01 SUNDAY
121-244 • WK 18 2022 MAY

Python

List	Tuple
① mutable data type	① immutable data type
② List is a container to contain diff. types of object and is used to iterate objects.	② Tuple is also similar to list but contain immutable objects.
③ List iteration is slower	③ Tuple processing is faster than List.
④ List consumes more memory	④ Tuple consumes less memory
⑤ syntax → list = ['a', 'b']	⑤ tuple = ('a', 'b').

What is decorator & explain with example.?

→ Decorator is a funct^n that takes another funct^n as an argument, add some kind of function ality and then returns another funct^n.

All of this without altering the source code of the original funct^n that you passed in.

How to delete a file in python

→ import os
os.remove("demofile.txt") # for file

import os
os.rmdir("myfolder") # for folder

```
def decorator_func(func):
    def wrapper_func():
        print("wrapper_func worked")
        return func()
    print("decorator_func worked")
    return wrapper_func

def show():
    print("show worked")
decorator_show = decorator_func(show)
decorator_show()
```

Output :-

```
decorator_func worked.
wrapper_func worked
show worked
```

String :- String is a sequence of characters.

integer :-

negative indexing :- indexing start from end is negative index

How memory manged in python

→

Memory mangement in python involves a private heap containing all python objects and data structures.
Interpreter takes care of python heap and that the programmer has no access to it

— The allocation of heap space for python objects is done by python memory manager. The core API of python provides some tools for the programmer to code reliable and more robust program.

Python also has a build-in garbage collector which recycles all the unused memory. When an object is no longer referenced by the program, the heap space it occupies can be freed. The garbage collector determines objects which no longer referenced by the program frees the occupied memory and make it available to the heap space

— The gc module defines funct^n's to enable/disable garbage collector.

1. gc.enable() — Enables automatic garbage collector
gc.disable() — disable automatic garbage collector

05
THURSDAY
2022 MAY
125-240 • WK 19

Diff betn Generator and iterator

05 May 2022

Q Generator -:

— Generators are iterators which can execute only once

— Generator uses "yield" keyword.

— Generators are mostly used in loops to generate an iterator by returning all the values in the loop without affecting the iteration of the loop.

— Every generator is an iterator.

Ex -:

```
def sqr(n):
    for i in range(1, n+1):
        yield i*i
a = sqr(3)
print(next(a))
print(next(a))
print(next(a))
```

Output -:

1
4
9

iter method → give it will object of iterator

next :- next value

Iterators :-

An iterator is an object which contains a countable number of values and it is used to iterate over iterable object like list, Tuples, sets etc.

Iterators are used mostly to iterate or convert other objects to an iterator iter() function.

Iterator uses .iter() and next() functions.

- Every iterator is not a generator

Example :-

```
iter_list = iter(['A', 'B', 'C'])
Print(next(iter_list))
Print(next(iter_list))
Print(next(iter_list))
```

Output

A
B
C

_Ars 06-june

minimum one argument

8) __init__() function-:

~~The __init__()~~

The __init__ method is similar to constructors in C++ and java. Constructors are used to initialize the object's state.

Ex-:

```
# A sample class with init method

class Person:
    # init method or constructor
        def __init__(self, name):
                self.name = name

    # sample Method.
        def say_hi(self):
                print("Hello, my name is", self.name)

P = Person("Nitin")
P.say_hi()
```

Output:
Hello, my name is Nitin

Ans 06-June

(2) ~~__init__()~~ function :- minimum one argument

~~The init()~~

The __init__ method is similar to constructors in C++ and java. Constructors are used to initialize the object's state.

Ex:-

```
# A sample class with init method.

Class Person:
    # init method or constructor
    def __init__(self, name):
        self.name = name

    # Sample Method.
    def say_hi(self):
        print("Hello, my name is", self.name)

P = Person("Nitin")
P.say_hi()
```

Output :-
Hello, my name is Nitin

09 MONDAY
129-236 • WK 20 2022 MAY

diff betn module and package

05 May 2022

06 Jun 2022

Modules :-

The modules is a simple python file that contains collections of functions and global variables and with having a .py extension file. It is an executable file and to organize all the modules we have the concept called Package in python.

A module is a single file (or files) that are imported under one import and used.

Eg :- import <my_module>
import numpy.

Packages :-

The packages is a simple directory having collections of modules. This directory contains python modules and also having __init__.py file by which the interpreter interprets it as a package. The package is simply a namespace. The package also contains sub-packages inside it.

A package is a collection of modules in directories that give a package hierarchy.

Eg :- from my-package.abc import a.

05 May 2022

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

diff betn range() and xrange()?

TUESDAY MAY 2022

10

130-235 • WK 20

~~struct~~ is

Parameters	Range()	Xrange()
Return types	It returns a list of integers	It returns a generator objects.
Memory consumptn	Since range() returns a list of elements, it takes more memory.	In comparision to range(), it takes less memory.
Speed	It execution speed is slower	Its execution speed is faster.
Python versin	Python 2, Python 3	Xrange no longer exists.
Operations	Since it returns a list, all kinds of arithmetic operations can be performed	Such operation cannot be performed on xrange().

Q What are in built data types in python Or Explain mutable and immutable data types in python

11 WEDNESDAY 2022 MAY (131-234 • WK 20)

05 May 2022

Data Type	Mutable / immutable
Boolean (bool)	Immutable
Integer (Int)	Immutable
Float	Immutable
string (str)	Immutable
tuple	Immutable
frozenset	Immutable
list	Mutable
set	Mutable
dict	Mutable

A first fundamental distinction that python makes on data is about whether or not the values of an object changes.

If the value can change, the object is called Mutable.

while if the value cannot change the object is called immutable.

— You can access global variables by any statements in the program.

— It is stored on a fixed location decided by the ~~compiler~~ compiler.

06 Jun 2022

Q

Local variab

- It is declar
- If it is no
- It is creat
dnt lost w
- Data sharin
variable co
- Parameters pa
access the va
- when the va
funct^n, the
Local variab
statements,
declared.
- It is store

Global variable

- It is declared
- If it is not i
- It is created
starts and lost
- Data sharing
same global
- Parameters possi
as it is visible
- When the value
funct^n changes

05 May 2022

06 Jun 2022

Diff betn Local and global variable

THURSDAY MAY 2022

12

132-233 • WK 20

Q

→ Local variable :-

- It is declared inside a function.
- If it is not initialized, a garbage value is stored.
- It is created when the function starts execution and lost when the function terminates.
- Data sharing is not possible as data of the local variable can be accessed by only one function.
- Parameters passing is required for local variable to access the value in other ~~format~~ function.
- When the value of local variable is modified in one funct^n, the change are not visible in another funct^n.
- Local variables can be accessed with the help of statements, inside a function in which they are declared.
- It is stored on the stack unless specified.

Global variable :-

- It is declared outside the funct^n.
- If it is not initialized zero is stored as a default.
- It is created before the program's global execution starts and lost when the program terminates.
- Data sharing is possible as multiple funct^n can access the same global variable.
- Parameters passing is not necessary for a global variable as it is visible throughout the program.
- When the value of global variable is modified in one funct^n changes are visible in the rest of the program.

13 FRIDAY 2022 MAY
233-232 • WK 20

05 May 2022

Explain Break, Continue and Pass Statement

Break :-

A break statement, when used inside the loop, will terminate the loop exit. It used inside nested loops, it will break out from the current loop.

Continue :-

A continue statement will stop the current execution when used inside a loop, and the control will go back to start of the loop.

Pass :-

A pass statement is a null statement. When the python interpreter comes across the pass statement, it does nothing and is ignored.

Break

```
for i in range (10)
  if i == 7:
     break
  Print (i, end=" ")
```

Output :-
0, 1, 2, 3, 4, 5, 6

Continue

```
for i in rangl(10):
  if i == 7 :
     Continue .
  Print (i, end=" ")
```

Output :-
0, 1, 2, 3, 4, 5, 6, 8, 9

Pass

```
def my_fun():
  Print ('pass function inside')
  Pass
my_func()
```

Output
pass inside function

06 Jun 2022

Self :-

The 'self' parameter of the class, belong to the

```
Class Person:
  def __init__
    self.nam
    self.ag
  def.info (self)
    print (f" my
c = Person ("
c = info ()
```

Output :
My name

self represen
it used to acces

05 May 2022

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

What is 'self' keyword in Python

SATURDAY
MAY 2022

14

134-231 • WK 20

*Self :-

The 'self' parameter is refrenced to the current instance of the class, and is used to access variables that belong to the class.

```
Class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age.

def.info(self):
    print(f"My name is {self.name}. I am {self.age}
                    years old.")

c = Person("Nitin", 23)
c = info()
```

Output:

My name is Nitin. I am 23 years old.

self represent the instance of a class and it used to access the variable of class.

15 SUNDAY
135-230 • WK 20 2022 MAY

Diff betn Pickling and unpickling

M	T	W	T	F	S	S	05 May 2022
30	31					1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	

Pickling :-

In python, the pickle module accepts any python object, transforms it into a string representation, and dumps it into a file by using the dump funct°. This process is known as pickling. The function used for this process is pickle.dump()

Unpickling :-

The process of retrieving the original python object from the stored string representation is called unpickling. The funct° used for this process is pickle.load.

- They are inverse of each other.
- Pickling, also called serialization, involves converting a python object into a series of bytes which can be written out to a file.
- Unpickling, or de-serialization, does the opposite - it converts a series of bytes into the python object it represents.

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

When you are not sure how many arguments you need to pass to a function, you can use *args and **kwargs to allow the function to accept any number of arguments.

The *args keyword is used to pass a variable number of arguments to a function. It is commonly used to pass a variable number of arguments to a function when the function is called with multiple arguments.

The **kwargs keyword is used to pass a variable number of keyword arguments to a function. The function can access each keyword argument by the name associated with it.

*args Python Example

```
def sum(*args):
    total = 0
    for a in args:
        total += a
    print(total)

sum(1,2,3,4,5)
```

output = 15

05 May 2022

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

what does *args and **kwargs mean? Eg

MONDAY
MAY 2022

16
136-229 • WK 21

When you are not clear how many arguments you need to pass to a particular functn, then we use *args and **kwargs.

The *args keyword represents a varied number of arguments - It is used to add together the values of multiple arguments.

The **kwargs keyword represents an arbitary number of arguments that are passed to a functn.
**kwargs keywords are stored in dictionary - You can access each item by refering to the keyword you associated with an argument when you passed the argument

*args python Example

```
def sum (*args):
    total = 0
    for a in args:
        total = total + a
    print (total)

sum(1,2,3,4,5)
```

output = 15

**kwargs python Explain

```
def show (**kwargs)
    print (kwargs)

show (A=1, B=2, C=3)
```

Output :

{'A': 1, 'B': 2, 'C': 3}

plain

17

137-228 • WK 21

TUESDAY

2022 MAY

Q What is "open" and "with" statement in python?

M	T	W	T	F	S	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

05 May 2022

M	T	W	T	F
		1	2	3
6	7	8	9	10
13	14	15	16	17
20	21	22	23	24
27	28	29	30	

06 Jun 2022

- Both statements are used in case of file handling.

- With the "with" statement, you get better syntax and exception handling.

```
f = open ("nition.txt")
content = f.read()
print (content)
f.close().
```

```
with open ("nition.txt") as f:
    content = f.read().
    print (content).
```

~~Text note ("+");~~

- Read only
  Open
  at the be
  raises
  in which

- Read and
  - Open
  - Raises

- Write only
  - Open
  - for
    over w
  - ~~The~~
    if the

- Create and
  - Opens
  - for exis
  -

- Append only
  - Open the
  - The file is
  - The data b
    end, after

Q Diff- ways to read and write in a file in python?

- Read only ('r'):
  Open text file for reading. The handle is positioned at the begining of the file. If the file does not exists, raises I/O errors. This is also the default mode in which file is opened.

- Read and write ('r+'):
  - Open the file for reading and writing
  - Raises I/O error if the file does not exist.

- Write only ('w'):
  - Open the files for writing
  - for existing file, the data is truncated and over written.
  - ~~The handle is positioned~~ Creates the file if the files does not exists.

- Write and read ('w+')
  - Opens the file for reading and writing.
  - for existing file, data is truncated and overwritten
  -

- Append Only ('a'):
  - Open the file for writing.
  - The file is created if does not exists.
  - The data being written will be inserted at the end, after the existing data

19 THURSDAY 2022 MAY

05 May 2022

06 Jun 2022

Q what is Pythonpath &

Pythonpath is an environment variables which you can set to add additional directories where python will look for modules and packages.

The 'PYTHONPATH' variable holds a strings with the name of various directories that need to be added to the sys.path directory list by python.

The primary use of this variable is to allow users to import modules that are not made installed yet.

Q How exception handled in python &

→ Try: This blocke will test the exceptional error to occur.

Except: Here you can handle the error.

Else: If there is no exception then this block will be executed.

Finally: Finally block always get executed either exception is genereated or Not

```
try :
    # some co
except :
    # Optional
    # Handling
else :
    # some
    # execute
finally :
    # some
```

Q Where python is

→

- Web Application
- Desktop Applica
- Database Appli
- Networking App
- Machine Lear
- Artifical Inte
- Data Analys
- IoT Applications
- Games and m

```
try :
# some code ... !

except :
    # Optional Block
    # Handling of exception (if required)

else :
    # some code ....
    # executed if no exception.

finally :
    # some code .... (always executed)
```

Q. Where python is used
→

- Web Applications
- Desktop Applications
- Database Applications
- Networking Applications.
- Machine Learning
- Artificial Intelligence
- Data Analysis
- IoT Applications
- Games and many more .... !

21
SATURDAY
2022 MAY
141-224 • WK 21

* How to use f string and format or replacement operator ?

05 May 2022

06 Jun 2022

* How to use f string.

```
name = "sanket"
role = "software engg."
Print(f"Hello, my name is {name} and I am {role} !)
```

Output :-
Hello, my name is sanket and I am software engg.

* How to use format or replacement operator ?

```
Print("Hello, my name is {} and I'm {}".format(name, role))
```

Output :-

Hello, my name is sanket and I'm Software engg.

Abs

1) Abstract
digit de

2) It is imp
unnecessa
withdrau

3) It high
of an obj
the obje

4) It focus
ing for e
car.

5) It is sy
the interf
abstract

In a nut
hiding im
the help
an abstr

Abstraction	Encapsulation
Abstraction works on the design levels.	Encapsulation works on the application level.
It is implemented to hide unnecessary data and withdrawing relevant data.	It is the mechanism of hiding the code and the data together from the outside world or misuse
It highlights what the work of an object instead of how the object works is.	It focuses on inner details of how the objects works, modifications can be done later to the settings.
It focuses on outside viewing, for example, shifting the car.	It focuses on internal working or inner viewing, for example, the production of the car.
It is supported in java with the interface and the abstract class.	Encapsulation is supported using eg. public, private and secure access modification system.
In a nutshell, abstraction is hiding implementation with the help of an interface and an abstract class.	In a nutshell, encapsulation is hiding the data with the help of getters and setters.

05 May 2022

M	T	W	T	F	S	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

23 MONDAY 2022 MAY

143-222 • WK 22

Q Diffrent betn .py and .pyc

→ .py files contain the source code of a program. whereas .pyc file contain the bytecode of your program.

Python compiles the .py files and saves it as .pyc files, so it can refrence them in subsequent invocations.

The .pyc contain the compiled bytecode of python source files. This code is then executed by python's virtual machine

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Q Can you con
is it possible &

t_1 = (1,
t_2 = (

$t_1 = t_1 + t$
Print (" After

Output :-
After : (

Q Can you concatenate two tuples. If yes, How is it possible & since it is immutable

```
t1 = (1, 2, 3)
t2 = (7, 9, 10)

t1 = t1 + t2
Print ("After concatenation is :", t1)
```

Output :-

After concatenation is : (1, 2, 3, 7, 9, 10)

25 WEDNESDAY
145-220 • WK 22 2022 MAY

05 May 2022

06 Jun 2022

List	Array.
1] This list can store the value of diff. data types	It can only consist of value of same type.
2] The list cannot handle the direct arithmetic operations	It can directly handle arithmetic operations.
3] The lists are build-in data structure so we don't need to import it	We need to import array before work with array.
4] The lists are less compatible than array to store the data	An array are much compat to the list.
5] It consumes large memory	It is more compact in memory size comparatively list.
6] It is suitable for storing the longer sequence of the data items	It is suitable for storing shorter sequence of data items
7] We can print the entire list using explict looping	We can print the entire list without using explict looping.

Q How to rea

→ By using

```
x = list(ra
print("Li
```

```
x = [int(x)
print("Num
```

```
x = [int(
print("Num
```

Q How to co

→

* Delete by

```
d1 = {'
d1.clea
print(d1)
```

* Delete by

```
d1 = {'
d1.pop('
print(d1)
```

* Delete by

```
del d1
print
```

Q How to read multiple values from single Input?

→ By using split()

```
x = list(map(int, input("Enter multiple value: ").split()))
print("List of values: ", x)
```

```
x = [int(x) for x in input("Enter multiple values: ").split()]
print("Number of list is: ", x)
```

```
x = [int(x) for x in input("Enter multiple value: ").split(",")]
print("Number of list is: ", x).
```

Q How to copy and Delete a dictionary.

→

* Delete by using clear():

```
d1 = {'A': 1, 'B': 2, 'C': 3}
d1.clear()
print(d1)
```

* Delete by using pop():

```
d1 = {'A': 1, 'B': 2, 'C': 3}
d1.pop('A')
print(d1)
```

→ {'B': 2, 'C': 3}

* Delete by using del():

```
d1 = {'A': 1, 'B': 2, 'C': 3}
del d1['B']
print(d1)
```

= {'A': 1, 'C': 3}

27 FRIDAY 2022 MAY
147-218 • Wk 22

Q. Diff b/n Anonymous and Lambda function.?

Lambda function :-

- It can have any number of arguments but only one expression.
- The expression is evaluated and returned.
- Lambda function can be used wherever function objects are required.

Anonymous function :-

- In python, Anonymous function is a function that is defined without a name.
- While normal function are defined using the def keyword, Anonymous functions are defined using lambda keyword.
- Hence, anonymous functions are also called lambda function.

Lambda -ex-:

```
adder = lambda x, y : x+y
print (adder(4,8)).
                    = 12
```

05 May 2022

M	T	W	T	F	S	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Q. What is Init

- --init-- i
python.
- The --init-
object is cre
- All classes h
which is also
initiated.
- Use the --in
object pro
- The --init-
but only initi

Q. What is slic

Python slice c
elements from
used with string

Syntax -: slice
slice

Q. What is Init function in python ?

- --init-- is one of the reserved method in python.
- The --init-- method can be called when an object is created from the class.
- All classes have a function called --init--(), which is always executed when the class is being initiated.
- Use the --init--() function to assign values to object properties.
- The --init-- method doesn't create the object but only initalize the objects attributes.

Q. What is slice / slicing in python ? (funct^n)

Python slice() functⁿ is used to get a slice of elements from collection of elements - slice() functⁿ used with string, list, tuple, set, bytes or range objects

Syntax :- slice(stop)
slice(start, stop, step)

29 SUNDAY
149-216 • WK 22 2022 MAY

05 May 2022

M	T	W	T	F	S	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

06 Jun 2022

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Q Is the python case sensetive language.

→ Yes, Python is case sensetive language. Because it differentiates the lower case and upper case identifiers.

Q. Is python interpreted language?

→ A python is an interpreted language.

Q What is scope in python?

Scope is a block of code where objects in python remain relevant.

Type of scope

Local :- A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.

Global :-

A variable created outside of a function is global and can be used by anyone.

```
local scope :-
def my_func():
    x = 300
    print(x)
my_func().

Global scope :-
def x = 300
def my_func()
    print(x)
my_func()
print(x)
```

Q Is Indentation

Indentation specifies block

Indentation python. bec python code, and the code

python uses however, the minimum of

Q What is nam

→ A namespac sure that na conflicts.

Is Indentation required in python?

Indentation is most important for python. It specifies block of code. All.

Indentation is a very important concept of python. because without proper indenting the python code, you will end up seeing Indentation Error. and the code will not get compiled..

Python uses 4 spaces as indentation by default. However, the number of spaces is up to you, but a minimum of 1 space has to be used.

What is namespaces in python.

A namespaces is naming system used to make sure that names are unique to avoid naming conflicts.

31 TUESDAY
151/214 • WK 23 2022 MAY

M T W T F S S
30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

05 May 2022

Action Plan

MON 6
TUE 7
WED 1 8
THU 2 9
FRI 3 10
SAT 4 11
SUN 5 12

① what is PEP 8 and why is it important?

PEP stands for python enhancement proposal. A PEP is an official design document which provides information to the python community, or describing a new feature for python or its processes.

PEP 8 is especially important since it documents the style guidlines for python code. Apparantly contributing to python open-source community require you to follow these style guidlines sincerly and strictly.

② what are attributes in python? { methods

A method is a function which is defined in the class. An attribute is an instance variable defined in the class

eg:

```
class Example (object):
    def __init__(self, name):
        self.name = name.
    def hello (self):
        print "Hi, I am" + self.name.
```

Here hello is a method, and name is an attributes

Definition

Attributes = A value name using dotted attributes a it wo

method = A functi

05 May 2022

Action Plan

ACTION PLAN
JUNE 2022

06
MONTH IN VIEW

MON		6	13	20	27
TUE		7	14	21	28
WED	1	8	15	22	29
THU	2	9	16	23	30
FRI	3	10	17	24	
SAT	4	11	18	25	
SUN	5	12	19	26	

JUNE

OBJECTIVES OF THE MONTH

Defination

Attributes = A value associated with an object which is referenced by name using dotted expressions. for ex:- if an object o has an attribute a it would be referenced as o.a.

method = A function which is defined inside ~~class body~~ a class.

01 WEDNESDAY
152-213 • WK 23 2022 JUNE

M T W T F S S 06 Jun 2022
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

07 Jul 2022

Q) What is docstring in python ?

Documentation string or docstring is a multiline string used to document a specific code segment.

The docstring should describe what the function or method does.

Q) How are arguments passed by value or by reference in python ?

→ Pass by ~~reference~~ value :-

It is used in some programming language where values to the argument of the function are passed by refrence which means that the address of the variable is passed and then the operation is done on the value stored at these address.

Pass by ~~reference~~ value :-

It means that the values is directly passed as the value to the argument of the function. Here, the operation is done on the value and then the value is stored at the address. Pass by value is used for a copy of the variable

07 Jul 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

THURSDAY
JUNE 2022

02
153-212 • WK 23

Diff btn -

Call by refrence

While calling a Functn, in a programming language instead of copying the values of variables, the address of the variable is used, it is known as "call by refrence."

It is method, a variable itself is passed.

Change in the variable also affects the value of the variable outside the functn

Allows you to make changes in the values of variables by using functn calls.

The original value is modified

call by value.

While calling a functn, when we pass values by copying variables, it is known as "call by values".

A copy of the variable is passed in a call by value.

Changes made in a copy of a variable never modify the value of the variable outside the functn.

Does not allow you to make any changes in the actual variables

Original value is not modified.

03 FRIDAY
154-211 • WK 23 2022 JUNE

13 14 15 16 17 18 19 Jun
20 21 22 23 24 25 26 2022
27 28 29 30

* decorators example...:

When we need to change the behaviour of a funcn without modifying the functn itself.

Ex-:

```
def div (a, b):
    print (a/b)          (divide)


def smart_div (func)          :.' decorator funct°
    def inner(a, b):
        if a<b:
            a, b = b, a
        return func (a, b)
    return inner          -: returns inner func


div1 = smart_div (div)
div(2, 4)
```

06 Jun 2022

07 Jul 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SATURDAY
JUNE 2022

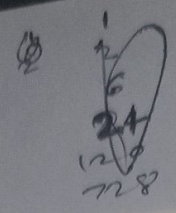
04
155-210 • WK 23

* Program to check leap year

```
year = int(input())

if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".format(year))
        else:
            print("{0} is not a leap year".format(year))
    else:
        print("{0} is a leap year".format(year))
else:
    print("{0} is not a leap year".format(year))
```

05 SUNDAY
156-209 • WK 23
2022 JUNE



M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

06 Jun 2022

07 Jul 2022

M	T
4	5
11	12
18	19
25	26

Program to check number is prime.

```
num = int(input())
flag = False  # define flag variable.
if num > 1:  # Prime number is > 1
    for i in range(2, num):  # if factor is found, set flg to true
        if (num % i) == 0:
            flag = True
            break  # break out of loop.
if flag:
    print(num, "is not a prime number")
else:
    print(num, "is a prime number").
```

1+1+

06 Jun 2022

07 Jul 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

fibonacci series

MONDAY
JUNE 2022

06

157-208 • WK 24

```
nterms = int(input())

# first two terms
n1, n2 = 0, 1
count = 0

# check number is valid
if nterms <= 0:
    print("Please enter a positive number")
elif nterms == 1
    print("Fibonacci sequence upto", nterms, ":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1
```

07
TUESDAY
2022 JUNE
158-207 • WK 24

153 = $1^3 + 5^3 + 3^3 = 1 + 125 + 27$

Armstrong number
↓
sum of cube of its number is equal to itself

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

06 Jun 2022

07 Jul 2022

```
num = int(input())
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10

if num == sum:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an armstrong number")
```

string is palindrom or not

07 Jul 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

WEDNESDAY
JUNE 2022

08

159-206 • WK 24

```
my_str = 'aIboRphoBiA'
# make it suitable for careless comparison
my_str = my_str.casefold()

# reverse the string.
rev_str = reversed(my_str)

# check if the string is equal to it reverse.
if list(my_str) == list(rev_str):
    print("The string is palindrom")
else:
    print("The string is not palindrome")
```

count number of vowels

09 THURSDAY 160-205 • WK 24
2022 JUNE

count vowels of string.

M T W T F S S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

06 Jun 2022

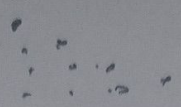
```
string = input ("Enter the string : ")
count = 0
string = string.lower()
for i in string :
    if i == 'a' or i == 'e' or i == 'i'
        or i == 'o' or i == 'u' :
        # if True
            count += 1
#check if any vowels found
if count = 0 :
    print ('NO vowels found')
else :
    print ("Total vowels are : " + str(count))
```

output -:

Enter the string : Sanket

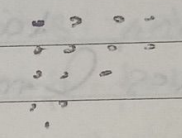
Output Total vowels : 2

Half pyramid



```
rows = 5
for i in range (0, rows):
    for j in range (0, i+1):
        print ("*", end = ' ')
    print ("\r")
```

Downward half pyramid



```
rows = 5
for i in range (rows+1, 0, -1):
    for j in range (0, i-1):
        print ("*", end = ' ')
    print ("\r")
```

11 SATURDAY 2022 JUNE

162-203 • WK 24

06 Jun 2022

07 Jul 2022

What is Access modifier, and its type

Pass in python

Pass is ~~keyword~~ ~~represents~~ a null operator in python. It is generally.

Access modifier -:

Access modifier are keywords in object oriented programing language that set the accessesibility of classes & methods.

(1) Public -: Any member declared as public can be accessed from outside the class through an objects. All members in python class are public by default

(3) Protected -: The data member of the class is declared protected by adding a single underscore (' _ ') and this prevent it from access

(2) Private -: A double underscore " __ " makes the variable private as well as secure. It is not possible to access from outside the class because it will throw an error.

06 Jun 2022

07 Jul 2022

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SUNDAY
JUNE 2022

12

163-202 • WK 24

Q. What are sdlc phases.

The software development life cycle is a structured process that enables the production of high-quality, low-cost software, in the shortest production time.

or

SDLC is a framework that defines activities that are performed during the software development process.

There are 6 phases in SDLC :-

1] Requirement
2] Design
3] Implementation
4] Testing
5] Deployment
6] Maintainance.

13 MONDAY
164-201 • WK 25 2022 JUNE

M T W T F S S
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
06 Jun 2022

07 Jul 2022

Q What is Type casting in python ?

→ There are two types
(i) Implict type casting
(ii) Explict type casting.

Implict type casting :-

Implict type conversion is automatically by the interpreter, without user intervention.
Python automatically converts one data type to another data type

Explict type casting :-

In this type casting / conversion the user or programmer converts the data type of an object to the required data type.
In python we use predefined funcn like int(), float(), str(), bool() etc to perform explict type casting.

13 MONDAY
164-201 • WK 25 2022 JUNE

M T W T F S S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

06 Jun 2022

07 Jul 2022

Q What is Type casting in python ?

→ There are two types

(1) Implict type casting

(2) Explict type casting.

Implict type casting :-

Implict type conversion is automatically by the interpreter, without user intervention.

Python automatically converts one data type to another data type

Explict type casting :-

In this type casting / conversion the user or programmer converts the data type of an object to the required data type.

In python we use predefined functions like int(), float(), str(), bool() etc to perform explict type casting.