ansible
- Terraform

Module 4

- Microservices Architecture
- Kubernetes
- Monitoring
- Log Management

DevOps Malayalam

Python for DevOps

rajg21@hotmail.com
Whatsapp +919986368187

DevOps Malayalam

Contents

- Python.
- Features of Python.
- History of ‘Python’.
- Application of Python language.
- Version and Flavors of Python.
- PSF,PEP,PyPI,andpip.
- Installation of Python.
- Python Code Execution.
- Byte Code.
- Different Implementation of Python.
- Python Shell.
- IDE.
- How to Run Python code.
- Help().
- Comments, Statement, Indentation and DocStrings.

Contents

- Check version.
- Python Identifiers.
- Python keywords.
- OOPS in python.
- Objects.
- Mutable and immutable.
- Variables and Constants.
- Input() and print().
- Literals.
- Data Types Introduction.
- Functions.
- Return Statement.

Devops Malayalam

Contents

- bytes and bytearray.
- Functions.
- Intro to oops concept.
- Intro to Methods.
- Intro to Libraries.
- Intro to Packages.
- Quiz Time.

Python

Interpreted, high-level, general-purpose, programming language

Features of Python

- Easy to learn and code, with simple syntax.
- Code readability and indentation.
- Multiple programming paradigms like Procedural,oops & Functional programming.
- Dynamically-typed and garbage-collected.
- Case sensitive.
- Platform Independent.
- Powerful packages.
- Can extended python functionality by attaching libraries from different languages.
- Supports embedding to other languages.
- Freeware, Open source, copyrighted under a GPL-compatible license.

History of Python

- Guido van Rossum first released Python in 1991 Feb 20 as Python 0.9.0.
- Python 2.0 was released in 2000
- Python 3.0 was released in

2008 and was a major revision of the language that is not completely backwards compatible and much Python 2 code does not run unmodified on Python 3.

- Python 2 was discontinued with version 2.7.18 in 2020.

The Name Python

- Monty Python’s Flying Circus is a BBC Comedy TV series from the year 1969-1974.
- While Guido van Rossum was implementing Python, he was also reading the

published scripts from Monty Python’s Flying Circus.

- Python = Inspired By Monty Python.

Version and Flavors of Python

- The latest version as of now is 3.9.6 and downloadable from PSF.
- Python version 2 is no more officially supported.
- Python3 is not backward compatible with python2.
- The default implementation of Python from PSF is Cpython.
- Different vendors created different flavors of python as per their need.
- Jython ,iron python, Anaconda Python, pypy, stackless Python(… and counting…)

are different flavors of Python available in Market.

- (openSource)

PSF

- https://www.python.org/.
- The place from where you can start your journey with Python.
- https://docs.python.org/3/.
- “The Python Software Foundation is the organization behind Python. Become a

member of the PSF and help advance the software and our mission”.

PEP

- Python Enhancement Proposals.
- PEP contains the index of all Python Enhancement Proposals, known as PEPs.
- PEP numbers are assigned by the PEP editors, and once assigned are never

changed.

- To know more about PEP.
- PEP 1 -- PEP Purpose and Guidelines.
- Try this …
- PEP 8 -- Style Guide for Python Code.

PyPI

- Python Package Index.
- The Python Package Index (PyPI) is a repository of software for the Python

programming language.

- PyPI helps you find and install software developed and shared by the Python

community.

- Package authors use PyPI to distribute their software.
- In a nutshell : Find, install and publish Python packages with the Python

Package Index.

- Visit us @ https://pypi.org/

pip

- pip is a de facto standard package-management system used to install and

manage software packages written in Python.

- Many packages can be found in the default source for packages and their

dependencies — (PyPI).

- Most distributions of Python come with pip preinstalled.

Installation of Python

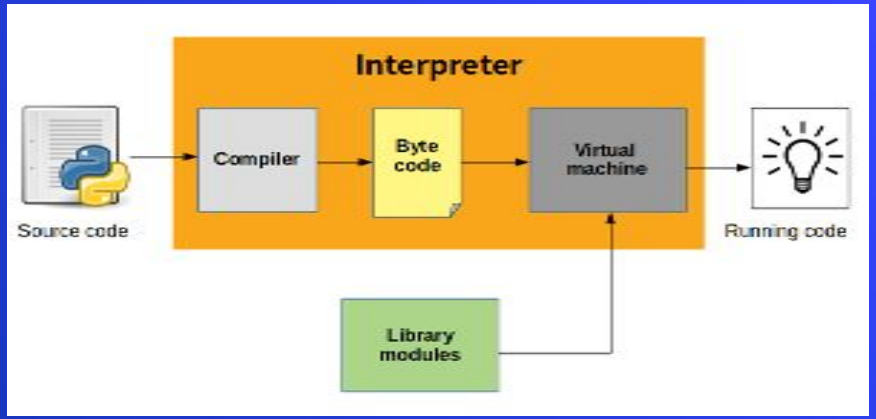
Lets get our hands dirty.

Python Code Execution

Python is considered as an interpreted language, but this is partially true, as it

involves a compilation step that converts the python code into byte code which is

stored with a .pyc or .pyo format that gets deleted once the program is executed.



Byte Code

• Byte code is a lower-level, and platform-independent , representation of our
source code.
• Byte code is binary representation executed by virtual machine (not by CPU
directly).
• The virtual machine (which is written different for different machines)
converts binary instruction into a specific machine instruction.
• One of the language that uses the concept of Byte code is python.

Different implementation of Python

• An “implementation” of Python should be taken to mean a program or environment which provides support for the execution of programs written in the Python language.
• The source code is first compiled and converted to a byte code.
• This byte code is then executed on a virtual machine.
• Python implementations are defined based on the language these virtual machines are build on or the way it is interpreted/compiled.
• Python program runs directly from the source code . so, Python will fall under byte code interpreted.
• This byte code can be interpreted (official CPython), or JIT compiled (PyPy).
• (flavors)

Python shell/REPL

Python provides a Python Shell, which is used to execute Python commands and
display the result.

- It is also known as REPL (Read, Evaluate, Print, Loop), where it reads the command,
evaluates the command, prints the result, and loop it back to read the command
again.
- To run the Python Shell, open the command prompt or powershell on Windows
and type [python or py] and press enter.
- A Python Prompt comprising of three greater-than symbols >>> appears.

What is an IDE

IDE is the Integrated Development Environment that provides the user interface
for code development, testing and debugging features.
• A software application which contains development tools such as text editors,
code libraries, compilers, and test platforms .
• Default IDE that comes with Cpython is IDLE.
• Many different IDE available like
pydev,pycharm,spider,Thonny, Atom/Atom-IDE

How to Run python codes

From the cmd or shell prompt .

- python –c cmd,Python file.py
- From the python shell. (Interactive mode)
- From your favorite IDE.
- Passing a file with .py extension to your shell or IDE.

help needed

help()
• If no argument is passed, Python's help utility (interactive help system) starts on
the console.
• >>> help()
• Then, you can enter the name of the topic to get help on writing Python programs
and using Python modules.
• To come out of help utility use quit.

Comments

- The comment is that part of the code which does not get executed.
- In programming, it is mainly used for describing the logic of the code and for other purposes.
- A comment may appear at the start of the line or following by the whitespace but never come in between the string.
- For multiline comments, you can use the hash character at the beginning of every line.
- # This is a one line comment.

Python Statement

- Instructions that a Python interpreter can execute are called statements.
- var1=‘PYTHON’
- This is an assignment statement.

Multi-line statement

- (Explicit)
- In Python, the end of a statement is marked by a newline character. But we can
make a statement extend over multiple lines with the line continuation character
(\).
- Implicit.
- (),[],{}.
- We can also put multiple statementsin a single line using semicolons, as follows:
- Var1=“learn”; var2=“python”.

Python Indentation

• Most of the programming languages like C, C++, and Java use braces{ } to define a
block of code. Python, however, usesindentation.
• A code block (body of a function, loop, etc.) starts with indentation and ends with
the first unindented line. The amount of indentation is up to you, but it must be
consistent throughout that block.
• Generally, four whitespaces are used for indentation and are prefered over tab.
• Indentation can be ignored in line continuation, but it's always a good idea to
indent. It makes the code more readable.
• Incorrect indentation will result in IndentationError.

Docstrings in Python

• Python docstrings are the string literals that appear right after the definition of a
function, method, class, or module that help explain the code.
• Docstrings are documentation strings It can span across multiple lines.
• To create a docstring either use triple quotes.’’’ or “””.
• “““
This is a docstring.
”””

The docstrings are associated with the object as their __doc__ attribute.

Check Version

- At cmd prompt
- python --version
- python –V.
- In IDE
- Print(sys.version_info)
- Chk ver.py

Try Those

```
import platform

Print(platform.python_imlplementation())
```

Python Identifiers

- An identifier is a name given to entities like class, functions, variables, etc.
- It helps to differentiate one entity from another.
- Identifiers can be a combination of letters in lowercase (a to z) or uppercase (AtoZ) or digits (0 too 9) or an underscore _.
- The 1st character can’t be a number.
- Python has a set of keywords that are reserved words.
- keywords cannot be used as identifiers.
- According to the docs, you can have an identifier of infinite length. However, the PEP-8 standard sets a rule that you should limit all lines to a maximum of 79 characters.

Keywords

- In python3 there are 33 keywords.
- Reserved words (also called keywords) are defined with predefined meaning and syntax in the language.
- Reserved words can’t be used as identifiers for other programming elementslike name of variable, function etc.
- Eg: True,False,None
- Chk keyword.py

OOPS

• Object-Oriented Programming is a methodology or paradigm to design a program
using classes and objects.
• It uses the idea of “ objects ” to represents data and methods.
• Itsimplifies the software development and maintenance by providing some
concepts.
• It is also, an approach used for creating neat and reusable code instead of a
redundant one.
• The program is divided into self-contained objects or several mini-programs.

Objects

- In Python, everything is treated as an object.
- Every object has three attributes:
- Identity – This refers to the addressthat the object refers to in the computer's memory. Id()
- Type – This refers to the kind of objects that is created. Strings, integers etc. (runtime). type()
- Value – This refers to the value stored by the object. List=[1,2,3] would hold the numbers 1,2 and 3
- While ID and Type cannot be changed once it's created, values can be changed for Mutable objects.

Mutable vs Immutable Objects in Python

- Every variable/name in python holds an instance of an object.
- There are two types of objects in python.
- Mutable and Immutable objects.
- Whenever an object is instantiated, it is assigned a unique object id.
- The type of the object is defined at the runtime and it can’t be changed afterwards.
- Mutable objects can change their state or contents
- Immutable objects can’t change their state or content.

Variables or is it just a name

• Variable is a name for a location in memory.
• It can be used to hold a value and reference that stored value within a computer program.
• Python does not have variables, instead, it has ‘names’. ‘’Surprised’’O
• Python worksin a different way and technically, in Python variable is just a ‘name’ and is a label for an object and used to refer to a value.
• We never specify ‘type’ information while creating an object.
• The assignment operator in ‘Python 3’ is a single equals sign (=).
• This operator assigns the value on the right-hand side to the variable/name on the left-hand side.

Variables

- A python ‘name’ can change its type.
- A single Python object can have lots of names.
- Two names will point to the same object if the id() method returns the same value.
- Values are not updated, instead, a new object is pointed.

Variables

- Single assignment.
- var1=5.
- Multiple Assignment .
- Python allows usto assign one single value to several variables at the same time
- var1=var2=var3=var4=‘Python’.
- Assign multiple objects to multiple variables.
- var1,var2,var=1, ’Python’,3+4J.
- Chk(var.py)

Constants

- A constant is a type of variable whose value cannot be changed.
- Assigning value to constant in Python
- In Python, constants are usually declared and assigned in a module.
- The module is a new file containing variables, functions, etc which is imported to the main file.
- Inside the module, constants are written in all capital letters and underscores separating the words.

Feeding User Input from stdin

- we can ask the program to take user input from the keyboard using input()
function.
- input("Enter your favorite language:").
- The py shell will wait for user to enter the value.
- All values entered will be considered as string value.
- Chk(input.py)

Getting output

- Using print() function the output can be displayed to the terminal.
- print(‘welcome’).
- var1=10 .
- print(var1).
- print() gives a newline.
- Default end of print() is a newline.
- Default delimiter is a whitespace.
- Chk(print.py)

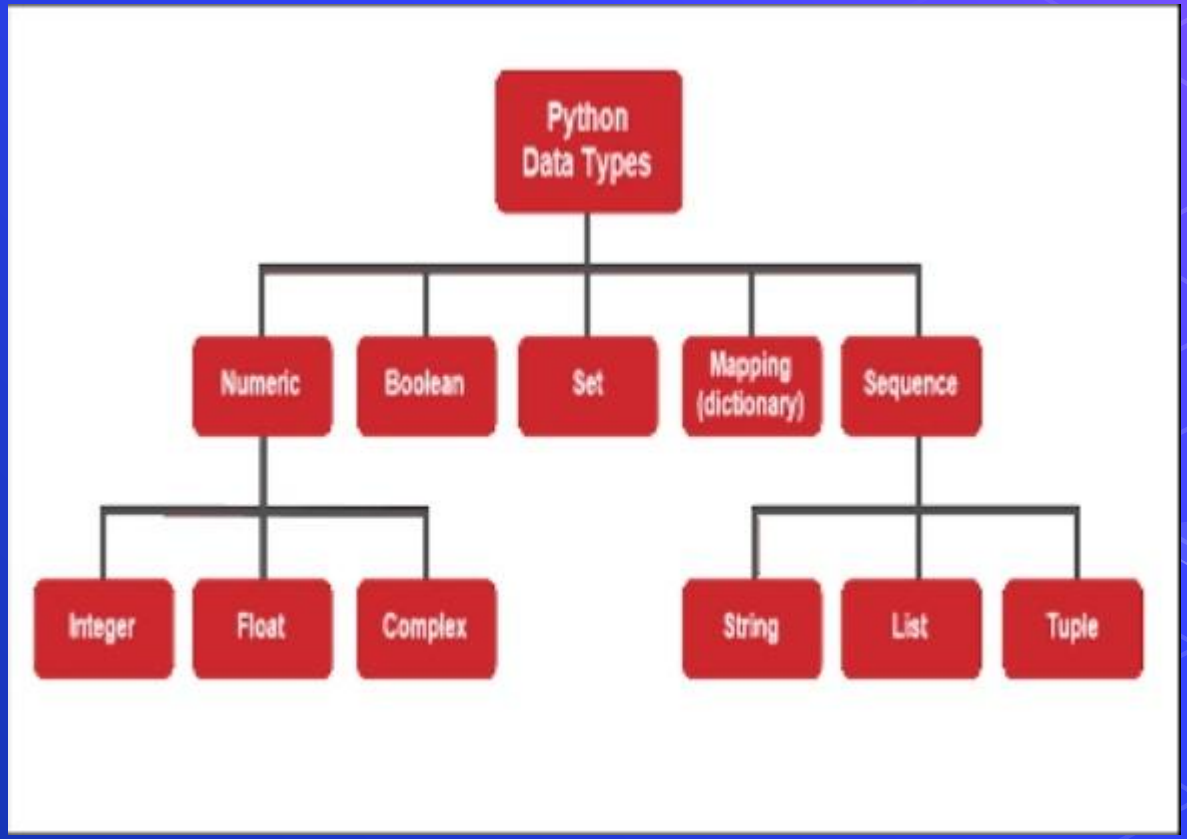
Literals

- Literal is a raw data given in a variable or constant.
- In Python, there are various types of literals.
- string, numeric, None,boolene,literalCollections…

Data Types Introduction

• Every programming language has built-in data types, including Python.
• Data types provide information about the different kinds of data that a variable/
name can have and dictate the programming flow.

Data Types in Python



Data Types in Python

- Numeric data types: int, float, complex
- String (text)data type: str
- Sequence types: list, tuple, range
- Binary types: bytes, bytearray
- Mapping data type: dict
- Boolean type: bool
- Set data types: set, frozenset
- None data type

Categories of Data Types

• A data type is a characteristic that tellsthe compiler (or interpreter)
how a programmer intends to use the data.
• There are two general categories of data types, differing whether the
data is changeable after definition:
• 1. Immutable. Data types that are not changeable after assignment.
• 2. Mutable. Data types that are changeable after assignment.

Objects of built-in type that are mutable

- Lists.
- Sets.
- Dictionaries.
- User-Defined Classes

Objects of built-in type that are Immutable

- Numbers(Integer, Float, Complex)
- Booleans.
- Strings.
- Tuples.
- Frozen Sets.
- User-Defined Classes (It purely depends upon the user to define the characteristics).

Type Conversion in Python

- The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion.
- Python has two types of type conversion.
- Implicit Type Conversion.
- Explicit Type Conversion.

Implicit Type Conversion

- In Implicit type conversion of data types, the Python interpreter automatically converts one data type to another without any user involvement.
- Python avoids the loss of data in Implicit Type Conversion.

Explicit Type Conversion

(or)Type Casting

- Explicit Type conversion is also know as Type casting.
- The user usestype conversion functions (constructor functions) for converting purposes.
- In Type Casting, loss of data may occur as we enforce the object to a specific data type.
- syntax for such conversion will be:
- (desired_datatype)(expression).

Type Casting

- int(y [base]), float(y) , complex(real [imag]).
- str(y), tuple(y), list(y), set(y).
- dict(y) It creates a dictionary and y should be a sequence of (key,value) tuples.
- ord(y) It converts a character into an integer.
- bin(y), hex(y), oct(y). [base conversion].

Strings

• In Python3, a string is a sequence of Unicode characters enclosed in ‘single’ or “
double” quotes.
• A character is simply a symbol.
• The English language has 26 characters.
• Computers do not deal with characters, they deal with numbers (binary).
• Even though you may see characters on your screen, internally it is stored and
manipulated as a combination of 0s and 1s.
• This conversion of character to a number is called encoding, and the reverse processis decoding.
• ASCII and Unicode are some of the popular encodings used.
• Chk(encoding.py)

Multiline Strings

- Multiline strings can be created using triple quotes.
- var1 =

‘’’ Beautiful is better than ugly…
…Explicit is better than implicit...
…Simple is better than complex...
…Complex is better than complicated...
…Readability counts. !!! ’’’

- print(var1).
- Chk(multi.py)

string data type

- String data types are represented by the keyword str.
- String data types in python are surrounded by either single, or double quotation marks.
- No character data type in Python.
- Character is a string data type with length one.
- var3=
“”
- var4=
“ “
- var5=
“a”
- print((var3),type(var3),(var4),type(var4),(var5),type(var5),len(var5))
- Strings are immutable.
- Chk(str.py)

string data type

- String data types are represented by the keyword str.
- String data types in python are surrounded by either single, or double quotation marks.
- No character data type in Python.
- Character is a string data type with length one.
- var3=
“”
- var4=
“ “
- var5=
“a”
- print((var3),type(var3),(var4),type(var4),(var5),type(var5),len(var5))
- Strings are immutable.
- Chk(str.py)

Unpacking Strings

• Python strings are sequences of individual characters, and share their basic
methods of access with those other Python sequences – lists and tuples.
• The simplest way of extracting single characters from strings (and individual
members from any sequence) is to unpack them into corresponding variables.
• if the number of variables we supply doesn't match with the number of characters
in the string, Python will give us an error.
• To over come this we can uses indexing .

Indexing a string

• The index in Python returns the position of the element in the specified list or the
Characters in the string.
• syntax is same for list or a string.
• The reason being that a string in Python is considered as a list of characters
starting from the index 0.
• print(Str[0])
• print(Str[-1])
• Chk(string_index.py)

Slicing a string

- Python offers many ways to substring a string. It is often called ‘slicing’.
- syntax: string [start: end: step]
- Chk(string_slice.py)

Escape Characters

- The “backslash (\)” character is an escape character.
- It has a special meaning when we use it inside the strings.
- The escape character escapes the characters in a string to introduce unique
- inclusion.
- Backlash signifies that the next character after it has a different meaning.
- In Python some characters have a special meaning when used in a string.
- Escape characters removes the special meaning of those special characters

Escape Characters

- Code Result
- \' Single Quote
- \” Double Quote
- \\ Backslash
- \n New Line
- \r Carriage Return
- \t Tab
- \b Backspace
- \f Form Feed
- \N{name} Prints a character from the Unicode database
- \ooo Octal value
- \xhhHex value

Raw and Multiline Strings

- A raw string literal is preceded by r or R, which specifies that escape sequences in the associated string are not translated.
- Not all raw strings are valid.
- A raw string that contains only a single backslash is not valid. Similarly, raw strings with an odd number of ending backslash are also not valid.
- Triple-quoted strings are delimited by matching groups of three single quotes or three double quotes. Escape sequences still work in triple-quoted strings, but single quotes, double quotes, and newlines can be included without escaping them. This provides a convenient way to create a string with both single and double quotes in it:

- Chk(raw_triple.py)

None data type

- There is no null but there is None.
- None is the return value of the function that “doesn’t return anything“.
- None is often used to represent the absence of a value, as default parameters are not passed to the function.
- we can not assign a null value to the variable, and if you do, then it is illegal, and it will raise a SyntaxError.
- The following is not valid
- data = null
- print(data)

- Chk(None.py)

Key points of None in Python

- Comparing None to anything will always return False except None itself.
- None is not a 0.
- None is not an empty string.
- None is not the same as False.

- Use the keyword : None
- data=None
- type(data)

Numeric Data Type

- A number is an arithmetic entity that lets us measure something.
- Python allows usto store the integer, floating, and complex numbers and also lets us convert between them.
- Since Python is dynamically-typed, there is no need to specify the type of data for a variable/name.
- It can hold a value of any length, the only limitation being the amount of memory available.
- Long integer data type in not there in python3.
- Numeric data types are int, float and complex.

int

- int data type can hold signed integers.
- We can write an exponential number using the letter ‘e’ between the mantissa and the exponent.
- Remember that this is power of 10.
- print(2e5) 200000
- To raise a number to another’s power, we use the ** operator.
- a=(3**3).
- Int can represent a dec(10),bin(2),oct(8),orhex(16) number system.(any base)
- 0b,0o,0x. 0B,0O,0X.

- Chk(int.py)

float

- float data types in Python supports floating point real value.
- A float value is only accurate upto 15 decimal places. After that, it rounds the number off.
- Division always results in float value.

- Chk(float.py)

Operators

- Operators are special symbols in Python that carry out arithmetic or logical computation.
- The value that the operator operates on is called the operand.
- >>> 1+2
- 3
- Here, + is the operator that performs addition. 1 and 2 are the operands and 5 is the output of the operation.

Arithmetic Operators

- + addition.
- -subtraction.
- * multiplication.
- / division.
- ** exponentiation.
- //integer division.
- % modular (remainder).

- Chk(cal.py)

Comparison operators

- Comparison operators are used to compare values. It returns either True or False according to the condition.
- > Greater than - True if left operand is greater than the right x > y
- < Less than - True if left operand is less than the right x < y
- == Equal to - True if both operands are equal x == y
- != Not equal to - True if operands are not equal x != y
- >= Greater than or equal to - True if left operand is greater than or equal to the right x >= y
- <= Less than or equal to - True if left operand is less than or equal to the right x <= y

Logical operators

- Logical operators are the and, or, not operators.

and True if both the operands are true	x and y
or True if either of the operands is true	x or y
not True if operand is false (complements the operand)	not x

Bitwise operators

- Bitwise operators act on operands as if they were strings of binary digits.
- They operate bit by bit, hence the name.

&	Bitwise AND	x & y
|	Bitwise OR	x | y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x >> 2
<<	Bitwise left shift	x << 2

Assignment operators

- Assignment operators are used in Python to assign values to variables.
- a = 10 is a simple assignment operator that assigns the value 10 on the right to the variable a on the left.
- There are various compound operators in Python like a += 10 that adds to the variable and later assigns the same.
- It is equivalent to a = a + 10.

Assignment operators

=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

Assignment and Equality

- A single equal mark is used to assign a value to a variable, whereas two consecutive equal marks is used to check whether 2 expressions give the same value.
- = is an assignment operator
- == is an equality operator
- x=10
- y=20
- z=20
- (x==y) is False because we assigned different values to x and y.
- (y==z) is True because we assign equal values to y and z.

Special operators

- Python language offers some special types of operators like the identity operator or the membership operator.
- **Identity operators**
- is and is not are the identity operators in Python.
- They are used to check if two values (or variables) are located on the same part of the memory.
- Two variables that are equal does not imply that they are identical.
- **Membership operators**
- in and not in are the membership operators in Python.
- They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

- In a dictionary we can only test for presence of key, not the value.

Order of operations

- Exponentiation gets done first, followed by multiplication and division (including // and %).
- addition and subtraction come last.
- In general, if you’re not sure about something, adding parentheses don’t do any harm.

Complex Numbers

- A complex number is a Python numeric type made of real and imaginary parts.
- Python complex number can be created either using direct assignment statement or by using complex () function.
- It is represented as a+bj.
- a=10+5j
- Here, 10 is the real part, and 5j is the imaginary part.
- To denote the irrational part, you can’t use the letter ‘i’.
- It is mandatory to provide a coefficient to the imaginary part.

- Chk (cmplx.py)

Boolean Data Type

- The boolean data type is either True or False.
- In Python, boolean variables are defined by the True and False keywords.
- Integer value ‘0’ is False.
- Any non zero value is considered as True.

- a = True
- type(a)
- <class 'bool'>

- >>> b = False
- >>> type(b)
- <class 'bool'>

Functions

- A function is a block of related statements that performs a specific task, that only runs when it is called.
- Functions help break our program into smaller and modular chunks.
- It avoids repetition and makes the code reusable.
- Python functions return a value using a return statement, if one is specified.
- A function can be called anywhere after the function has been declared.
- By itself, a function does nothing. But, when you need to use a function, you can call it, and the code within the function will be executed.
- In Python, there are two types of functions: user-defined and built-in.

Built-in Functions

- https://docs.python.org/3/library/functions.html

Syntax of Function

- def function_name(parameters):
- """docstring""“ #Good to have a docsting
- statement(s)

Creating a function

- To create a user defined function we use the key word def
- def function_name(parameter): statement
- def wum():

  print(“9986368187")

- **To invoke the function**
- wum()
-
- The def keyword is used to indicate that we want to create a function.
- wum is the name of our function. This must be unique.
- () is where our parameters will be stored.
- : marks the end of the header of our function.
- Then We have to define the body of the function with proper intentation.

Arguments to the function

- We can create a function without any argument. Fun()
- With one argument/non default argument. Fun(x)
- Two or more arguments. Fun (x,y,z)
- With a default/keyword argument. Fun(length=0,width=0)
- Combination of both .Fun(x,length=0)
- When using combination of non default and default arguments while calling a function, non default arguments should be mentioned first.

- Fun(x,length=5) not Fun(length=5,x)

Return Statement.

- The python return statement is used in a function to return something to the caller program.
- We can use the return statement inside a function only.
- In Python, every function returns something. If there are no return statements, then it returns None.
- If the return statement contains an expression, it’s evaluated first and then the value is returned.
- The return statement terminates the function execution.
- A function can have multiple return statements. When any of them is executed, the function terminates.
- A function can return multiple types of values.
- Python function can return multiple values in a single return statement.