Python Language

Markdown Shared Library

    - assert error check
    - Raise Except
    - catch Except avoid termination
    - Create my own Error Type
    - pass by reference
    - function annotation
- Unit Test
- Regular Expression
- Logging
- algorithms
- Class
    - Design
    - Abstraction
    - class basic
    - use type to create class
    - Inheritance
    - Polymorphism
    - dunder functions
    - attribute scope
    - class tricks
    - class inheritance
    - Python Interface
    - Unit Test
    - Dunder Variables
    - Global Variables
- Blackjack Game
    - Blackjack Rules
- Physics Unit
- Blackjack Card Game
    - Object relationship
    - Game logic
    - Code Optimization
    - missing unit tests
    - Integration Test
    - Documentation
    - Software development life cycle
- Deployment
- Yahtzee Dice Game

- File Access
- nmpy
- pandas
- Data Analysis for Excel Users
- Plot
- Pandas
- Clean Code
  - Design Principles SOLID
- Turtle
- SQLite
- MongoDB
- tkinter(windows based GUI)
  - open window
  - switch between frame
  - Label, Button, Entry widgets
    - pack attributes
  - pack layout
  - grid layout
  - place layout
  - icon and title
  - Other widgets
  - color
  - mouse
  - table
  - canvas
  - tab window frame
  - titled frame
  - plot chart in tkinter
  - display image
  - popup window
  - card game GUI
  - sqlite DB
  - Web Service API
  - practices
  - Application
- Application Development Process
- PyInstaller
- Data Structure

    - stack
    - queue
    - Priority Queue
    - Linked list
    - doubly linked list
    - binary search
    - insertion sort
    - selection sort
- quick sort
- Functional Programing
    - Function decorator(timer)
    - Lambda expression
    - map() function
    - filter() function
    - reduce() function
    - sort() function
    - zip() function
    - Calculate Square root
    - Non-strict evaluation
    - monad
    - Either
- Design pattern
    - Reactivex design pattern
- 18 modules
    - web development
    - Data Science
- GUI Window Application
- Game

Getting Start

getting formiliar your keyboard

Keyboard

My First python program

hello world
line plot
x,y plot

❓ What is Python?

> Python is computer programming language.
> ✔ Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

print

print

- place holder (%s, %d, %f)
- print with tuple
- formated print: print(f"x={x}")
- Homeworks
  - Math Competition
  - print-01
  - print-02

comment

comment

- single line comment: #
- multiple lines comment: """, '''
  ❓Why I need use comment?
  ✔

Variabls

❓ What is variable?
✔ Contain two parts: 1. variable name; 2. variable value. It is called variable declaration when we assign a value to a variable name.

```
width = 10
name = "John"
```

❓ Why we use variables?
✔ I want use same value on many place, so that I can only change one place for all usage of the variable.

- Variable Naming Rules

1. variable name cannot start with number
2. variable can be combination of letters and numbers _, a~z, A~Z, 0~9, no other special characters
3. don't use reserved keywords as variable name

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

Python Keywords

4. Avoid using existing function name as your variable name.
   otherwise, your python builtins functions no longer works the way you expected.

```
>>> dir(__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'E
'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'list', 'locals', 'map
```

Python Playground

❓ How do I get into python playground?

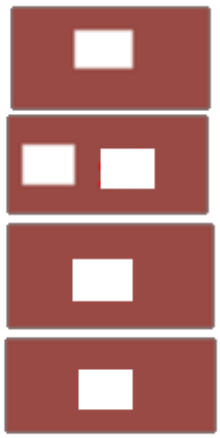
✔ type in **python** in command prompt shown below:

```
C:\Users\12818\workspace\2021fall\python>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

dunder: double underscore (hold shift + -)

```
>>> dir(__builtins__)
```

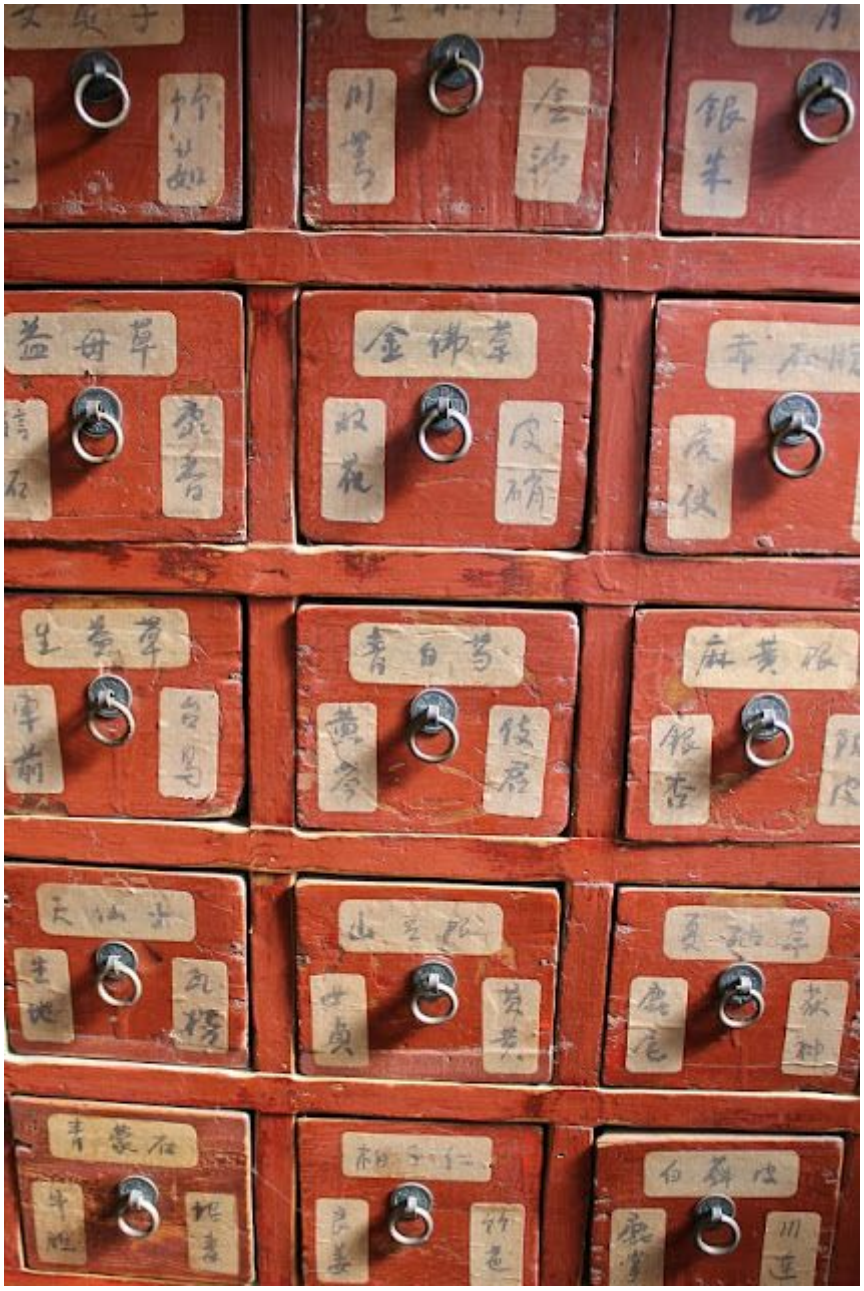
1. can execute any python code, it is good for your code testing.
2. execute multiple line of code.
3. to find document for all imported modules, and builtin functions.

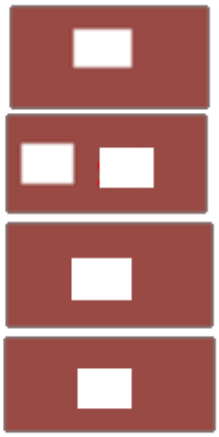


Hello

> you can define more than one variable to point to same memory location.

Variable and memory





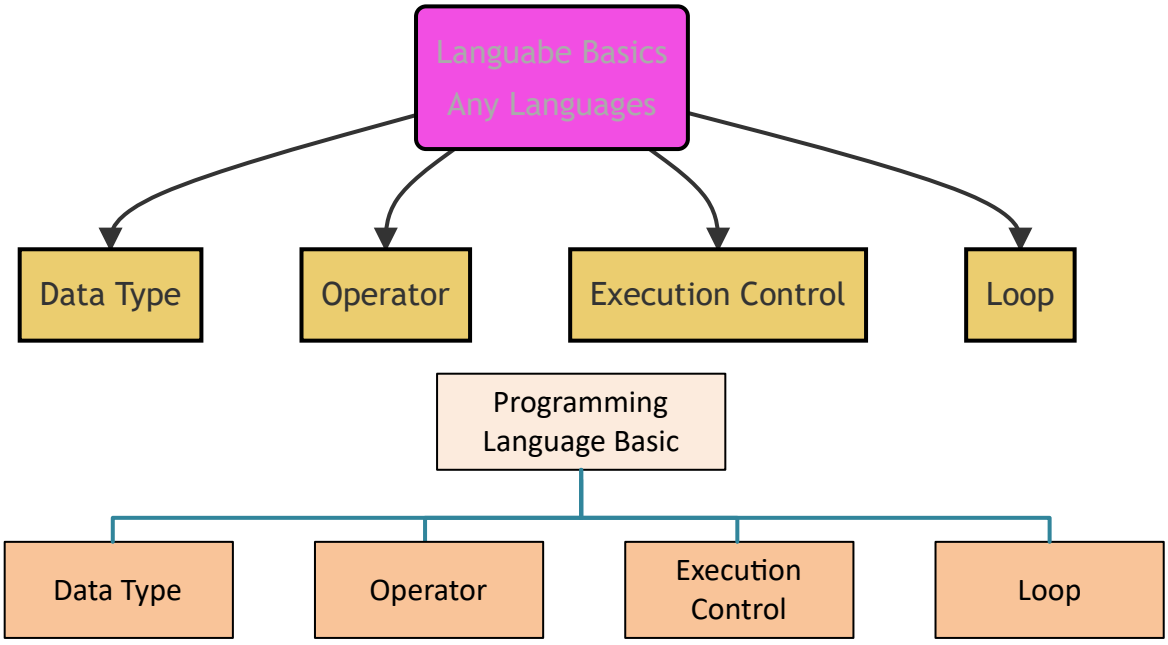
Hello

- Homeworks
  variable-01
  variable-02
  variable-03

Ways to Learn

1. learn from teacher;
2. learn from coding, practice;
3. learn from python document (help on playground)
4. learn from mistakes (your own, others)
5. document (take good note for everything we learn)

Language Basics



Data Type

❓ What is data type in Python?
✅ 1. built ins; 2. developer defined;

- built in data type:

number

Numbers: int, float, complex

❓ What is number in python?

✔ Python supports integers, floating-point numbers and complex numbers. They are defined as int, float, and complex classes in Python.

📌❗ Knowledge Base

> python is smart enough so that you don't need difine a variable type, based on the type of value you assigned to the variable, the variable will be that data type.

```
- int: a=4
- float: a=3.4
- complex: c=4-3j
```

- Homeworks
  number-01

string

- String

❓ What is a str?

✔ str variable is group of letters surounded by ' ', " ", ''' ''', """ """.

- s = 'hello'
- single ' double " are no difference, but must to be pair.
- ""; """ [put your text here] """; ''; ''' [your text here] '''
- string is iterable
- string slicing: start]:[end]:[step
- String operator +, *
- as function str(object)
- string functions (isalnum(), isdigit(), strip(), split(), lower(), upper(), startswith(), endswith(), ...)
- use built-in functions: len(), sorted(), reversed()

- Homeworks
  string-01
  string-02
  string-03

tuple

Tuple

❓ What is a tuple?

✔ tuple is a set of object element, separate by comma, suround by (). it is immutable once you defind.

- (1.5,2,3,4,'hello')
- elements can have different data type in tuple
- we can use len() built in function to get total number of elements in the tuple
- tuple is iterable
- tuple is immutable
- tuple slicing: tuple1start]:[end]:[step
- tupler operator +, *
- as function: tuple(iterable)
- tuple() as functions () can be used as function to convert iterable object to tuple.

list

List

❓ What is a list in python?

✔ a **list** is a special data type in python. A List is an ordered collection of python objects that is iteralbe,mutable, separated by comma, surrounded by [].

- [2, 3, 'hello']
- elements can have different data type in tuple
- we can use len() built in function to get total number of elements in the list
    - list is iterable
    - list is mutable
    - list slicing: list1start]:[end]:[step
    - list operators +, *
    - modify list
    - as function: list(iterable)
    - list functions (append, insert)
- Homeworks

  list-01

  list-02

set

Set

❓ What is set in Python?

✔ A Set is an unordered collection pyhton objects that is iterable, mutable, and separated by comma, surrounded by {}, has no duplicated elements.

- {1,2,3}, set(), set(range(5)), set("hello")
- elements can have different data type in set
    - set is iterable
    - set is mutable
    - set operators: &, |, <, >, ==
    - modify set
    - as function: set(iterable)
    - set functions ()

dict

Dictionary

❓ what is a dictionary data type in Python?

✔ A dictionary consists of a collection of key-value pairs.

It is unordered, iterable, mutable, and each pair separated by comma, surrounded by '{}', and no duplicated key. the key-value pair separated by ':'.

- {1:'Monday', 2:'Tuesday'}
- key-value pair can have different data type in dict
- dict is iterable: only iterate key
- dict cannot be slicing due to unordered key-value pair
- dict operators: **, ==
- dict is mutable: add, update, delete key-value pair
- functions in dict: get(), items(), values(), pop(), ..
- dict() can be used as function to convert iterable object to dict (need to be key-value pair).
- dict modify==>CRUD

date/time

date/time
Get time interval
Date time convertion
Date time detail

📌❗ Knowlodge Base

> the variable value can be reassigned to other data type. Once it is reassigned, the data type changes.

```
i = 5
i = 'hello'
i = [1,2,3,4]
```

Developer defined data type

```
class Student:
  pass

s = Student()
print(type(s))
```

Operator

- operator

- Arithmatic Operator: +; -; *; /: %; **;//(floor divisor)
  arithmatic
- Assignment Operators: =; +=; -=; *=; /=; %=; **=; //=
  assignment
- Comparison Operators: ==, !=, <, >, <=, >=
  comparison
- Identity Operator: is, is not
  identity
- Logical Operator: and, or, not
  logical
- Membership Operator: in, not in
  membership
- Multiple times operator: **
  others
- Ternary operator: if-else, and-or
  ternary
- Bitwise Operator: &, |, ^, <<, >>
  bitwise

$$2^7\times0+2^6\times1+2^5\times0+2^4\times0+2^3\times0+2^2\times0+2^1\times0+2^0\times1=65$$

7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1

> ASCII Table: American Standard Code for Information Interchange

USASCII code chart

$b_7 b_6 b_5$ → / Bits	000	001	010	011	100	101	110	111
b_4 b_3 b_2 b_1 ↓ / Column → / Row ↓	0	1	2	3	4	5	6	7
0 0 0 0	0 NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1 SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2 STX	DC2	"	2	B	R	b	r
0 0 1 1	3 ETX	DC3	#	3	C	S	c	s
0 1 0 0	4 EOT	DC4	$	4	D	T	d	t
0 1 0 1	5 ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6 ACK	SYN	&	6	F	V	f	v
0 1 1 1	7 BEL	ETB	'	7	G	W	g	w
1 0 0 0	8 BS	CAN	(	8	H	X	h	x
1 0 0 1	9 HT	EM	)	9	I	Y	i	y
1 0 1 0	10 LF	SUB	*	:	J	Z	j	z
1 0 1 1	11 VT	ESC	+	;	K	[	k	{
1 1 0 0	12 FF	FS	,	<	L	\	l	|
1 1 0 1	13 CR	GS	-	=	M	]	m	}
1 1 1 0	14 SO	RS	.	>	N	^	n	~
1 1 1 1	15 SI	US	/	?	O	_	o	DEL

ASCII Code

Char.	ASCII	Char.	ASCII	Char.	ASCII
@	64	U	85	j	106
A	65	V	86	k	107
B	66	W	87	l	108
C	67	X	88	m	109
D	68	Y	89	n	110
E	69	Z	90	o	111
F	70	[	91	p	112
G	71	\	92	q	113
H	72	]	93	r	114
I	73	^	94	s	115
J	74	_	95	t	116
K	75	`	96	u	117
L	76	a	97	v	118
M	77	b	98	w	119
N	78	c	99	x	120
O	79	d	100	y	121
P	80	e	101	z	122
Q	81	f	102	{	123
R	82	g	103	|	124
S	83	h	104	}	125
T	84	i	105	~	126

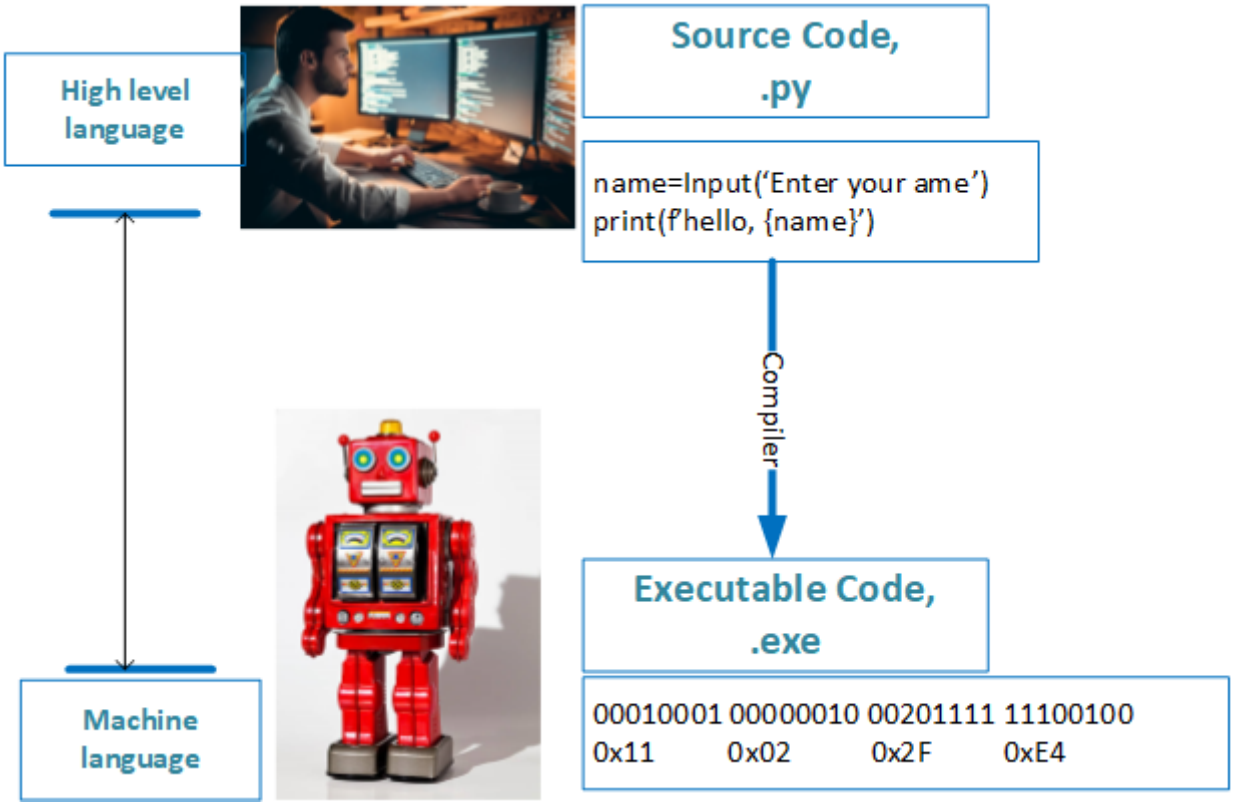
B → 1000010

L → 1101100

U → 1110101

e → 1100101

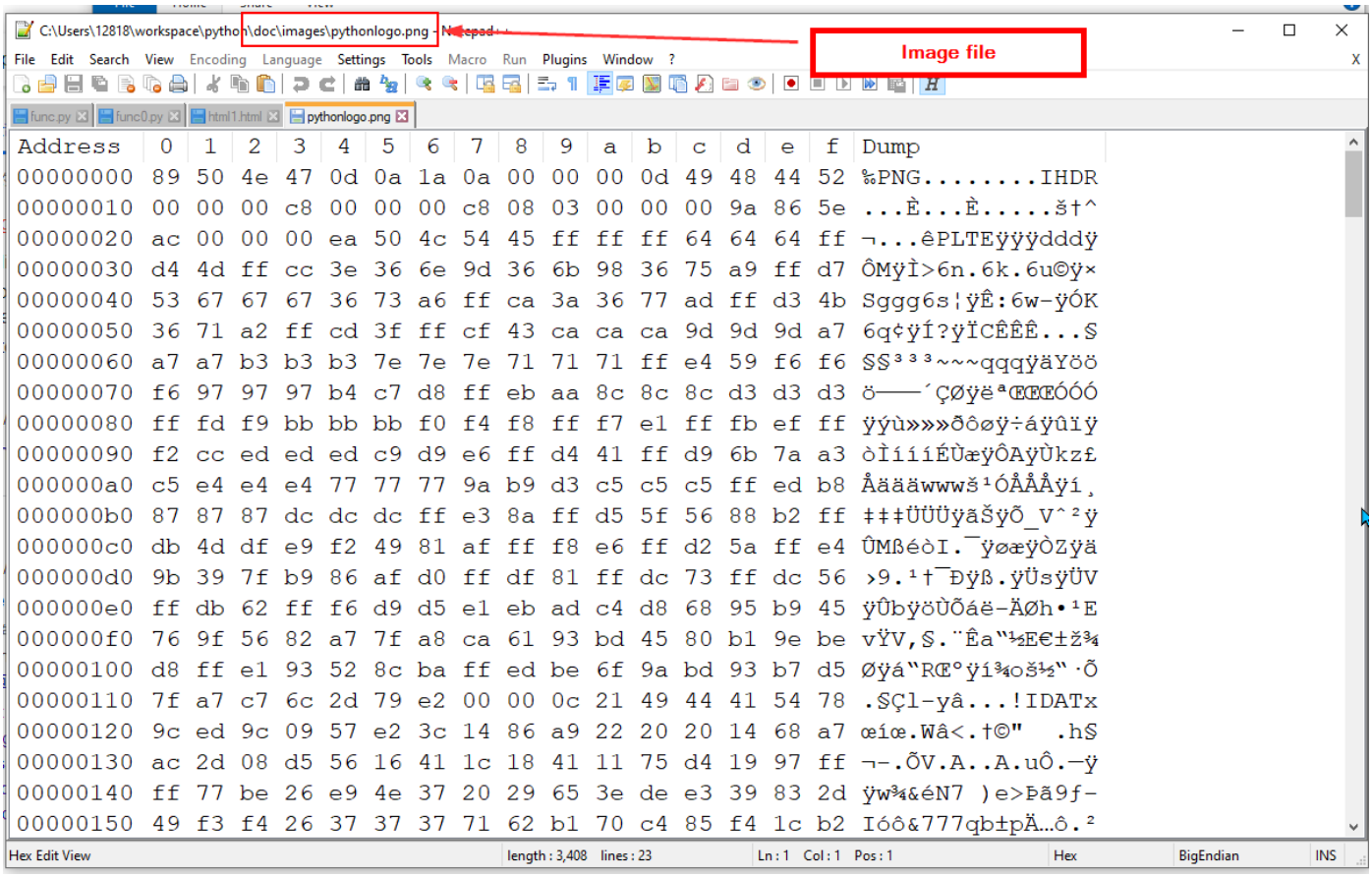
byte for character



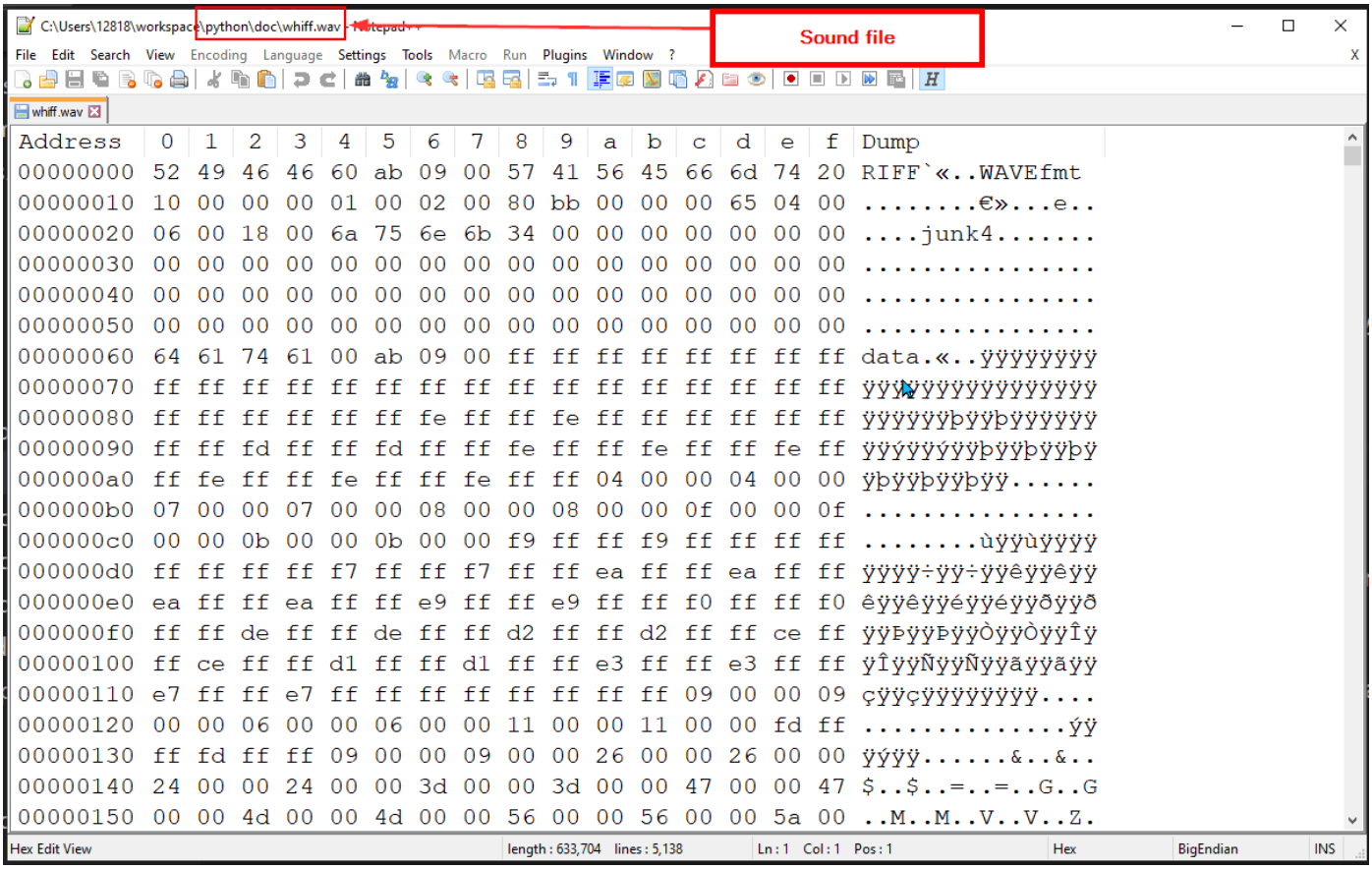
- Understand Image Pixel RGBA mode
-


python™

-



- gunshot sound



- operator precendence: *, /; +, -;

  multiply has higher precendence than +, -, use () to change precendence

- . operator

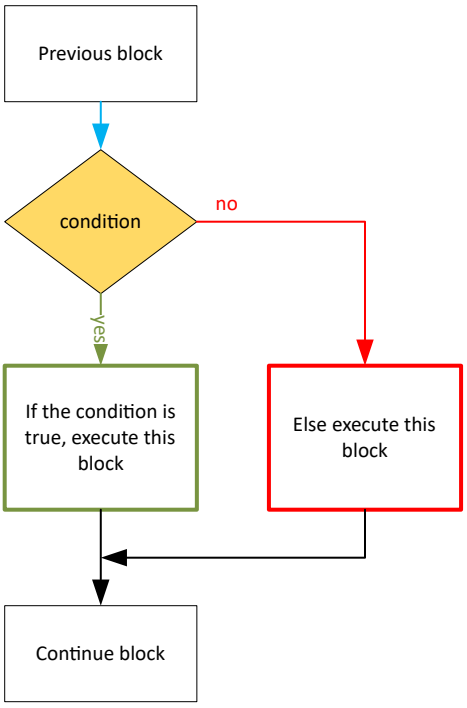
> in OOP, . operator is used to refer to an attribute of class instance.

```
greeting = "Hello"
print(greeting.upper())
```

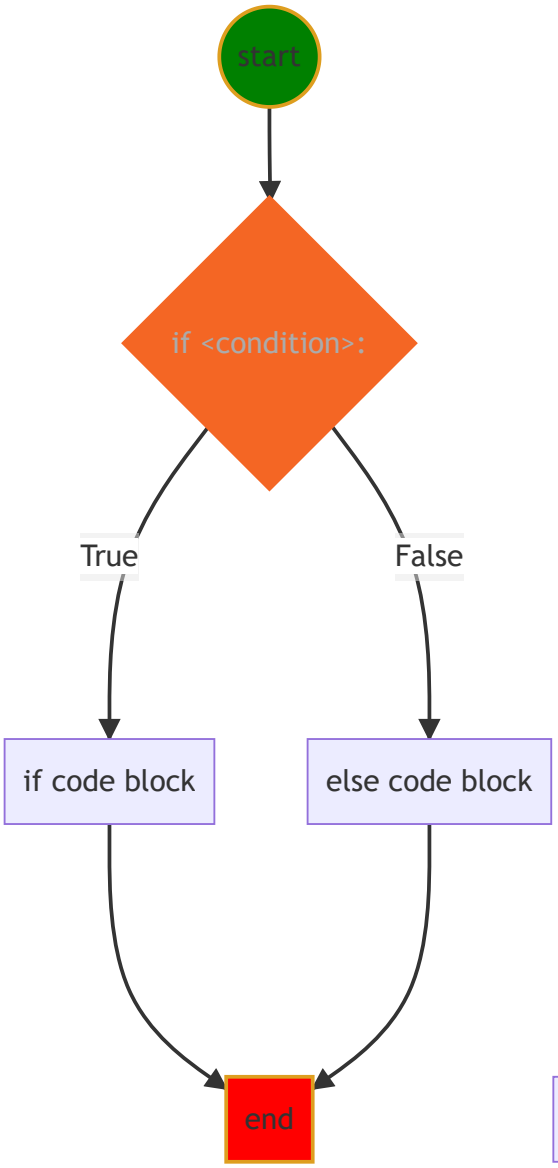
Execution Control

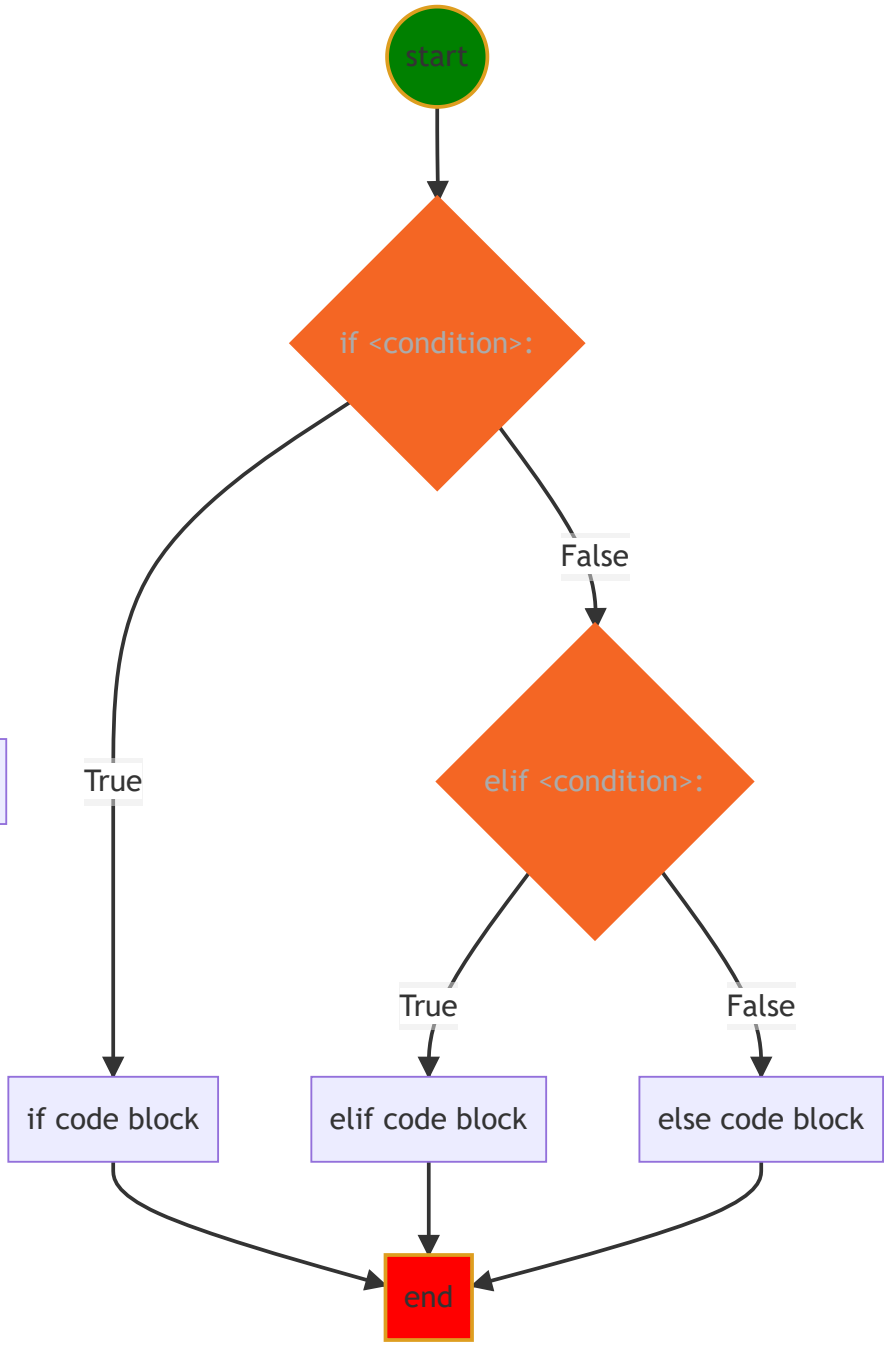
- **if-elif-else** statement Syntax

```
if <condition>:
    # if code block here
elif <condition>:
    # elif code block here
else:
    #else code block here
# code continue ...
```



- Mermaid Diagram for if-else statement





- User input
  - Homeworks

    ifelss-01

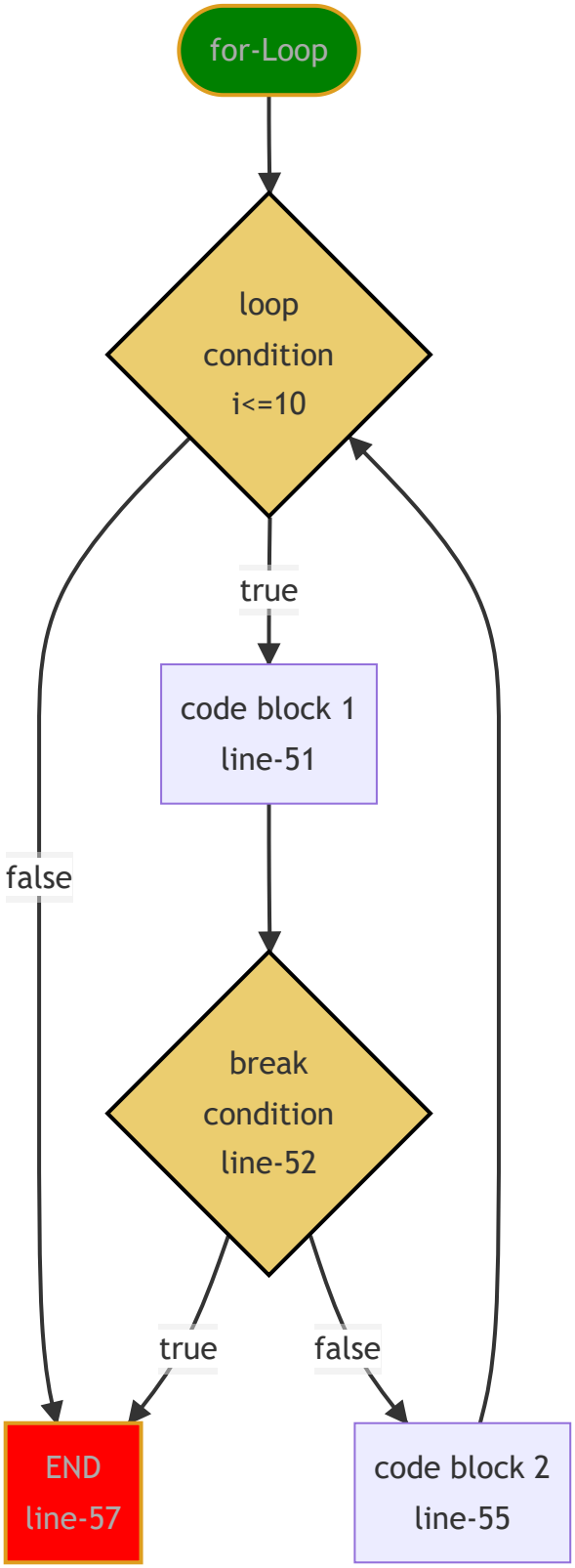
    ifelss-02

    ifelss-03

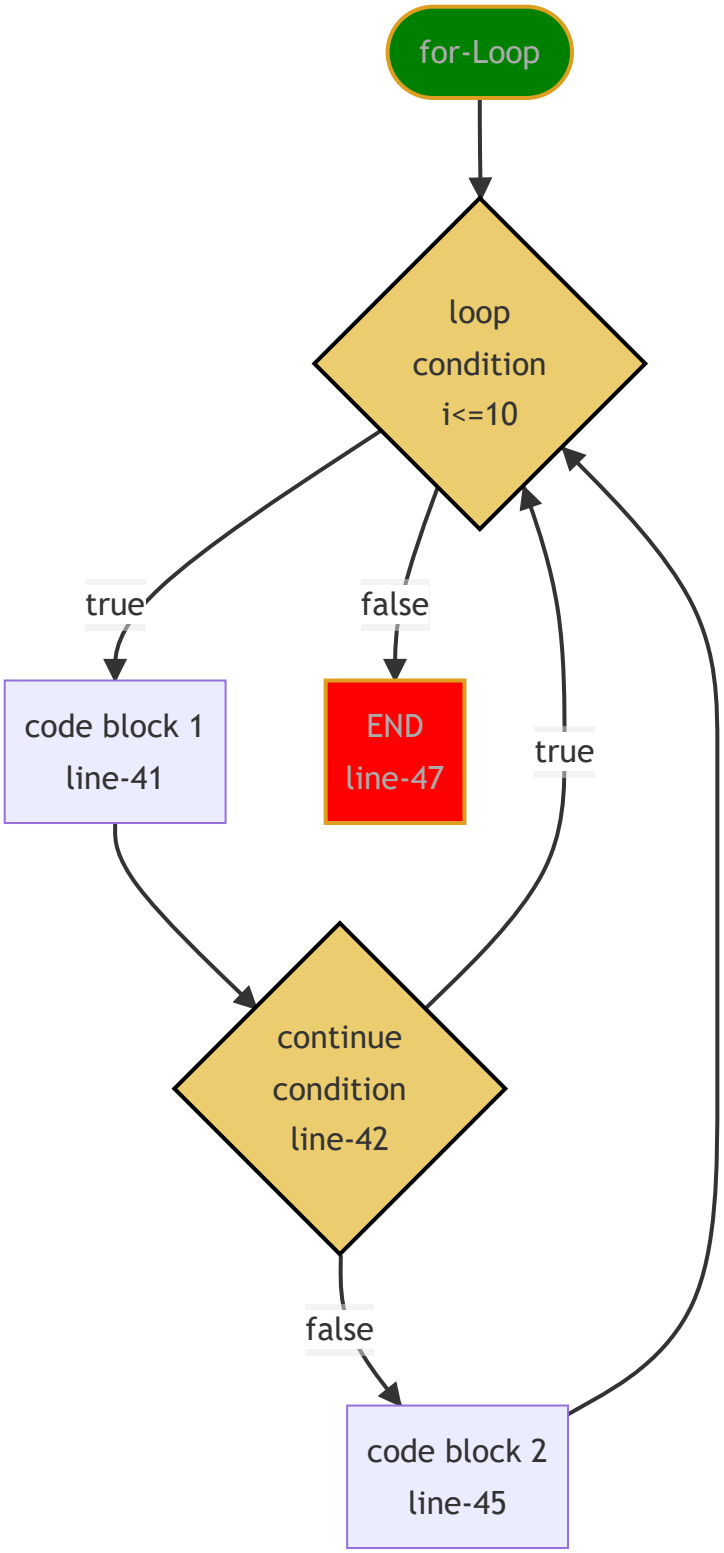
Loop

- for1.py

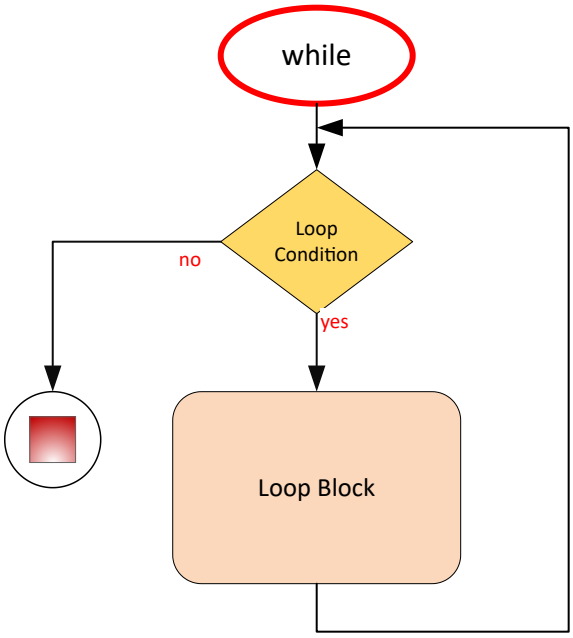
- for2.py
- forBreak.py
- forContinue.py
- Break on for-loop

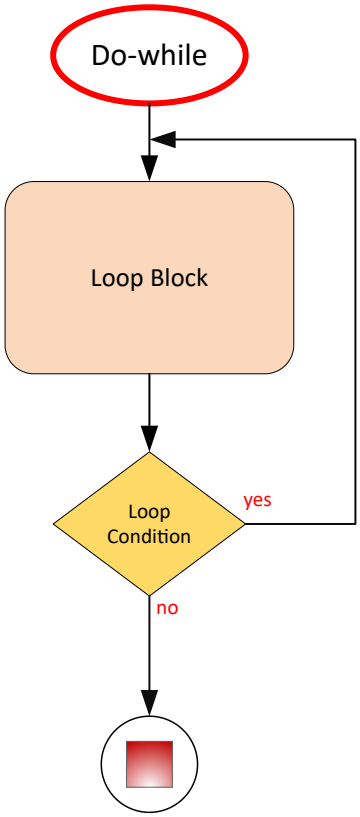


- continue on for-loop


for-Loop
loop
condition
i<=10
true
false
code block 1
line-41
END
line-47
true
continue
condition
line-42
false
code block 2
line-45

- while.py



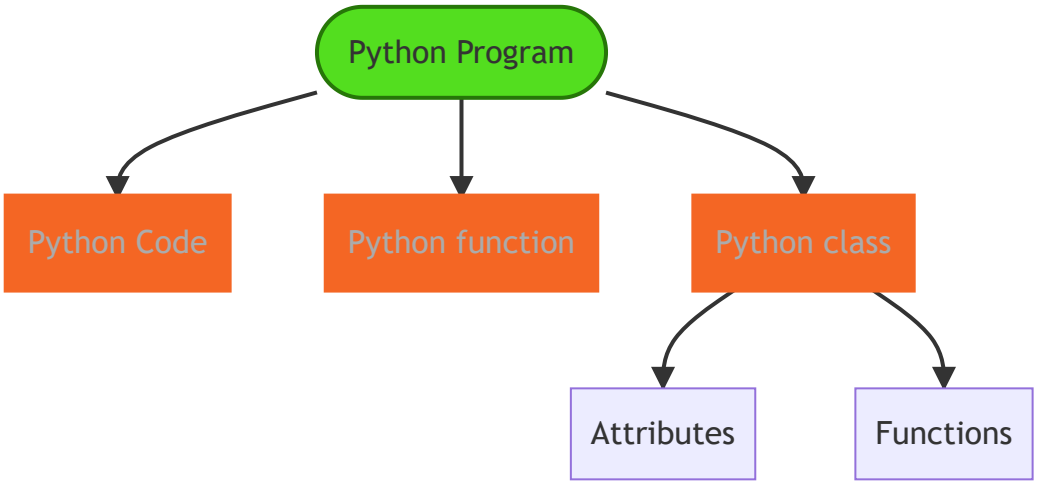


- Homeworks
  - loop-01
  - loop-02
  - loop-03
  - loop-04
  - loop-05

📌❗ Knowlodge Base

> Always adjust the condition variable toward making loop failed.

How to write Python?



direct python code

- guess number game
- roll two Dices
- Probability of rolling dices
- small dice game

Function

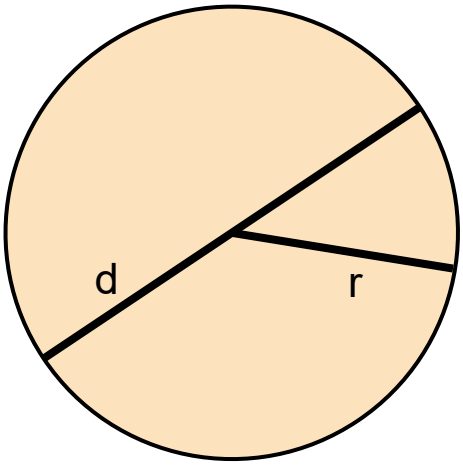
❓What is function?

> ✔A block of code defined by name and arguments, can be used by calling the name many times repeatedly. In Python, functions are first-class citizens. That means functions have the same characteristics as values like other data type such as strings and numbers. Anything you would expect to be able to do with a string or number you can do with a function as well.

- function basic
- define a function
  - def, Python reserved keyword
  - function name, anything you want, but need follow the naming rules
  - (), must have open/close parenthesis pair, no matter it has arguments or not
  - arguments, positional or keyword arguments separated by comma ,

- :, must end with colon
- the function body must indent
- ❗⚡function can be overridden
- 😄return more than one value
- 💡single response
- call a function by function name and (), and arguments if there is any

$$\underbrace{def}_{keyword}\ \underbrace{circle\_area}_{function\ name}\left(\underbrace{a, b, c...}_{positional\ args}\ *\underbrace{e = None, f = 200}_{keyword\ args}\right)\underbrace{:}_{eol}$$



Circle area formula: $A = \pi r^2$

- circleArea
- function arguments
- function parameters
- understand **name**
- understand if **name**=='**main**':
- Python document

```
>>> from src.function.defineFunction import *
>>> help(f)
```

- One time assign default value
- collision.py
- check user input
- access function attribute
- simple math function
- optionalKeywordArgs.py
- optionalPositionalArgs.py

define function in function

- innerFunction01.py
- innerFunction02.py
- define function in another function
- function In Function

return function from function

- dynamicall generated quadratic function

pass function as argument

- passFuncAsArg.py
- 👍function as dictionary value

function decorator

- my_timer.py
- make simple job complecated
- add timer
- add decorator
- add user check on function

global variable

❓ What is global variable?

> ✔global keyword allows you to modify the variable outside of the current scope. It is used to create a global variable and make changes to the variable in a local context.

use global variable

1. When we create a variable inside a function, it is local by default.
2. When we define a variable outside of a function, it is global by default. You don't have to use global keyword.
3. We use global keyword to read and write a global variable inside a function.
4. Use of global keyword outside a function has no effect.
5. It is not necessary to declare global variable outside function

recursive function

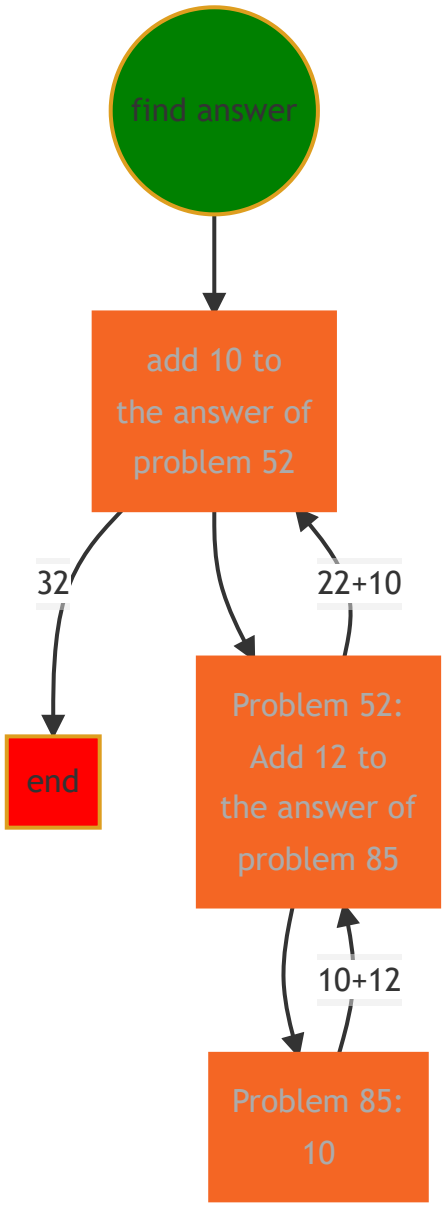
A function is recursive if it calls itself.

1. termination condition.
2. adjust status for each call.
3. Python stops the cunction calls after a depth of 1000 calls.

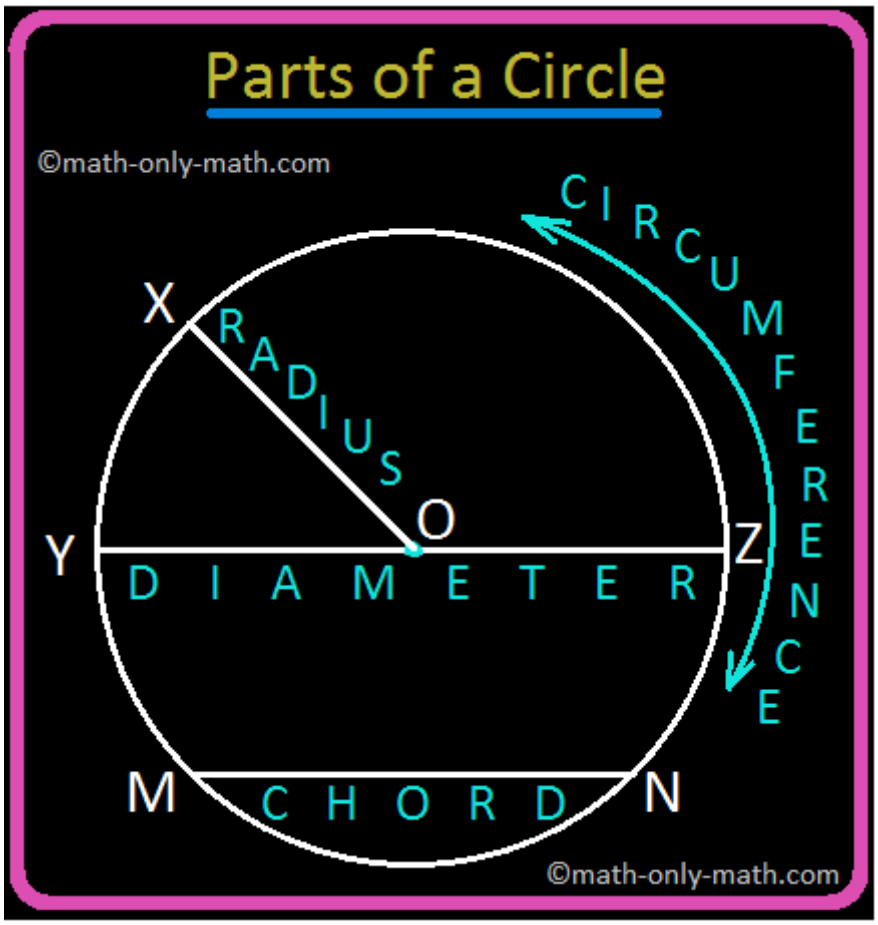
- factoria.py

$$f(n) = n! = n(n-1)(n-2)\cdots 1$$

- recursiveBinarySearch.py
- Understand recursive find.



useful functions



$$d = 2 \cdot r$$

$$c = d \cdot \pi = 2 \cdot r \cdot \pi$$

$$a = \pi r^2$$

where r, d, c, a are radius, diameter, circumference and area respectively.

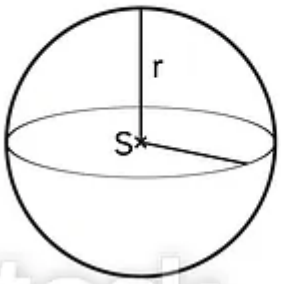
- Circle

SPHERE

A three-dimensional geometric surface
having all of its points the same distance from a given point.

Volume $V=\frac{3}{4}\pi r^2$

Surface area $A=4\pi r^2$



Wireframe

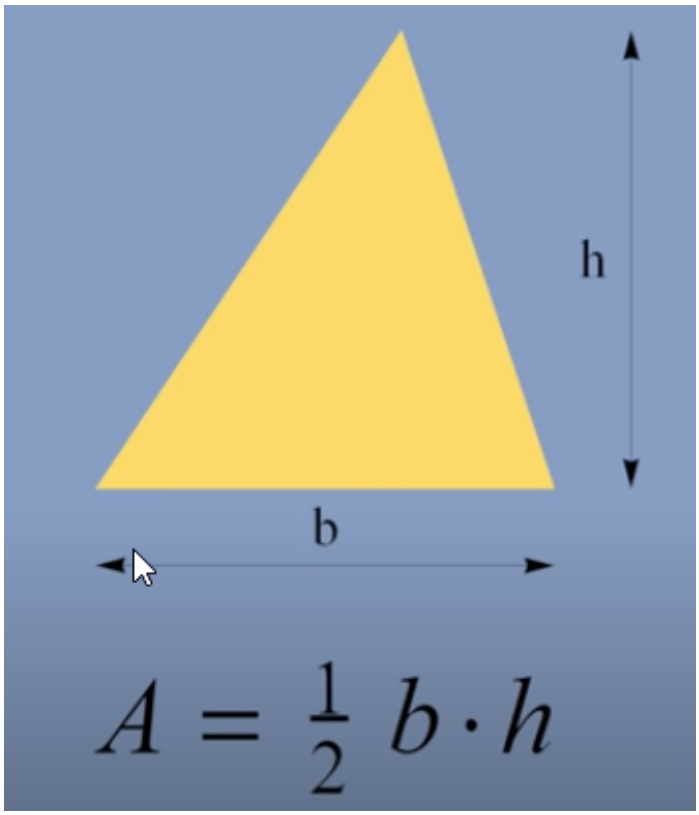
Front

Isometric left

Off-Axis front

Isometric up

- Sphere volume and surface area



$$a = \frac{1}{2} b \cdot h$$

where a, b, h are area, base and height respectively.

- triangle area
-
- check an number for Prime
- better implementation
- def isPrime() return True or False
-
- def rangePrime()
- use my_timer() check which function run faster

> the greatest common divisor (GCD) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers.

$$n! = n(n-1)(n-2)...1$$

$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

- factorial
- Greatest common divisor
- recursive GCD

> the least common multiple, lowest common multiple, or smallest common multiple of two integers a and b, is the smallest positive integer that is divisible by both a and b.

- LCM least common multiple

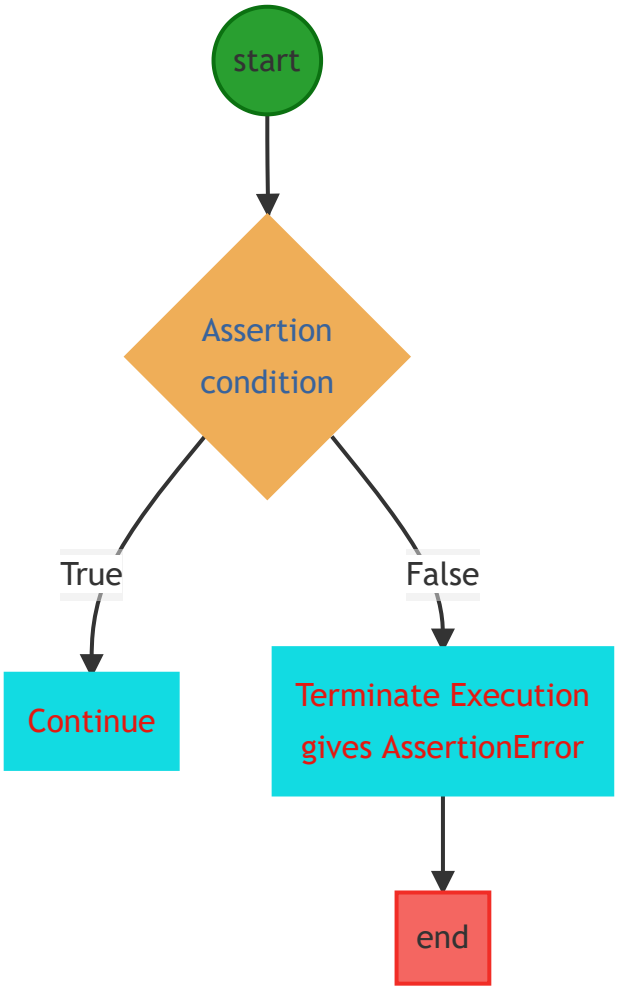
$$lcm(a,b) = \frac{|ab|}{gcd(a,b)}$$

- use above formula

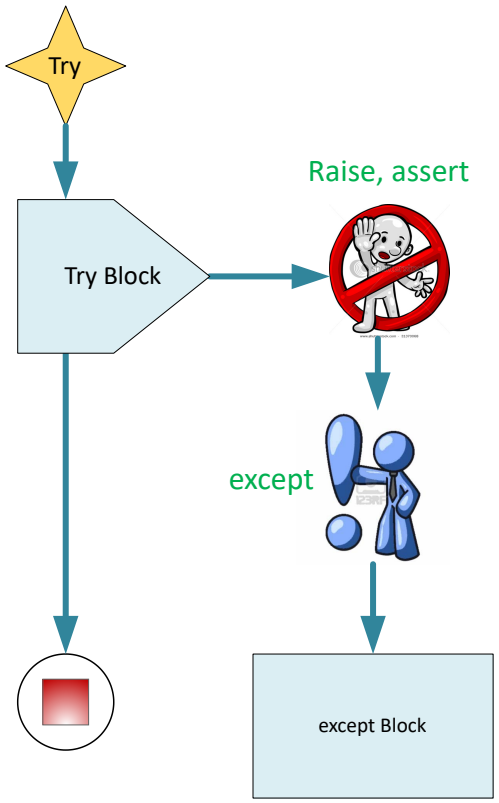
assert error check

- ❓what's wrongh?
- ✔Assert check before calculation

```
assert <condition>,<error message>
```



Raise Except



- ✋Raise TypeError
- assert
-
-

The difference between raise and assert:

1. assert: I swear this must be true, in case it happens, let me know. ❌❗You have big problem! Debug aid for developer find root cause, not for handling run-time error. only give you one kind of error which is AssertionError.

> 💡I swear int('1')==1

2. raise: Try to catch run-time error. Developer sometimes use raise for execution control.

> 👌💡define isFloat(str) function

- 👌👎catch Assertion Error
- 👌👎catch Different Error

✔Better solution is solve the issue at compiling time.

catch Except avoid termination

- catch assert

- catch raise
-

Create my own Error Type

> 1. look for the error type you may need;

```
>>> dir (__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'E
'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError', 'ConnectionRefusedError', 'ConnectionResetError
```

> 2. create your own error type

- create application specific Error

pass by reference

- Understand pass by reference
- Understand pass by value

function annotation

👍Avoid unexpected function call with wrong data type arguments.

👍Find out calling error before runtime.

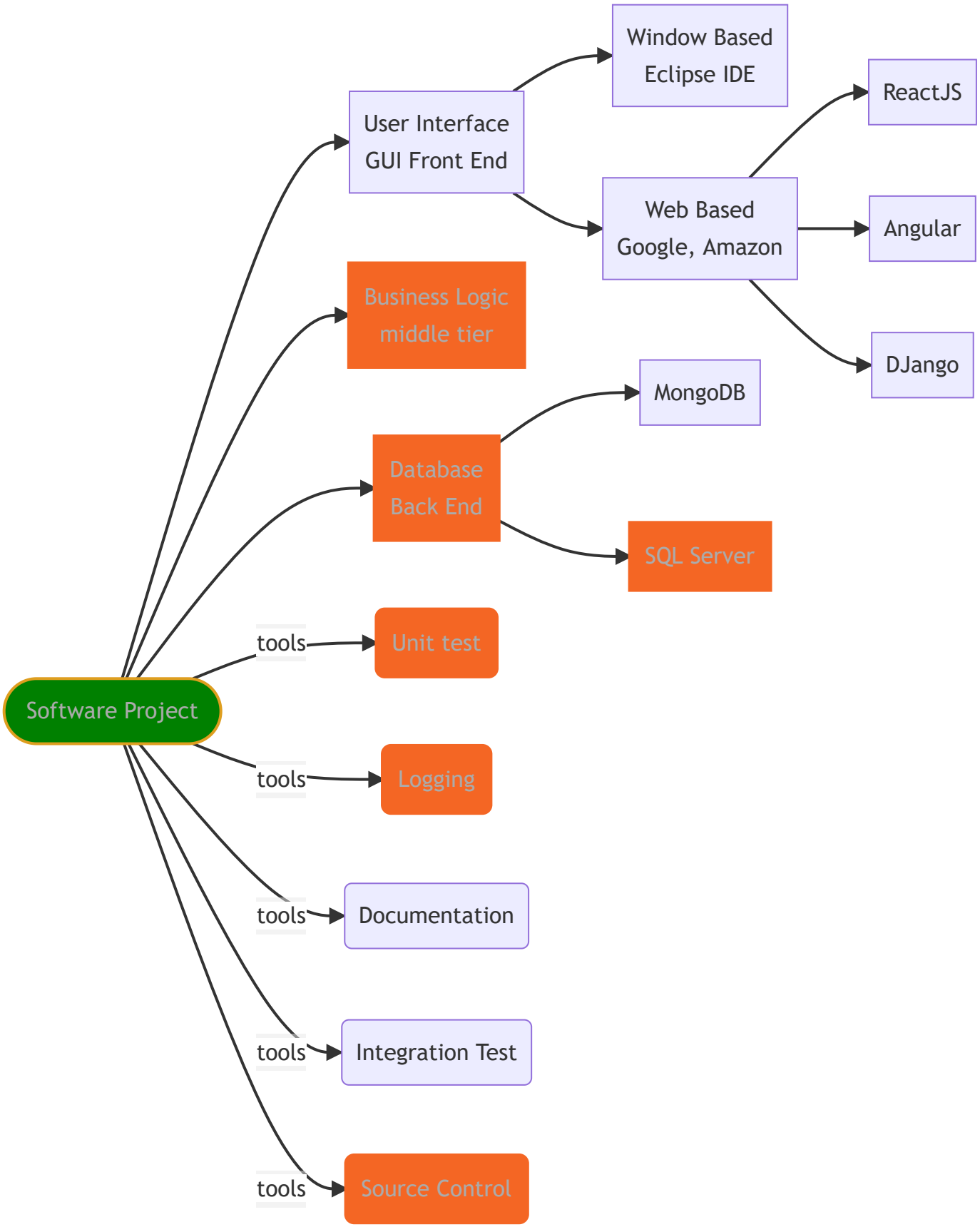
- ❓what's wrong?

```
mypy <filename.py>
```

```
(env) C:\Users\12818\workspace\python-I>mypy src/function/annotation1.py
Success: no issues found in 1 source file

(env) C:\Users\12818\workspace\python-I>mypy src/function/annotation1.py
src\function\annotation1.py:14: error: Argument 1 to "circle_area" has incompatible ty
pe "complex"; expected "float"
Found 1 error in 1 file (checked 1 source file)
```

- annotation1.py
- annotation2.py
- annotation3.py


Window Based
Eclipse IDE
User Interface
GUI Front End
ReactJS
Web Based
Google, Amazon
Angular
DJango
Business Logic
middle tier
MongoDB
Database
Back End
SQL Server
tools
Unit test
Software Project
tools
Logging
tools
Documentation
tools
Integration Test
tools
Source Control

Unit Test

> A unit is a specific piece of code to be tested, such as a function or a class. Unit tests are then other pieces of code that specifically exercise the code unit with a full range of different inputs, including boundary and edge cases.

Right-Click inside Editor window ⟹ Command Palette... ⟹ Python Cofigure Tests ⟹ unittest ⟹ test ⟹ test_*.py

```
[project root]
    ├── 🔥src/
    |    └── function/
    |           ├── circle3.py
    |           └── circle.py
    ├── test/
    |    └── 🚧test_circleArea.py
    └── 👉ReadMe.md
```

- 😟👎unittest cannot find the file unless
    1. test file name match the pattern
    2. test file located on right folder
    3. unittest always find test file from current running folder
    4. 👎module and function can be found in the module
    5. 👎module must be no compiler error

😄✔👍所有的错误，都是因为vscode Python Extension中的python执行命令。
Python永远都是从当前文件夹开始查找所有的module。Python本身并没有错。大部分网上的解释都没有切中要害。要害是python的执行命令与python的设计相违背。

```
(env) C:\Users\12818\workspace\python-I>c:/Users/12818/workspace/python-I/env/Scripts/python.exe
👎❌❗c:/Users/12818/workspace/python-I/src/blackjack/blackjackcard.py
Traceback (most recent call last):
  File "c:\Users\12818\workspace\python-I\src\blackjack\blackjackcard.py", line 1, in
<module>
    from src.blackjack.card import Card # use this line for unit test
ModuleNotFoundError: No module named 'src'

(env) C:\Users\12818\workspace\python-I>
```

从命令行看出，命令是从project根目录发出（C:\Users\12818\workspace\python-I>），直接运行该文件（c:/Users/12818/workspace/python-I/src/blackjack/blackjackcard.py）

```
# blackjackcard.py

# from src.blackjack.card import Card # use this line for unit test
from .card import Card # use relative path
# from card import Card # use this line for product
```

如果程序使用第三行，python当然知道从当前目录下寻找card module，所以直接运行该文件没有任何的问题。但是当使用unittest的时候，unittest就找不到card module了。（😄当然如果unittest也像python一样聪明，在相同文件夹中寻找，就更好了。目前想让unittest找当前文件夹下的module文件，必须使用相对路径，即第二行的 from .card， 又或者第一行的绝对路径）。unittest总是从绝对路径开始查找，所以第一行对unittest和python来说是轻车熟路。但是第一行和第二行对于命令行直接运行都是大问题，因为在该文件所在的文件夹中，根本不存在src或者.card的文件夹，所以相应的module当然也找不到。解决的办法是修改命令行命令。

```
✔😄python -m src.blackjack.blackjackcard
```

这样一来，第一行和第二行都可以直接运行，也同时能够让unittest找到。
网上很多解释说，你一定要在package文件夹中加入__init__.py的文件云云，其实没有一毛钱的关系。

‼️😟👎可惜的是，没有人修改执行命令‼️

👌Work around: it is hard to type in -m command, butter way to do this is add two line for the import, one for unittest, one for local run. switch the comment when you do different thing.

```
# tests/test_basics.py

import unittest
from mycoolproject import my_module_1
from mycoolproject import my_module_2

class TestMe(unittest.TestCase):
    def test_stuff(self):
        assert my_module_1.my_string == 'whoa, this is so kewl'

    def test_other_stuff(self):
        assert my_module_2.my_new_string == 'carl said: whoa, this is so kewl'

if __name__ == '__main__':
    unittest.main()
```

C:\Users\12818\workspace\python1-2>python -m unittest test/test_basics.py

unit test for circleArea

Regular Expression

- search() function

```
import re
```

start with, end with

> ^: start with
> ...$: end with
> *: Zero or more occurrences

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"planet$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
|	Either or	"falls|stays"
()	Capture and group	

Logging

❓What is logging?
✔write software execution record to console, file or database used for application analysis.
there are at least 5 level of logging: Debug, Info, Warning, Error, Fatal

- send log message to a file
- use print do the logging

nothing wrong with it, for small program it is fine. once the program getting bigger, this is no good. you start dealing with complecated problem, you need log level, log analysis, and more

,

- import logging

> get bunch of function right out of the box.
> 🔑😄 **Knowlodge Base**
>
> 1. default logging level is "warning"
> 2. root is the default logger name

- write logging to a file
-
- use get logger
- use config file

Logging configuration

algorithms

Big O

- O(n)
- O(log n)
- linearSearch.py
- binarySearch.py

Operations on Data Structure

1. Access and read values
2. Search for an arbitrary values
3. Insert values at any point into the structure
4. Delete the value in the Data Structure

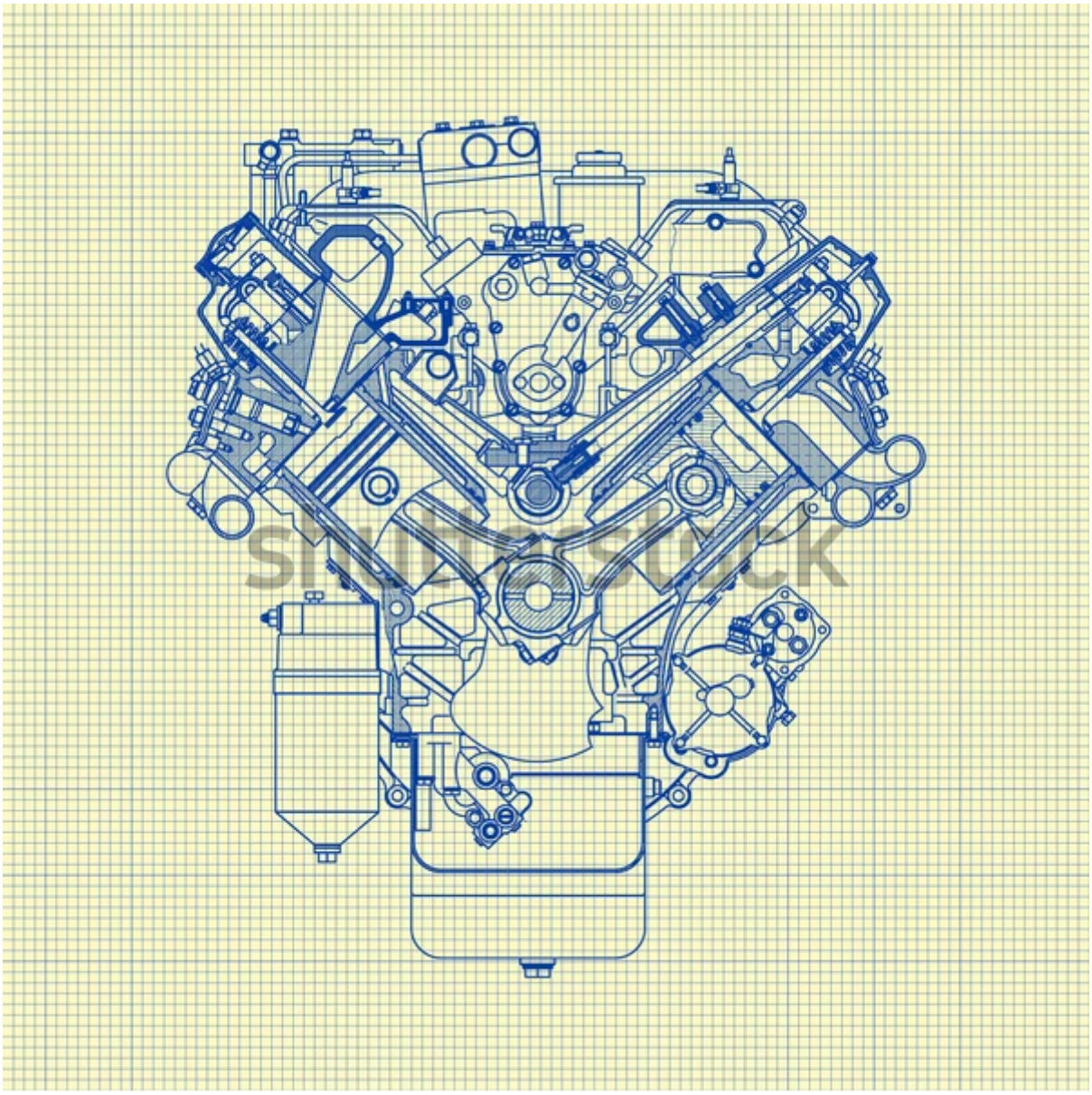
- arrays.py

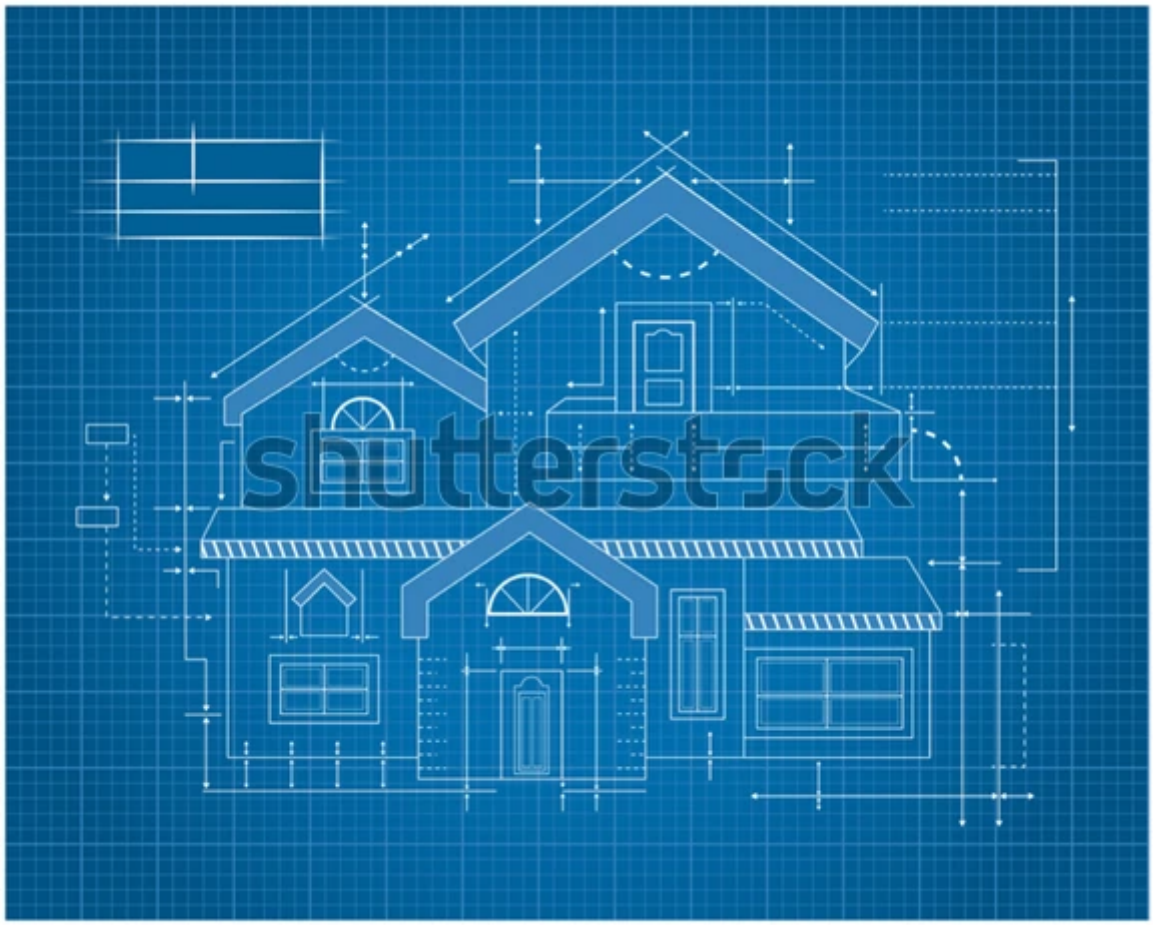
Class

❓ what is class?

> Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state.

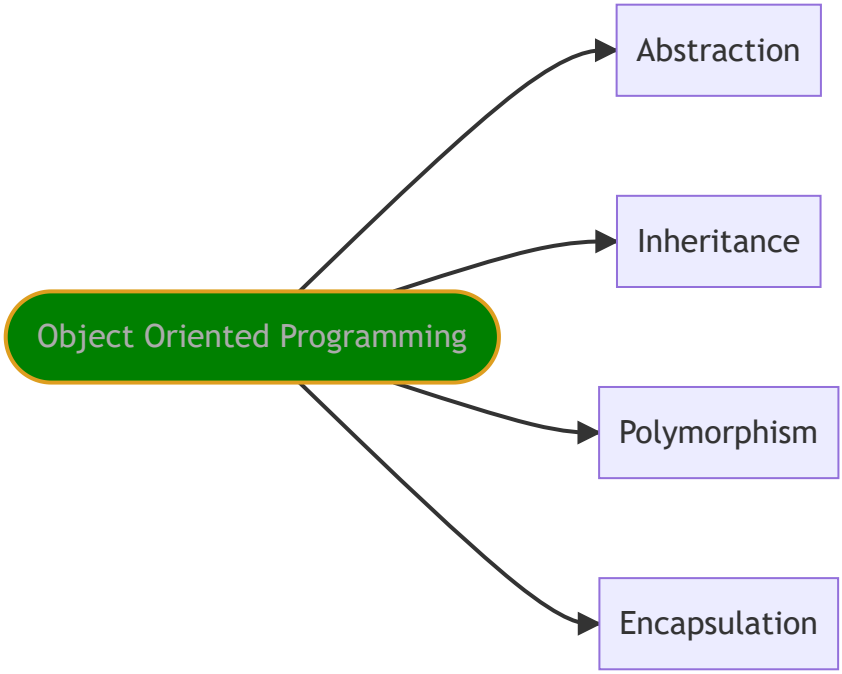
Design





www.shutterstock.com · 99718403

www.shutterstock.com · 1930415813



4 Features of OOP

1. Abstraction:class is a abstraction of object in real world to python program object type.（实体模拟）
2. Inheritance: a class can inherit from multiple other class to increase code reusability.（共性继承）
3. Polymorphism:same function behavior differently by different object type.（异类同功）
4. Encapsulation：avoid data or function being called outside the class unintentionally（自我保护）

❓ How to make it happen?

> ✔ build a class.

ClassName
attributes
functions()

class in python is a definition of group of attributes (class name, variables & functions) which is abstracted from real world to computer world (virtual world), and can be used to create object. those objects can then do something based on its variable attributes.

> Behavior: do something based on what it knows!

in python, both variable and function are treated same as attrubutes.

```
class Robot:
  pass

robot = Robot()
robot.jump() # AttributeError: 'Robot' object has no attribute 'jump'
```

```
>>> class Robot:
...     pass
...
>>> r = Robot()
>>> r.jump()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Robot' object has no attribute 'jump'
>>>
```

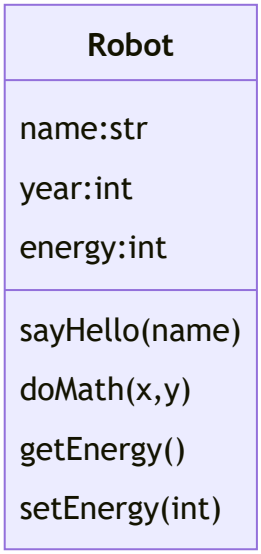
as you can see, we call jump() function by r.jump(), Python raises a AttributeError. because we never defined a function named jump().

Abstraction

❓ why abstraction?

> ✔ because it is impossible to copy every single feature of an object from real world, we can only pick something that we are interested in the virtual world. That is so-called abstraction.

- Robot from real world



class basic

- add attribute dynamically

🔑😁 Knowlodge Base

> define a class means to create a new data type.

```
(env) C:\Users\12818\workspace\python>c:/Users/12818/workspace/python/env/Scripts/python.exe c:/Users/12818/works
<class '__main__.Robot'>
<class 'object'>
```

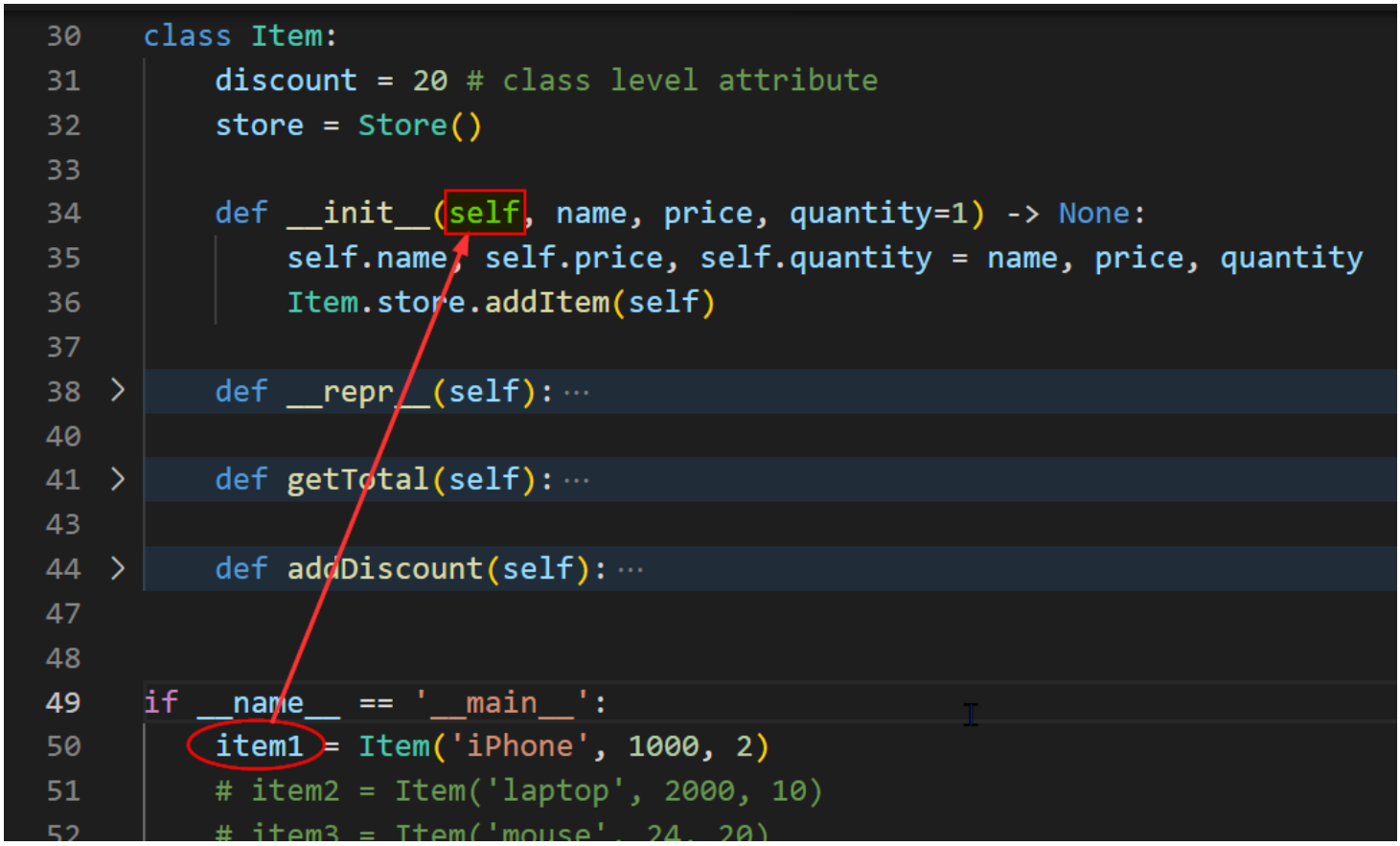
```
>>> from src.myclass.class01 import *
>>> x = Robot()
>>> type(x)
<class 'src.myclass.class01.Robot'>
>>>
```

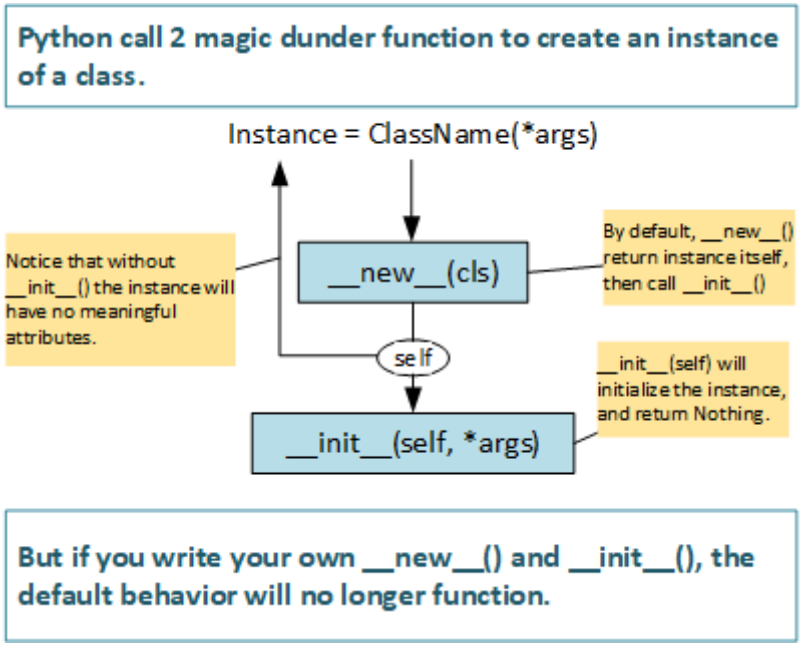
- everything in Python is class
- define function outside of class

- define function inside class, class level, instance level
- define **init**()
- understand **new**() and **init**()

🔑😄 Knowlodge Base

> 1. **new**() is constructor which is called when creating an instance of class;
> 2. **new**() will automatically call **init**() to initialize the instance;
> 3. create your own **new**() will override the functionality of super class;
> 4. **init**() is called automatically when creating an instance of a class;
> 5. once **init**() is defined, the **init**() of super class will be overridden with no functioning, unless
> 6. **init**() call super().**init**() to reuse whatever defined in super class;
> 7. **init**() is used to initialize the instance which returns nothing.
> 8. It is the **init**() makes each instance from the same class different.



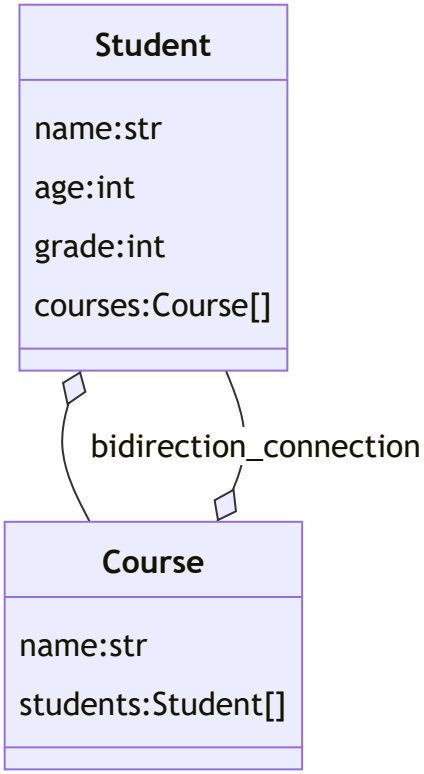


use type to create class

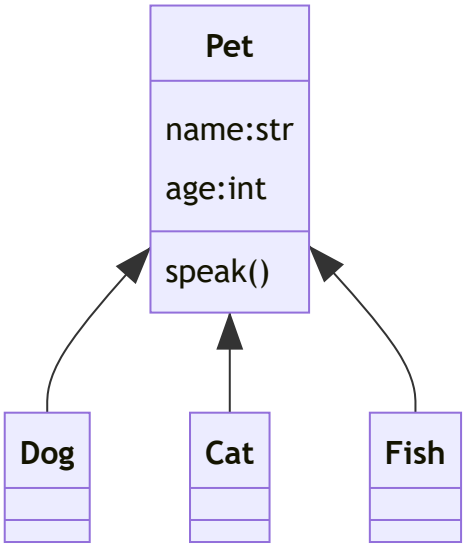
Use type() to create new class dynamically

- Excersize:

- class level attribute vs. instance level attribute
- dynamically assigned attribute
- save and load data with file
- use csv DictReader() function
-
- **init**(self, inputName=None)
- private attribute __energy
- getter, setter, property
- class level attributes
- related classes



Inheritance



- Why inheritance?

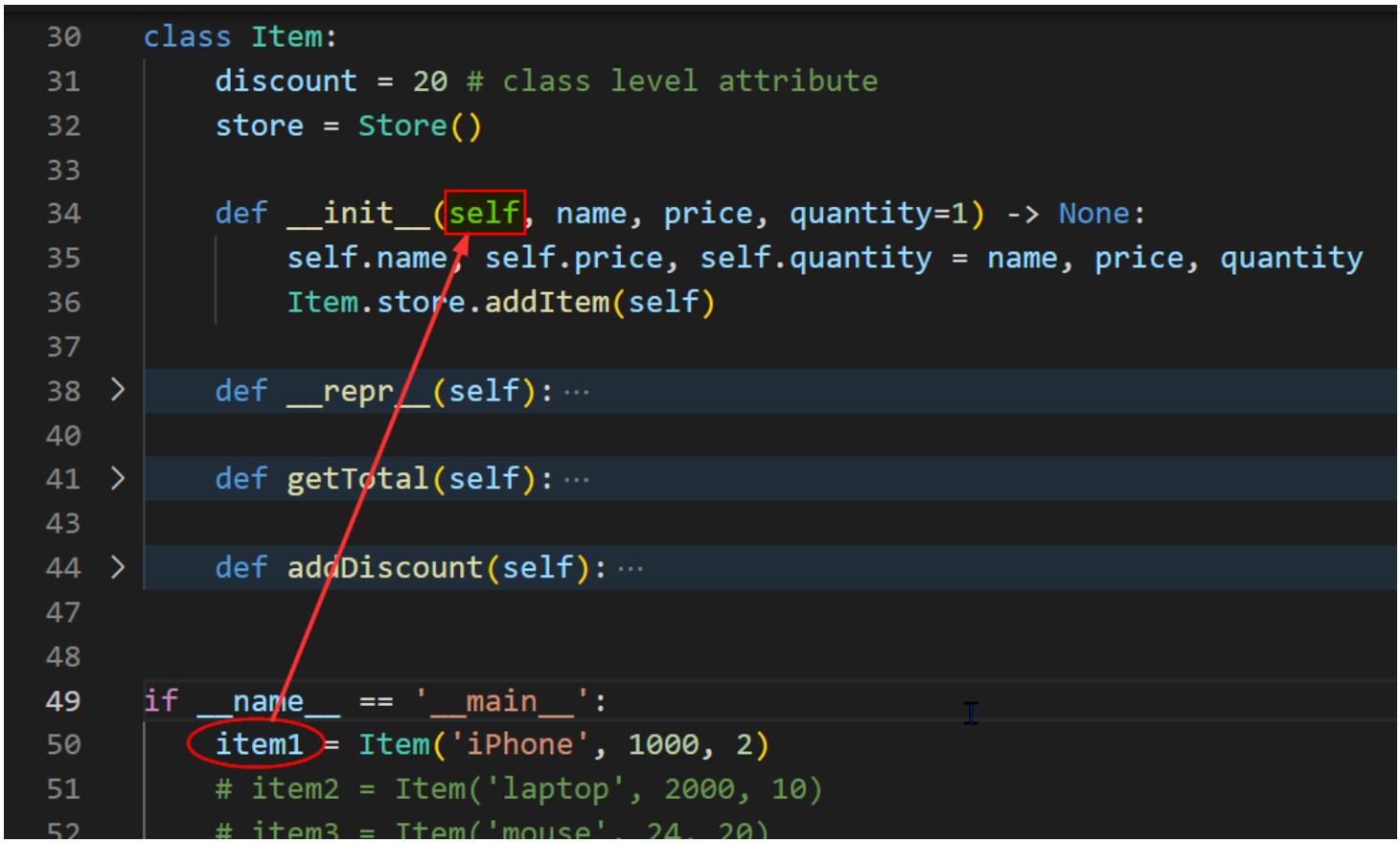
Polymorphism

- Polymorphism

🔑😄 Knowlodge Base

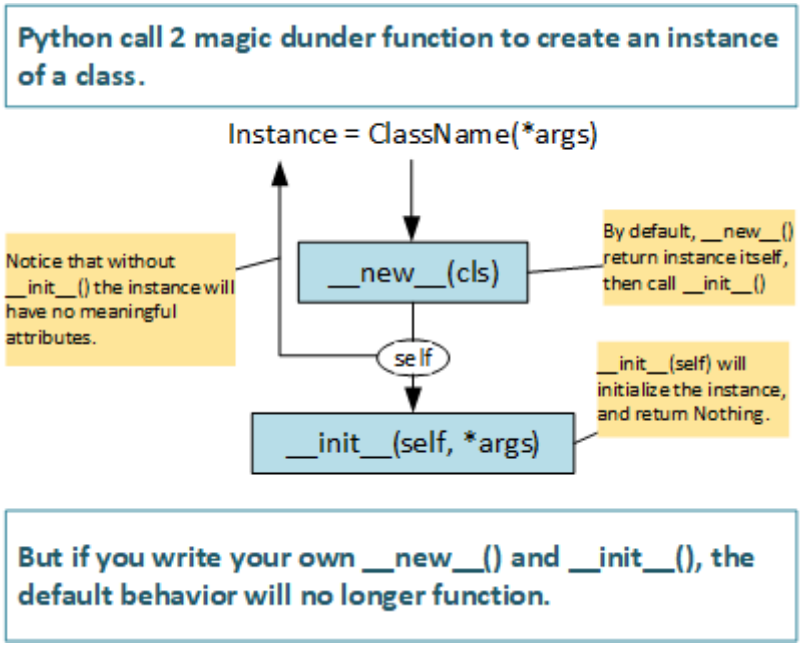
> Define a class means to create a new data type.

1. The difference between regular function and instance function is the very first positional argument is always **self** in class function.
2. instance level attributes can be defined in any instance function.
3. class level attributes should be used by class name.
4. static function does NOT use any instance level attribute, so no **self** argument is needed.
5. all class inherits from **object** class.
6. every thing is class instance in python (such as 3, float, set, [], (), ...).
7. **self** is the first positional argument of all instance level function, which can be used to access all variable and function attributes, it represents the instance itself, it then can be passed as the instance.



dunder functions

- **repr**(good enough) vs. **str**
- **init**(good enough) vs. **new**



- ❓override **new**, return other class instance
- override **iter**, **next**, create iterable

🔑😄 Knowlodge Base

> 1. each iterable object implement **iter**()
> 2. built-in function: iter() call **iter**()
> 3. built-in function: next() call **next**()

- YouTube Itertools
- 👍😄 itertools Document
-

🔑😄 Knowlodge Base

> 1. anything is iterable it implements **iter**() function;
> 2. anything is iterator it implements **next**() function;
> 3. when all elements are iterated, raise StopIteration Error.

- range1 start from 1, include stop
- **call**() make object callable
- **eq**(check if same), **add**
- **eq**(), == vs is

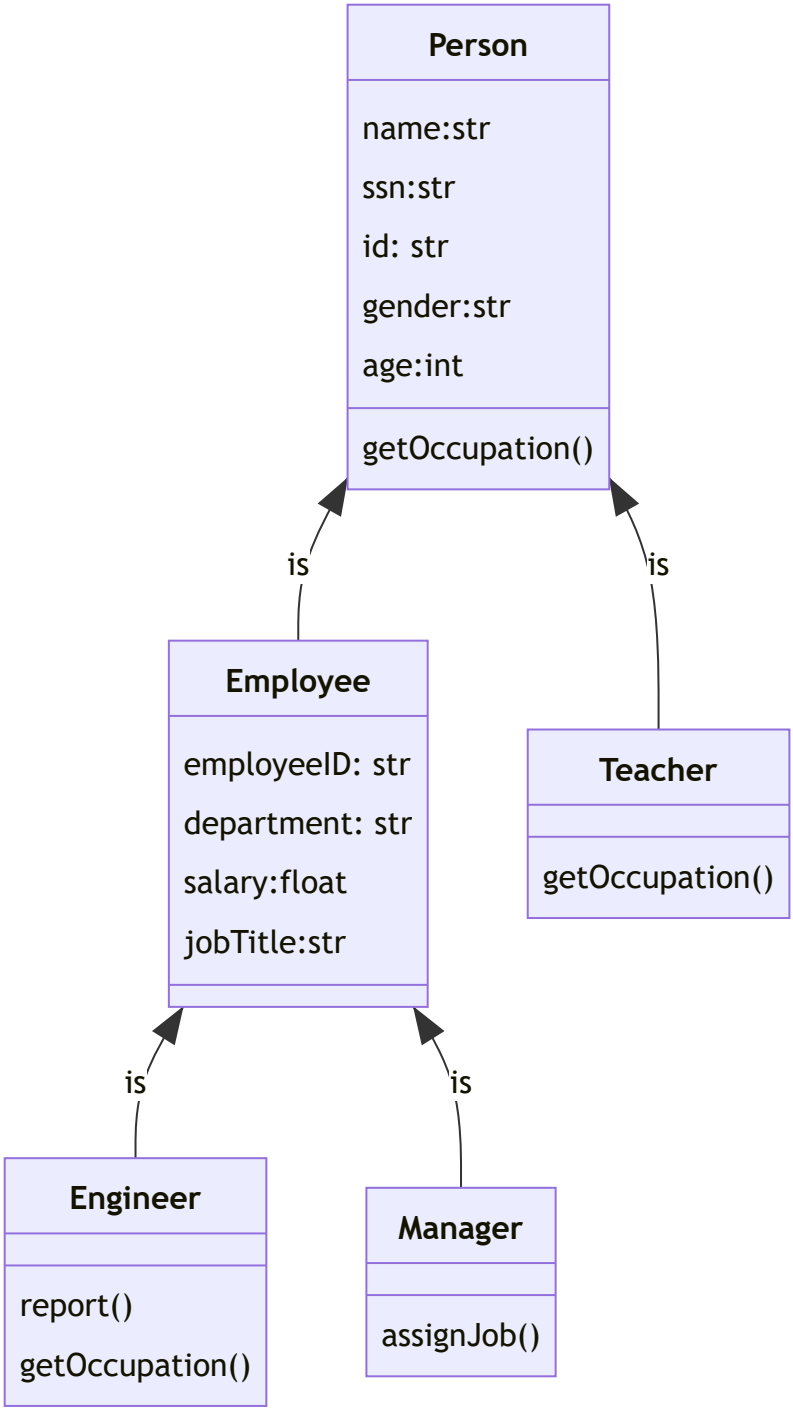
attribute scope

- class level attribute
- instance level attribute

class tricks

- override **new**, return other class instance
- pass outside function to class
- internal function call another internal
- nested class
- composition vs. inheritance
- class method, static method
- define a class as default value

class inheritance

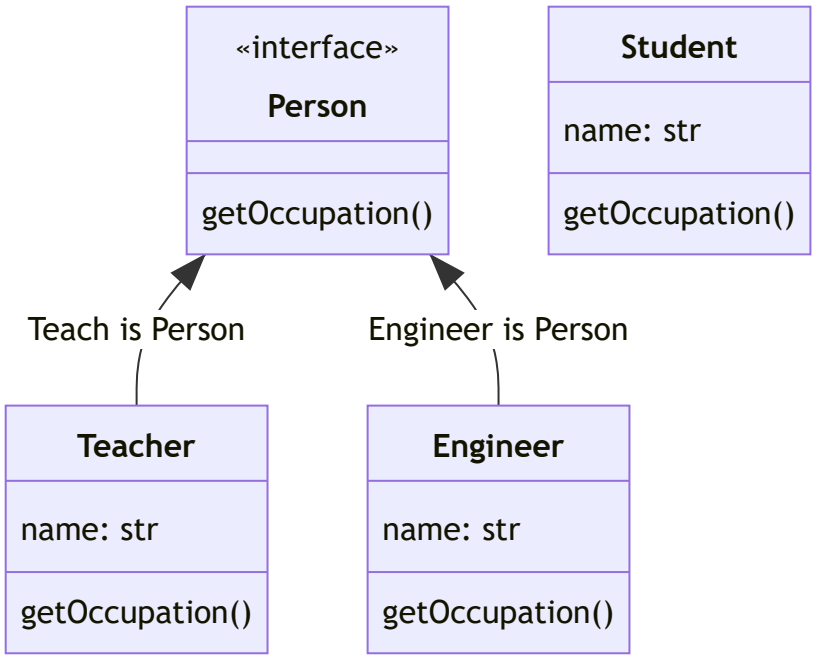


- class inheritance, isinstance(obj, class)
- advatage of inheritance, inherite from Enum
- inherit from Enum
- multiple inheritance

Python Interface

@abstractmethod

interface.py



- @abstractmethod decorator from abc
- **subclasshook**(), **subclasscheck**(), issubclass(), isinstance()

Unit Test

> A unit is a specific piece of code to be tested, such as a function or a class. Unit tests are then other pieces of code that specifically exercise the code unit with a full range of different inputs, including boundary and edge cases.

Dunder Variables

- **doc**
- **name**
- **code**

Global Variables

Blackjack Game

Blackjack Rules

- Black Jack Rules

1. Object of the game:

> beat the dealer by getting a count as close to 21 as possible, without going over 21

2. Card Values

> ace is worth 1 or 11, J,Q,K are 10, other card is its pip value

3. Betting

> for simplicity, we don't bet.

4. Shuffle and cut

> the dealer shuffles the pack of card, no need player cut

5. Deal

> dealer gives one card face up to each player, and one card face up for himself. Another round of cards is then dealt face up to each player, but the dealer takes the second card face down.

6. Naturals

> If a player's first two cards are an ace and a "ten-card" (a picture card or 10), giving a count of 21 in two cards, this is a natural or "blackjack." If any player has a natural and the dealer does not, the dealer lose. If the dealer has a natural, other doesn't, dealer win. If both dealer and player have natural, no body wins.

7. The Play

> any player on his turn must decide whether to "stand" (not ask for another card) or "hit" (ask for another card in an attempt to get closer to a count of 21, or even hit 21 exactly). Thus, a player may stand on the two cards originally dealt to them, or they may ask the dealer for additional cards, one at a time, until deciding to stand on the total (if it is 21 or under), or goes "bust" (if it is over 21). In the latter case, play loses the game. The dealer then turns to the next player and serves them in the same manner. The combination of an ace with a card other than a ten-card is known as a "soft hand," because the player can count the ace as a 1 or 11, and either draw cards or not. For example with a "soft 17" (an ace and a 6), the total is 7 or 17. While a count of 17 is a good hand, the player may wish to draw for a higher total. If the draw creates a bust hand by counting the ace as an 11, the player simply counts the ace as a 1 and continues playing by standing or "hitting" (asking the dealer for additional cards, one at a time).

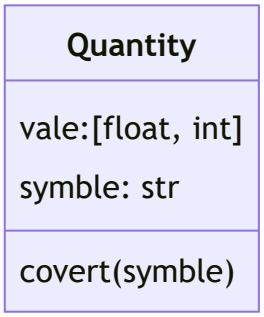
8. The Dealer's Play

> When the dealer has served every player, the dealers face-down card is turned up. If the total is 17 or more, it must stand. If the total is 16 or under, they must take a card. The dealer must continue to take cards until the total is 17 or more, at which point the dealer must stand. If the dealer has an ace, and counting it as 11 would bring the total to 17 or more (but not over 21), the dealer must count the ace as 11 and stand. The dealer's decisions, then, are automatic on all plays, whereas the player always has the option of taking one or more cards.

9. No Splitting Pairs
10. No Doubing Down
11. No Insurance
12. Reshuffling when start new game.

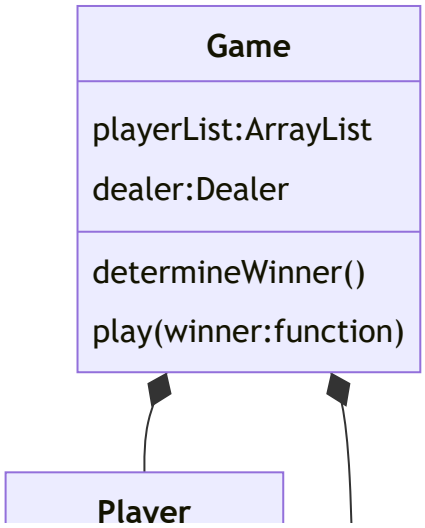
Physics Unit

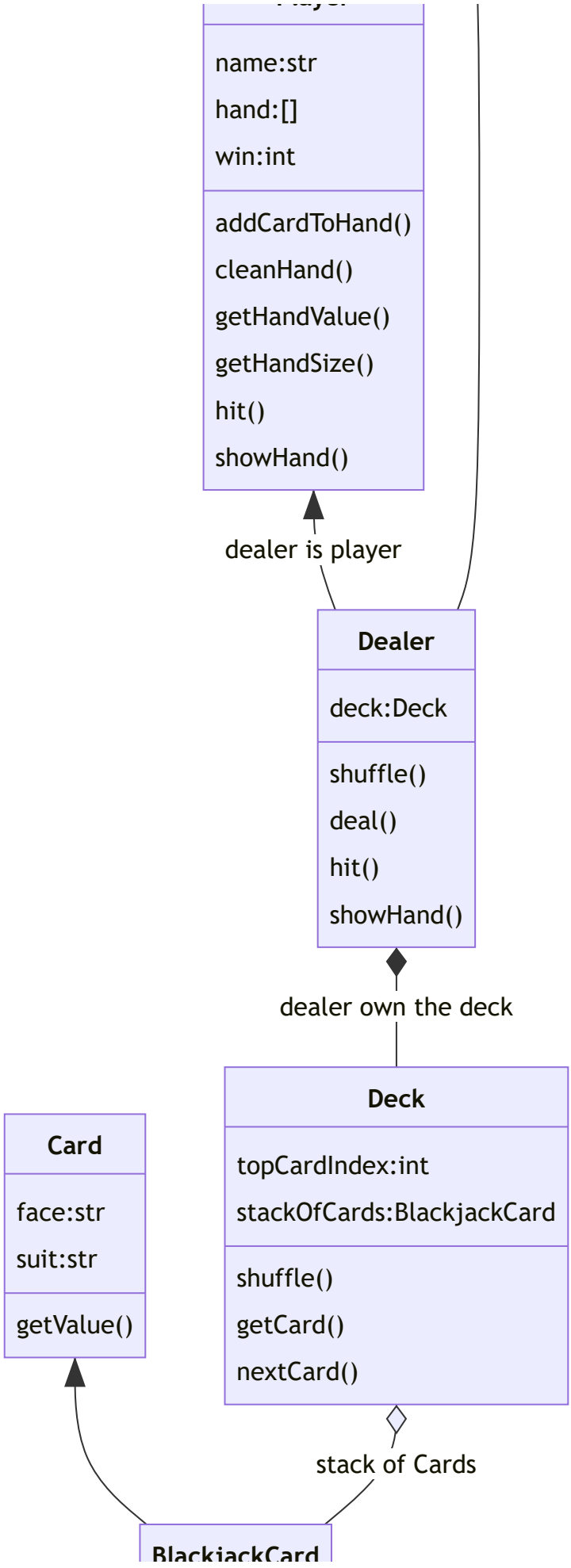
- physics unit

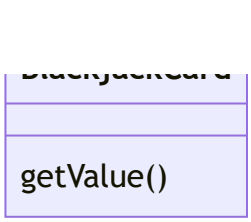


Blackjack Card Game

Object relationship



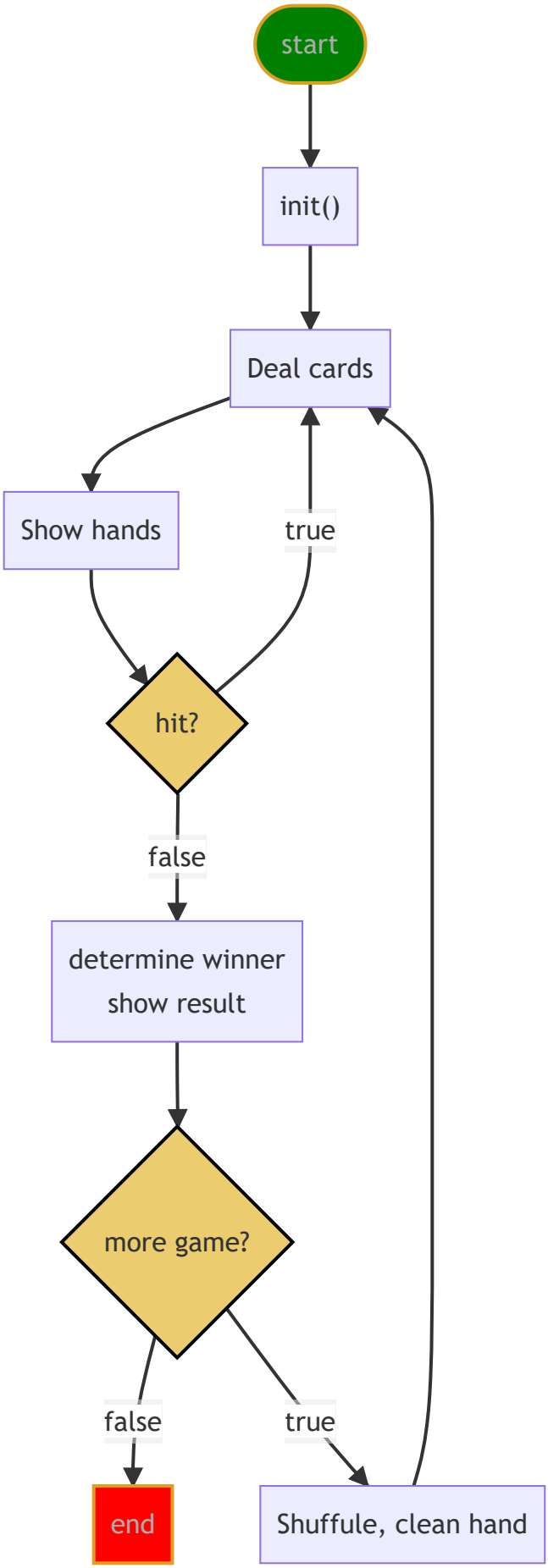

name:str
hand:[]
win:int
addCardToHand()
cleanHand()
getHandValue()
getHandSize()
hit()
showHand()
dealer is player
Dealer
deck:Deck
shuffle()
deal()
hit()
showHand()
dealer own the deck
Deck
topCardIndex:int
stackOfCards:BlackjackCard
shuffle()
getCard()
nextCard()
Card
face:str
suit:str
getValue()
stack of Cards
BlackjackCard



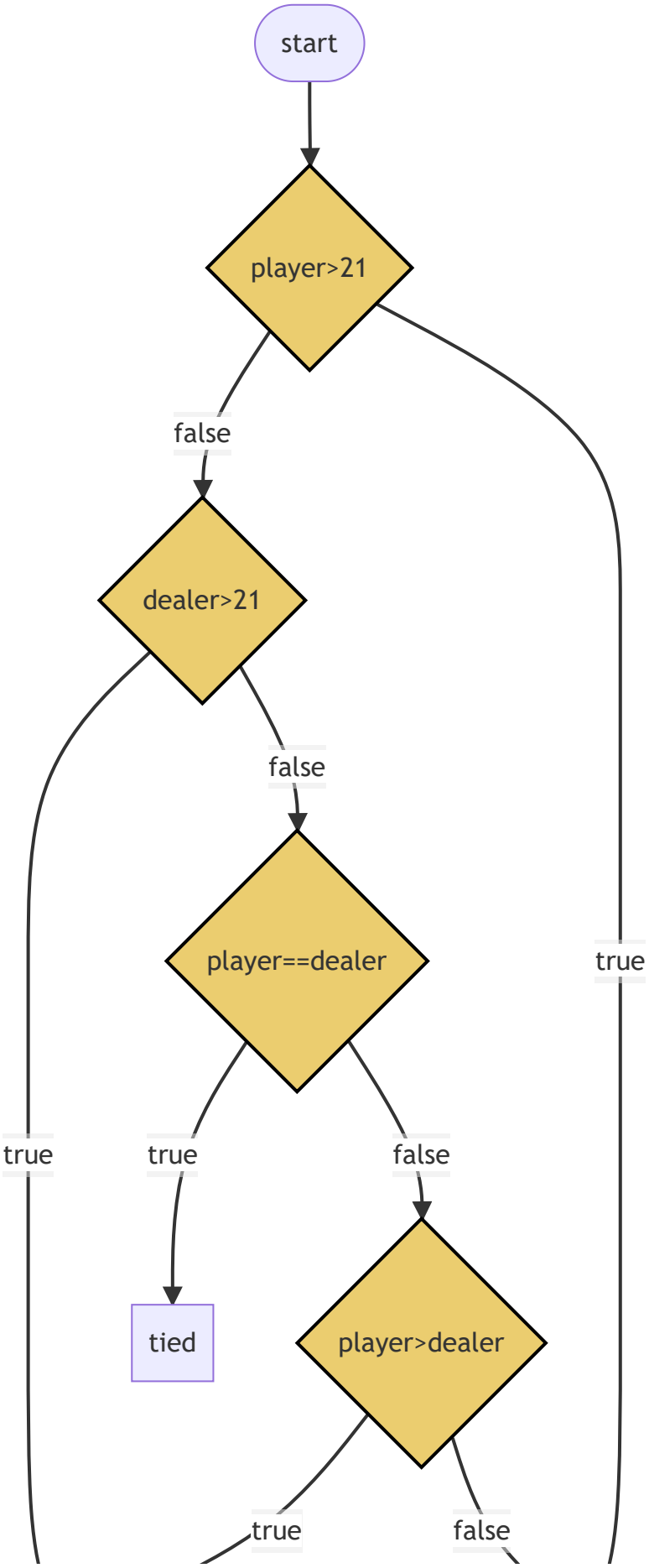
blackjack

Unit test

Game logic


start
init()
Deal cards
Show hands
true
hit?
false
determine winner
show result
more game?
false
true
end
Shuffule, clean hand

Code Optimization



missing unit tests

- player get 4 Ace
- player get 3 Ace
- player get 2 Ace

Integration Test

❓What is Integration Test?

> ✔INTEGRATION TESTING is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Integration test

✔Play the Blackjack game by running Game class.

Documentation

❓How to document Python code?

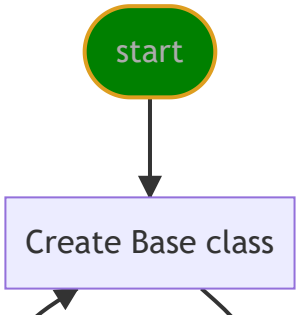
✔

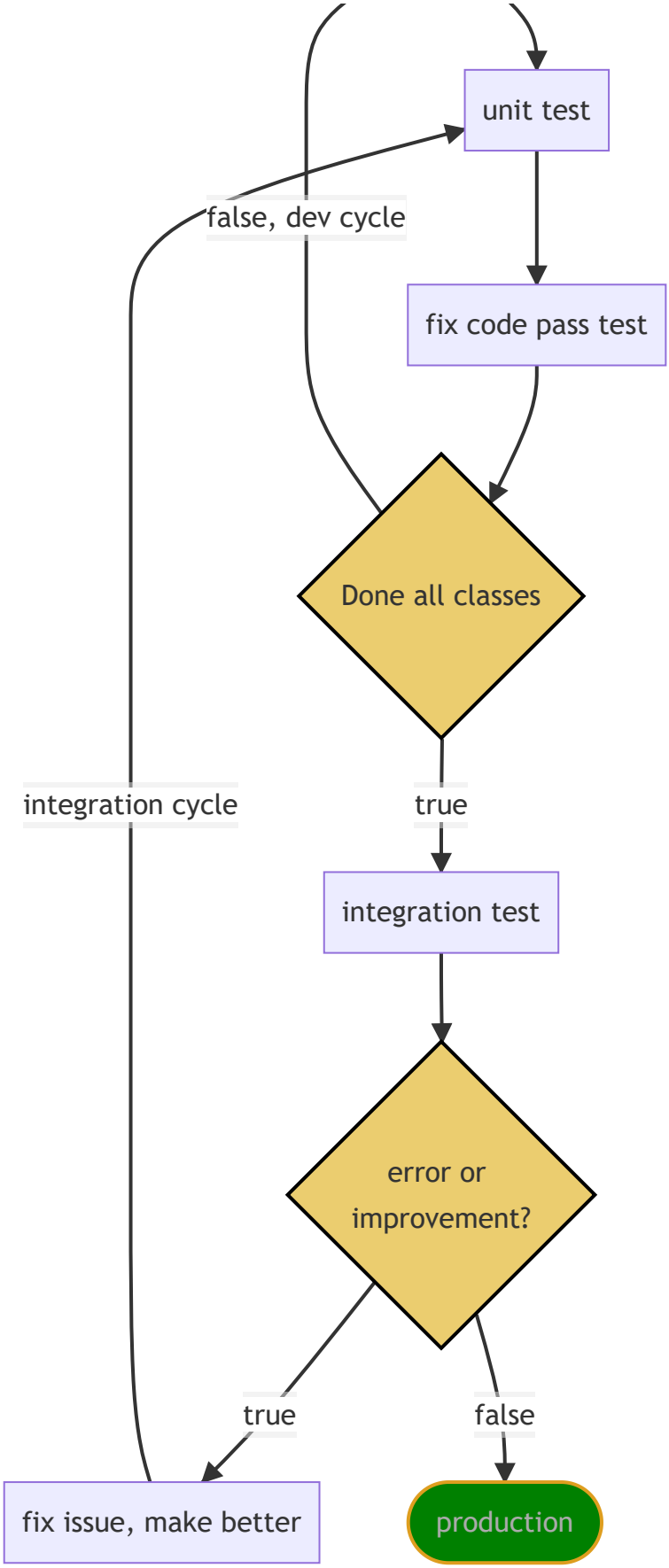
❓How to read Python code?

✔ How to read Python code

Software development life cycle

- Test Driven Development (TDD)





- After integration test, we find
  ❗missing test of 2,3,4 Ace?
  ✔modify getHandValue() method defined in Player class, put if condition in for loop.

test2Ace(),test3Ace(),test4Ace()

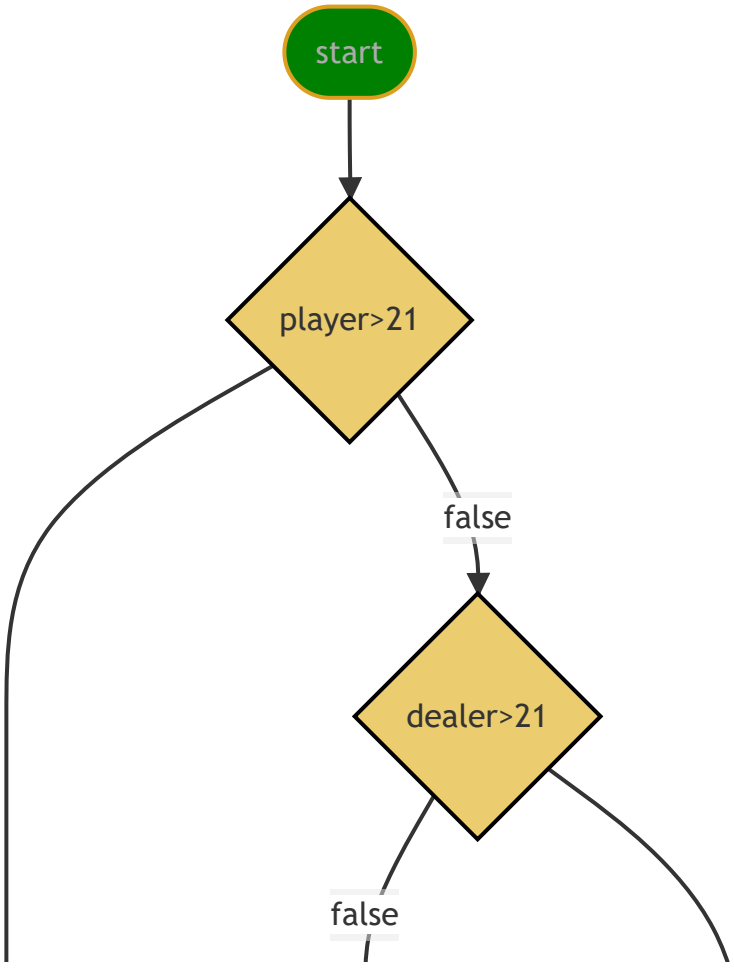
❌we treat one Ace more than one time!

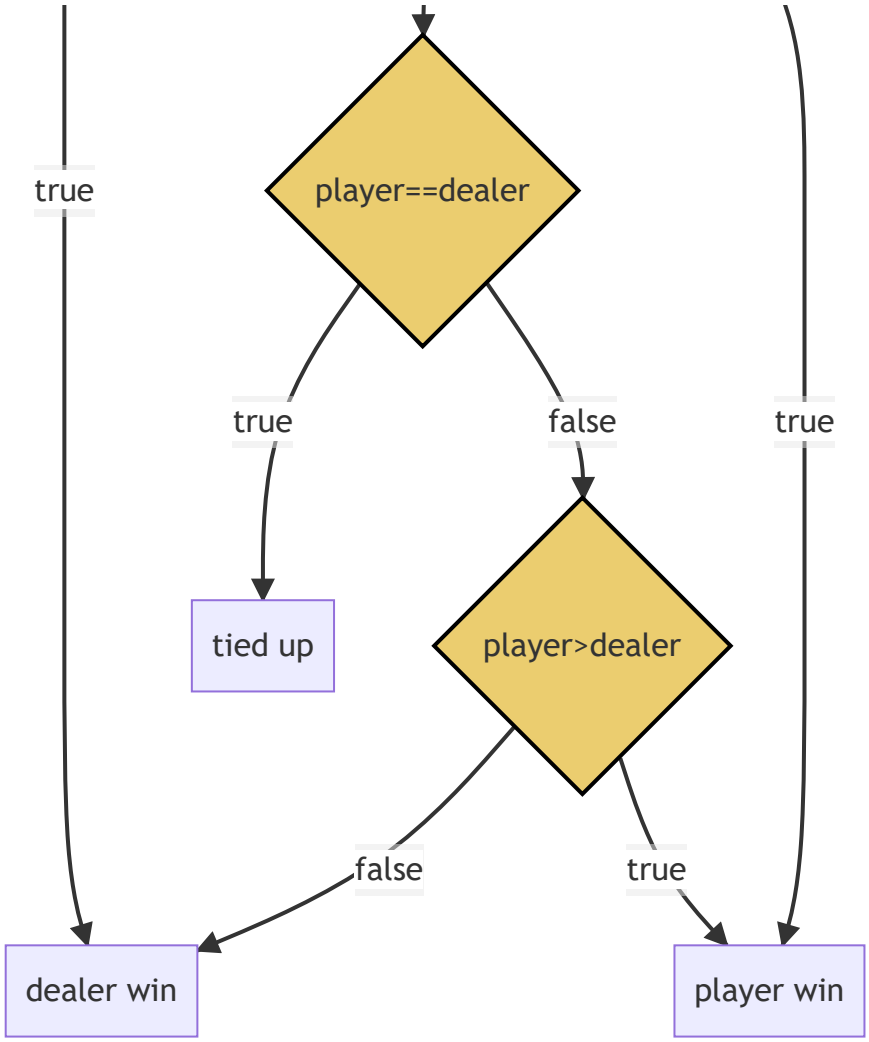
✔

```
def getHandValue(self):
    value = 0
    for card in self.hand:
        value += card.getValue()
    a = self.numberAce()
    while value > 21 and a>0: # A=11,
        value -= 10 # change A=1
        a -= 1
    return value

def numberAce(self):
    count = 0;
    for card in self.hand:
        if card.face == 'A':
            count += 1
    return count # return number of Ace in hand
```

❗the determineWinner() method looks ugly



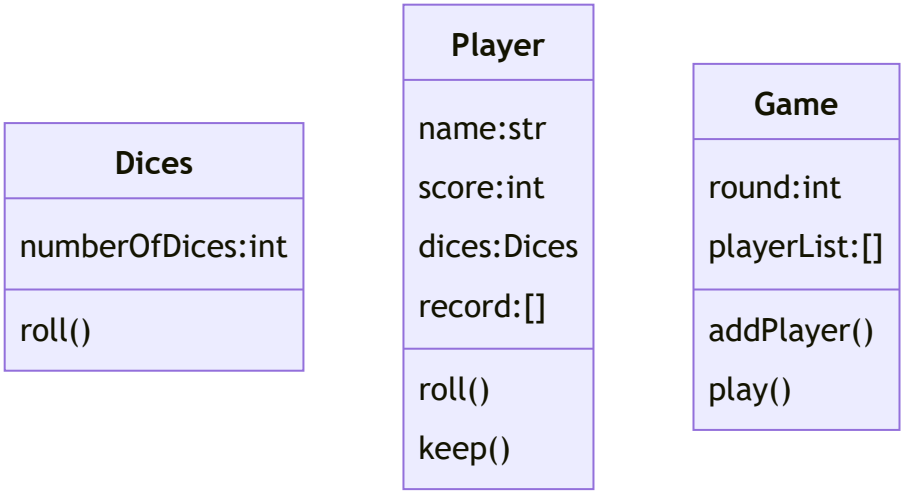


Deployment

python deployment

Yahtzee Dice Game

Yahtzee Dice Game
Yahtzee Python



File Access

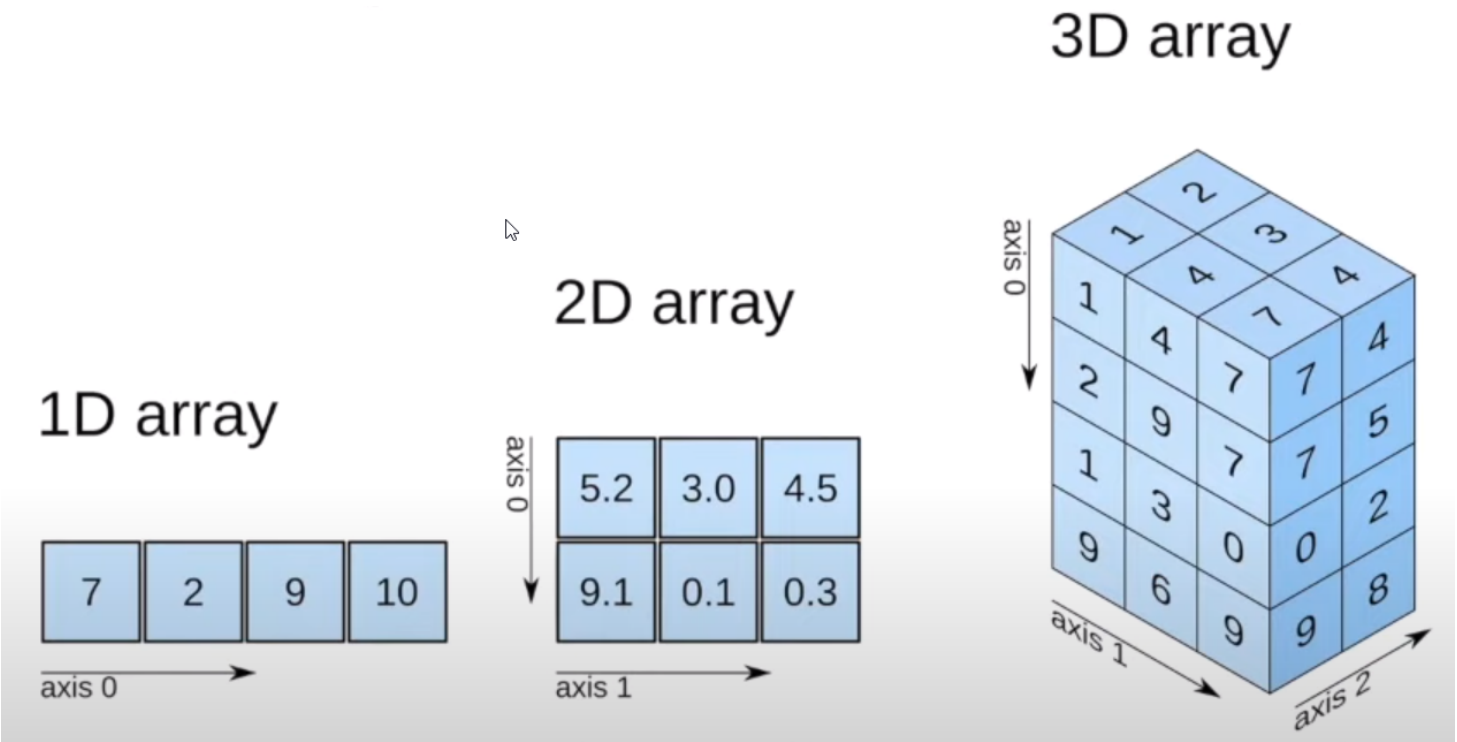
- write plain text to file Hello.txt
- read/append plain text from/fo file
- read text file
- with open
- read CSV file
- read CSV file using csv module
- read CSV file using pandas
- read large CSV file pokemon.csv
- write dict to CSV file
- write json file
- read json file
- load json string as dict
- use pandas read json file
- plot student score
- list all files in directory

nmpy

❓ What is numpy module in python?

> ✔NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

website



- Mathematics(MATLAB Replacement)
- Plotting (Matplotlib)
- Backend(Pandas, Connect 4, Digital Photography)
- Machine Learning
- array basics
- Homework 01
- Homework 01 solution
- linear algebra
- read data from file
- matrix dot product
- Weighted average

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}$$

- standard mean sqrt标准均方差
- linear algebra几个老头几个梨

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 5 \\ 2 & 5 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 27 \end{bmatrix}$$

pandas

Document website

- simple curves
- Covid-19
- Create a instance of DataFrame
-

Data Analysis for Excel Users

- Anaconda3
  File: Anaconda3-5.3.0-Windows-x86_64.exe

Plot

- covid
- plot list
- plot sine wave
- plot both sine and cosine wave
- plot 3 functions in one chart
- plot 6 lines in one chart with legend
- multiple pages with title icon
- stack plot with custom color
- box plot with data list
- box plot with random data
- errorbar plot
- circle pie chart

- 3D pie chart
- 4 sub plots in one page
- 4 sub plots with each lenged
- adjust plot programtically
- math symbol in legend
- custom tick label
- custom tick for sine wave
- color ticks
- remove chart box lines
- add grid to chart
- curve fit
- change background to dark
- log style
- generate animated sine wave
- 3D animation generator

Pandas

padans chart visualization

pandas DataFrame

create a DataFrame

1. use dict to create a DataFrame, key is column name, value is list
2. use array to create a DataFrame
3. use dataclass to create a DataFrame
4. use df.at(row, column) to get individual value of DataFrame
5. use df.info() to get definition detail of a DataFrame
6. use df.values to get all values
7. use df.axes to get all columns

Clean Code

Clean code

Design Principles SOLID

SOLIS website

1. Single Responsibility principle

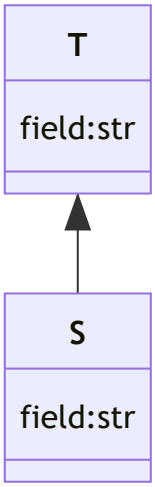
> A class should have one, and only one, reason to change. You need to change your class as soon as one of its responsibilities changes. it makes your software easier to implement and prevents unexpected side-effects of future changes.

2. Open/Close Pricinple

> Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification.

3. Liskov Substitution Principle

> Let Φ(x) be a property provable about objects x of type T. Then Φ(y) should be true for objects y of type S where S is a subtype of T.



$$\phi(x) = \phi(y)$$

where

```
x = T()
y = S()
```

4. Interface Segregation Principle

> Clients should not be forced to depend upon interfaces that they do not use.”

5. Dependency Inversion

> High-level modules, which provide complex logic, should be easily reusable and unaffected by changes in low-level modules, which provide utility features.

应变（requirements change over time. least code change on requirement changes.）

Turtle

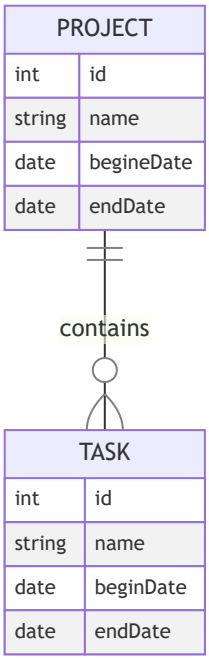
turtle document

- Open window
- move turtle
- mouse click move turtle
- Randomly moving turtle on mouse click.
- limit the turtle within window bounds
- turtle moving within certain area.
- Display cards on turtle screen
- draw star
- draw half circle
- draw spiral turtle
- draw dragon curve
- draw sun and house
- draw snowman by using class Snowman
- draw ellipse
- define functions for line, triangle, circle and rectangle

SQLite

CRUD: Create, Retrieve, Update, Delete

- Create database, table
- Insert many rows
- Retrive data
- Update data
- Delete data
- One-to-Many Create table

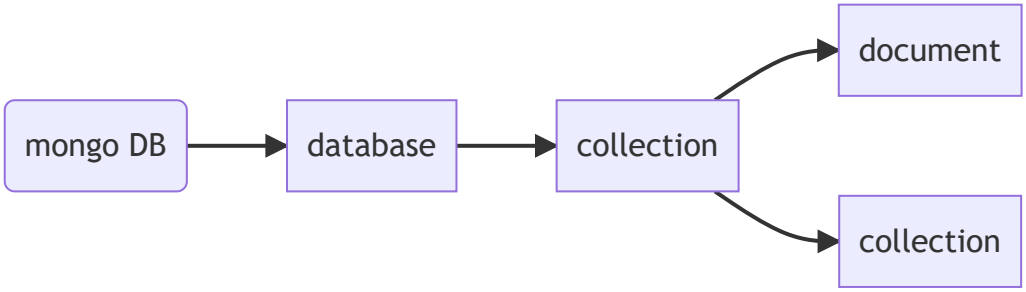


- Create one-to-many data
- Retrieve one-to-many data
- Create books table
- insert sample books
- many-to-many relationship, DB helper
- Product, Provider class
- CRUD for books

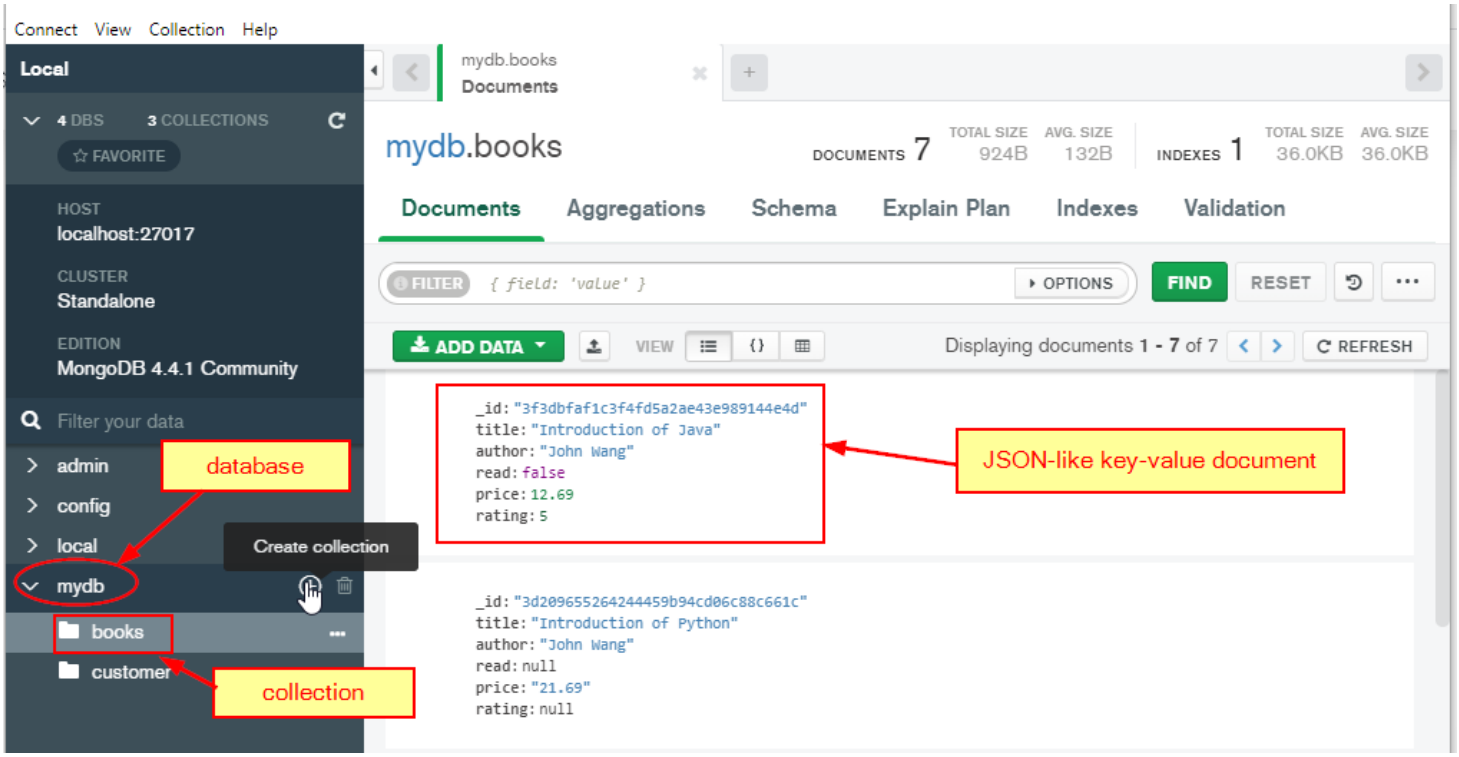
MongoDB

❓What is MongoDB?
✔Object-Oriented based One of NoSQL database application written in C++.

1. stores data in JSON-like documents that can have various structures
2. uses dynamic schemas, which means that we can create records without predefining structure such as SQL relational database table.
3. the structure of a record can be changed simply by adding new fields or deleting existing ones.



4. document database
5. key-value database



❓What is NoSQL database?

✔NoSQL databases (aka "not only SQL") are non tabular, and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

❓What is SQL?

✔SQL stands for Structured Query Language specially for relational database.

SQLite: Python built in SQL database.

- CRUD, Relationships and More
- Schema Validation, Advanced Queries and More
- Full-Text Search with Atlas Search
-

tkinter(windows based GUI)

```
pip install tk
```

image converter, mp4>gif, png>ico

❓What is tkinter?

> ✔This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.

- Good tkinter document online
- YouTube tutorial
- 油管系列讲座
- atom version

in tkinter everyting is widget

open window

- open window with title

switch between frame

- Switch between two Frames

Label, Button, Entry widgets

- Add label to window

pack attributes

- after=widget1 - pack this widget after widget1
- before=widget1 - pack this widget before widget1
- anchor='border' - where **border** must be n, ne, e, se, s, sw, w, nw, or center
- fill=NONE, or X, Y, BOTH
- padx, pady
- ipadx, ipady
- side='top', 'left', 'right', 'bottom'

- dynamically change label text
- change font of the label
  - 💡Find available font: ✔Control Panel ⇒ Apperance and Personalization ⇒ Fonts
  - font folder location
- add button
- button action > print text on console
- lambda expression on button action
- button, quit, icon, image

> In order to use PIL, you need install module Pillow

```
pip install Pillow
```

- use tkinter Entry

pack layout

- GUI Layout Management>pack
- pack(fill=tk.X)
- w.pack(fill=tk.X, padx=10, pady=5)
- ☝pack(fill=tk.X, pady=10)
- pack(padx=5, pady=10, side=tk.LEFT)

grid layout

- grid()
- Grid system
  - pack() and place() can be used together
  - grid() and place() can be used together
  - pack() and grid() cannot be used same time.

place layout

- Place system

icon and title

- Place system
-

Other widgets

- using Combox, dropdown box
- using checkbox
- using spinbox
- using text field
- using form
- radio button
- using radio button and message box
- using yes/no, ok/cancel message
- Same as above, use pack instead of grid
- using scrolled text
- Horizontal scrooled bar
- using progress bar
- enable/disable button
- using list box
- same as above with multiple selection
- using dialog
- using tk.Frame
- close window call back, display image
- using filedialog
- better version of tkinter22
- using save file dialog
- menubar = tkinter.Menu(root)
- slider (Scale)
- Scale: command
  😄👍Good website

color

- random background color
- built in color list

mouse

- mouse double click
- mouse position

table

- create a table

canvas

- using canvas draw pie chart
- draw line in canvas
- 👎😢data not available

tab window frame

- tab window

titled frame

- titled frame
- using frame
- open more than one window

plot chart in tkinter

- canvas.draw() chart on canvas
- plot chart dynamically

display image

- put image in frame
- display image on frame
- deal cards in window
- class MyFrame basic
- Calculator button on frame
- delete label
- display animated gif

popup window

- popup window

card game GUI

- card game gui

- Switch between frames
- use class switch frames
- Extends from Frame

sqlite DB

DB basic operations >> CRUD: Create, Retrieve, Update, Delete

- Add record to DB >> Create
- Load records from DB >> Retrieve
- Modify record >> Update
- Delete record >> Delete

Web Service API

Air Now Login
wangqianjiang/Qianjiang1122
API Key: 84B7917D-C980-407F-ACBC-B29E3D2E4458

- get all information
- selected information

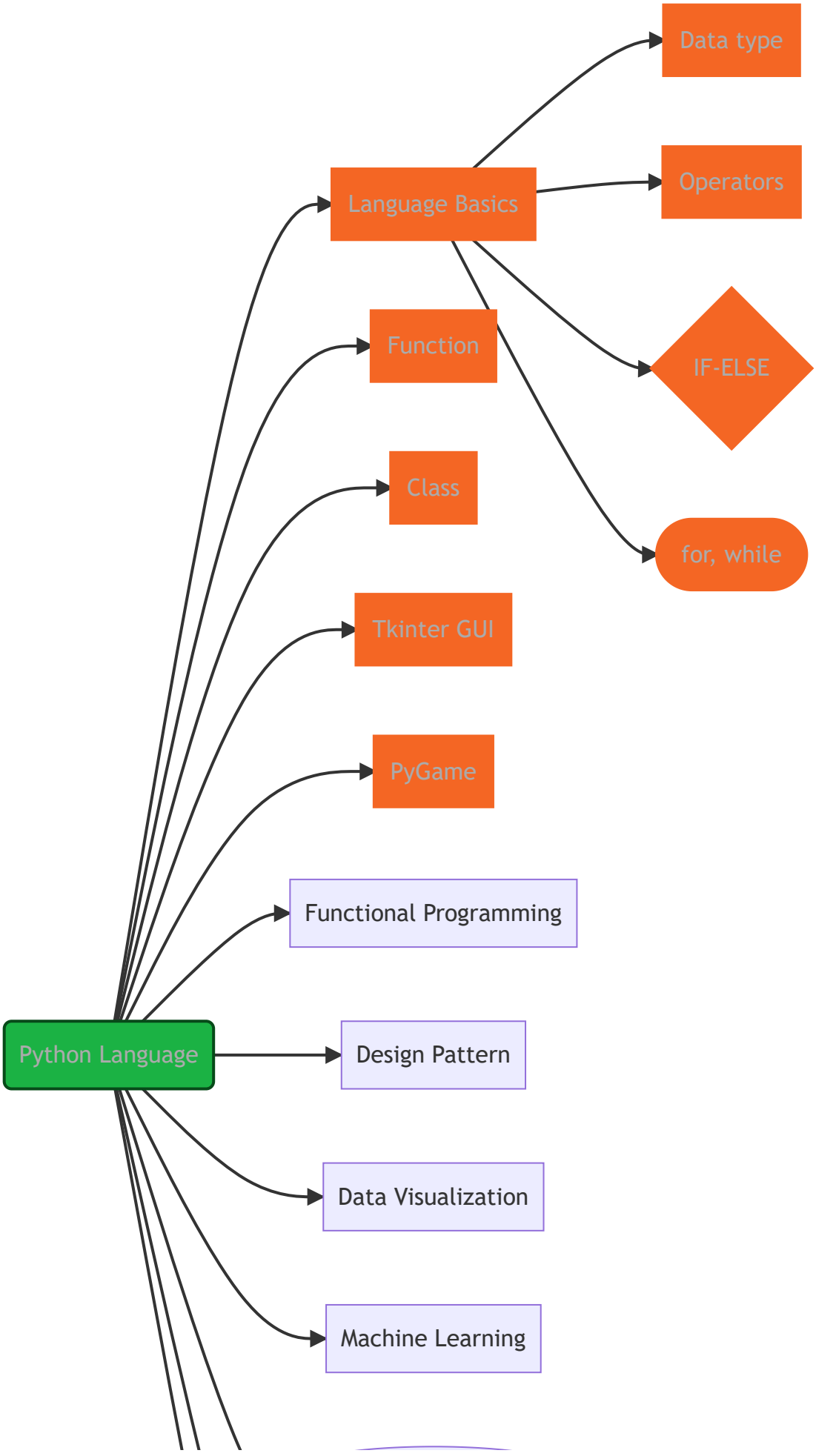
practices

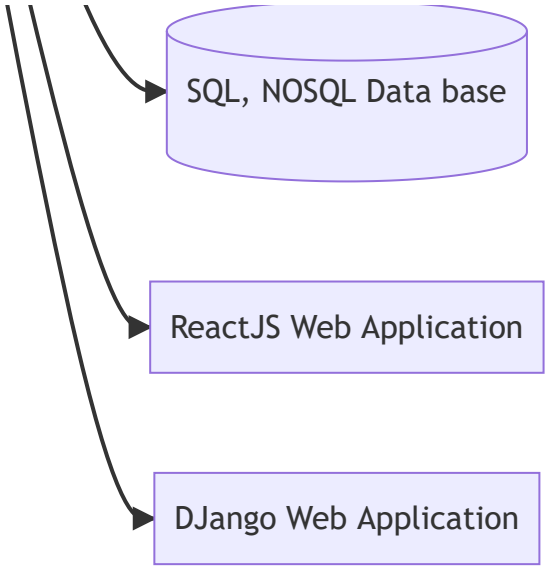
- do math
-

Application

- calculator
- tic tac toe
- Temperature Converter
- image viewer
- Date interval calculator
- Note Pad
- Sticker
- Blackjack

Application Development Process


Data type
Operators
Language Basics
IF-ELSE
Function
Class
for, while
Tkinter GUI
PyGame
Functional Programming
Python Language
Design Pattern
Data Visualization
Machine Learning



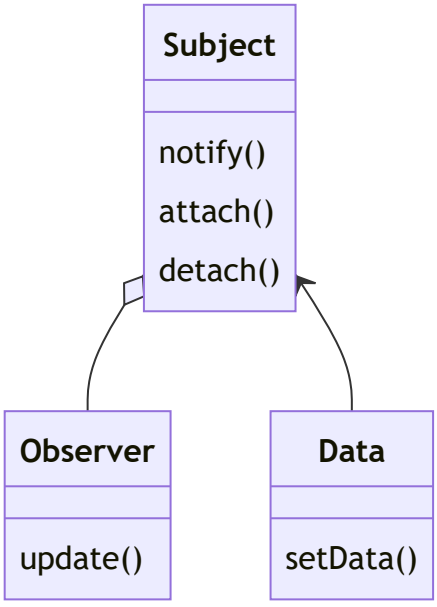
Knowing all python programing basics such as function, class does not make you a good programer.

- Sample Functional Programing
- Shopping actions
  Key point for functional programming is function binding and data wraper.

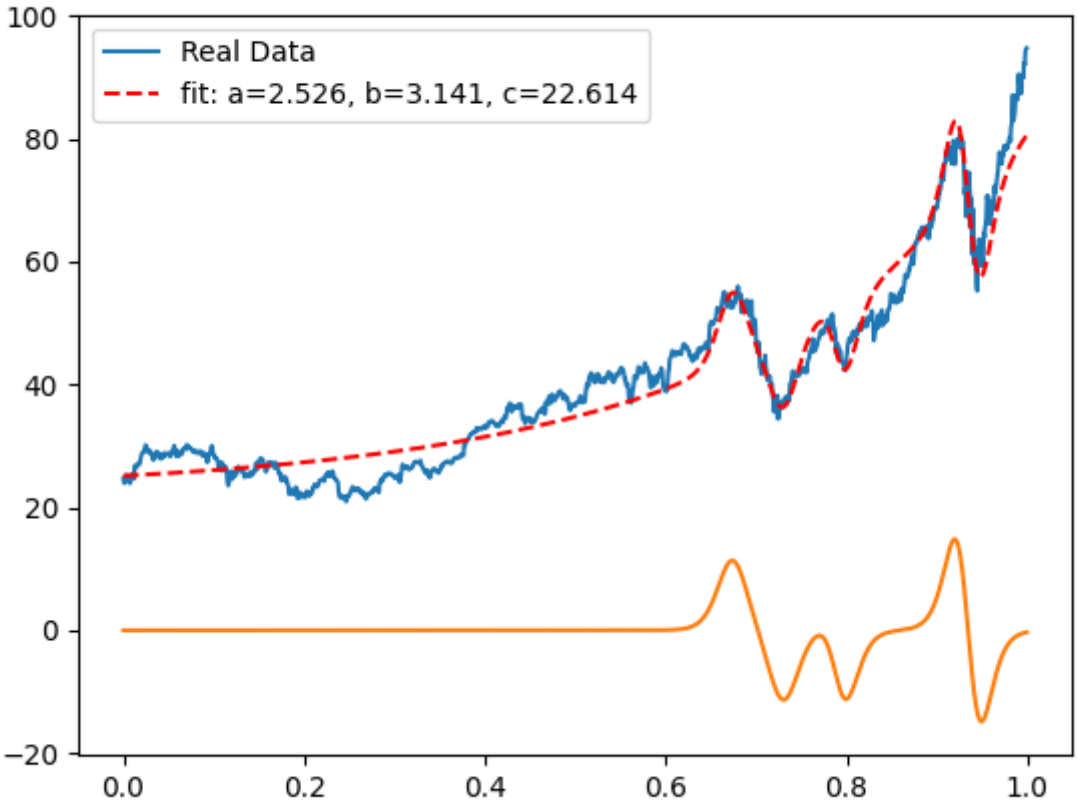
❓ What is design pattern

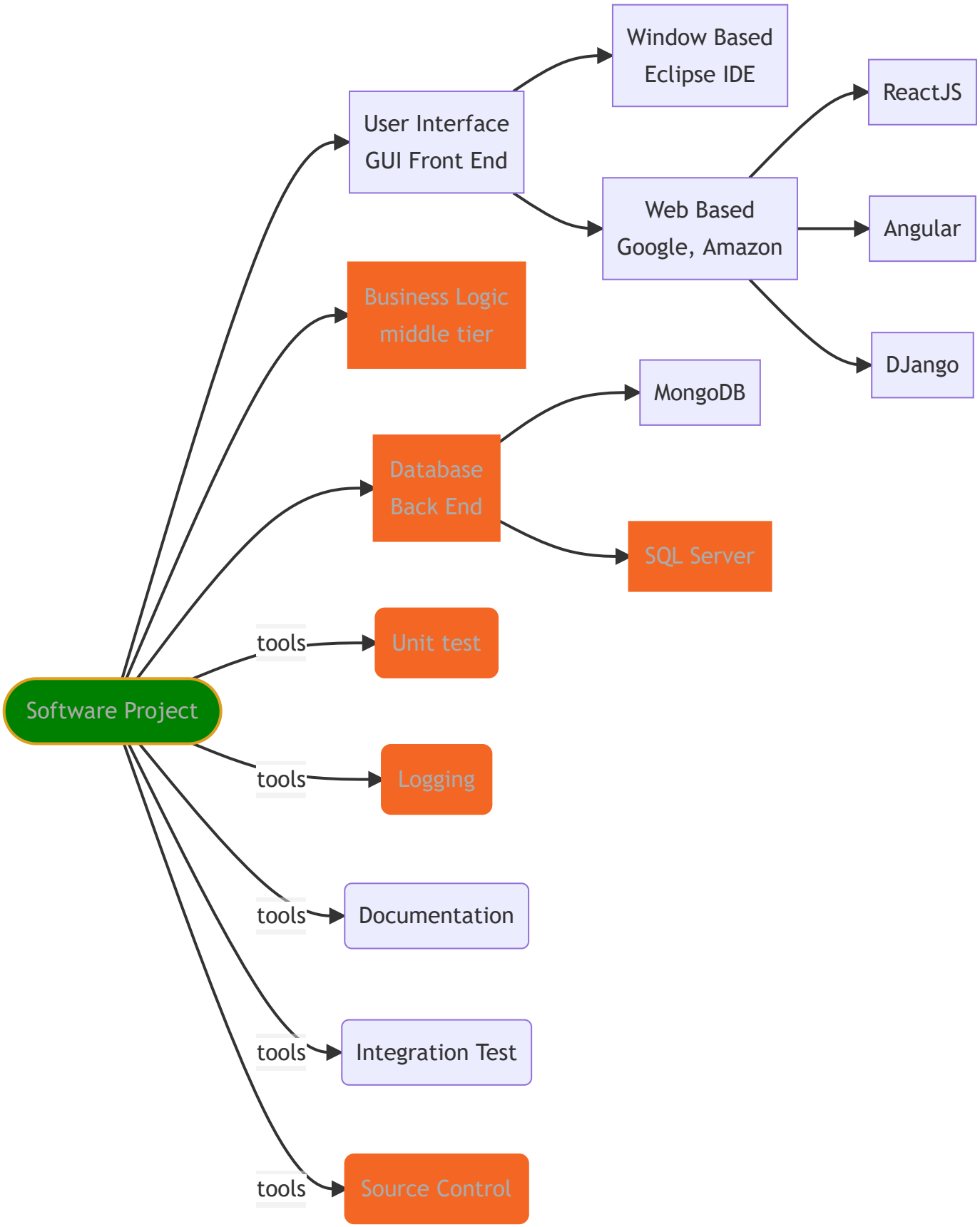
> ✔Design patterns can speed up the development process by providing tested, proven development paradigms. Design pattern provides special data type(class), functions and behavior and relations to solve generic problem.

- Design patterns
- Design pattern sample



- Machine Learning Sample


Real Data
fit: a=2.526, b=3.141, c=22.614
100
80
60
40
20
0
−20
0.0
0.2
0.4
0.6
0.8
1.0



PyInstaller

Tkinter to Exe

```
pip install pyinstaller
mkdir exe
cd exe
<copy python file to this folder>
pyinstaller -F -w temperatureConverter.py
```

隶书字体下载网站

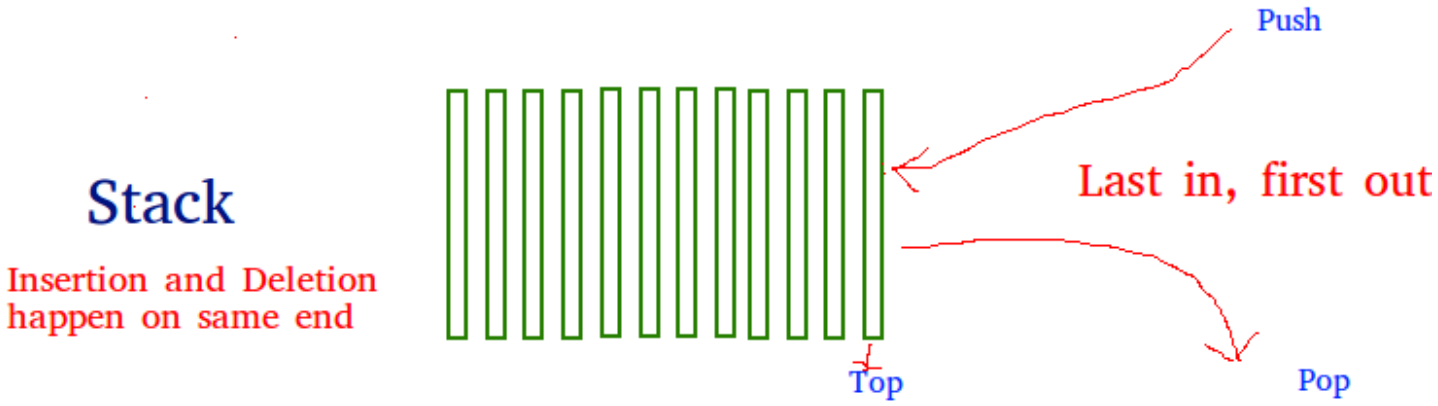
Data Structure

stack

❓What is stack?
✔a stack is an abstract data type that serves as a collection of elements, with two main principal operations: (LIFO) last in first out.

1. Push(), which adds an element to the collection, and
2. Pop(), which removes the most recently added element that was not yet removed.
   you have list of element, stack each other.
3. empty(), returns whether the stack is empty
4. size(), returns the size of the stack
5. top(), returns a reference to the top most element of the stack

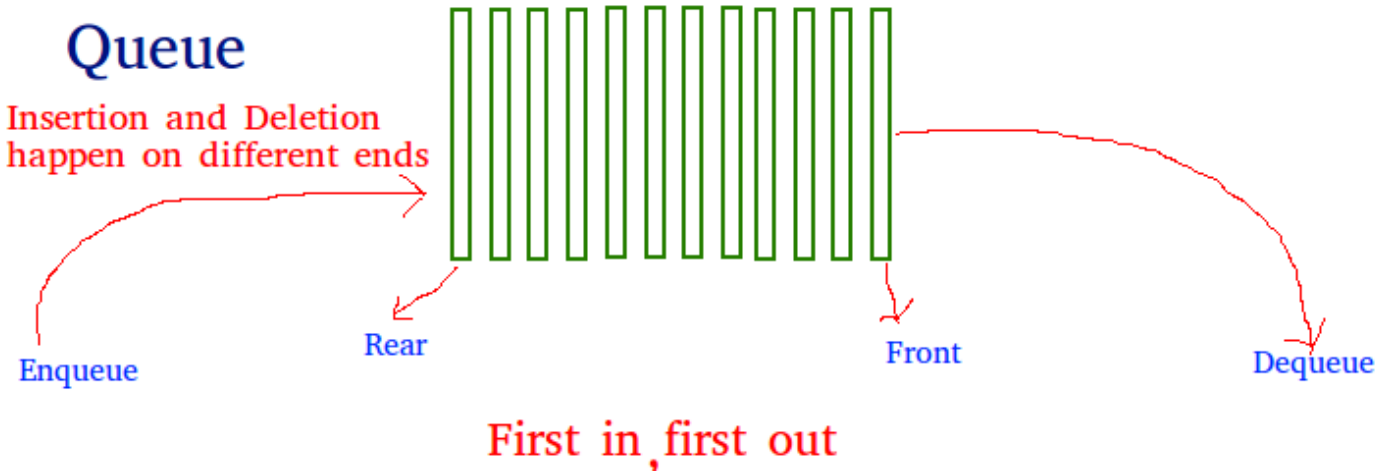
- stack.py



queue

❓What is queue?

✔A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO).



1. Enqueue: Adds an item to the queue. If the queue is full, then it is said to be an Overflow condition – Time Complexity : O(1)
2. Dequeue: Removes an item from the queue. The items are popped in the same order in which they are pushed. If the queue is empty, then it is said to be an Underflow condition – Time Complexity : O(1)
3. Front: Get the front item from queue – Time Complexity : O(1)
4. Rear: Get the last item from queue – Time Complexity : O(1)
5. push(): insert a new data into the queue
6. pop(): return the front data in the queue, and remove it from the queue
7. peek()/top(): return the front data in the queue without remove it

Priority Queue

❓What is priority queue?

✔Priority Queue is an extension of queue with following properties.

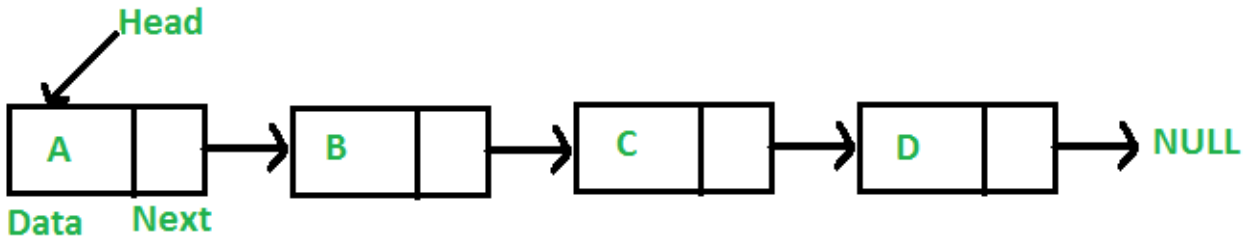
1. Every item has a priority associated with it.
2. An element with high priority is dequeued before an element with low priority.
3. If two elements have the same priority, they are served according to their order in the queue.

- priorityqueue.py
- priorityqueue2.py
- priorityqueue3.py

Linked list

❓What is linked list?

✔A linked list is a sequence of data elements, which are connected together via links.



Linked list

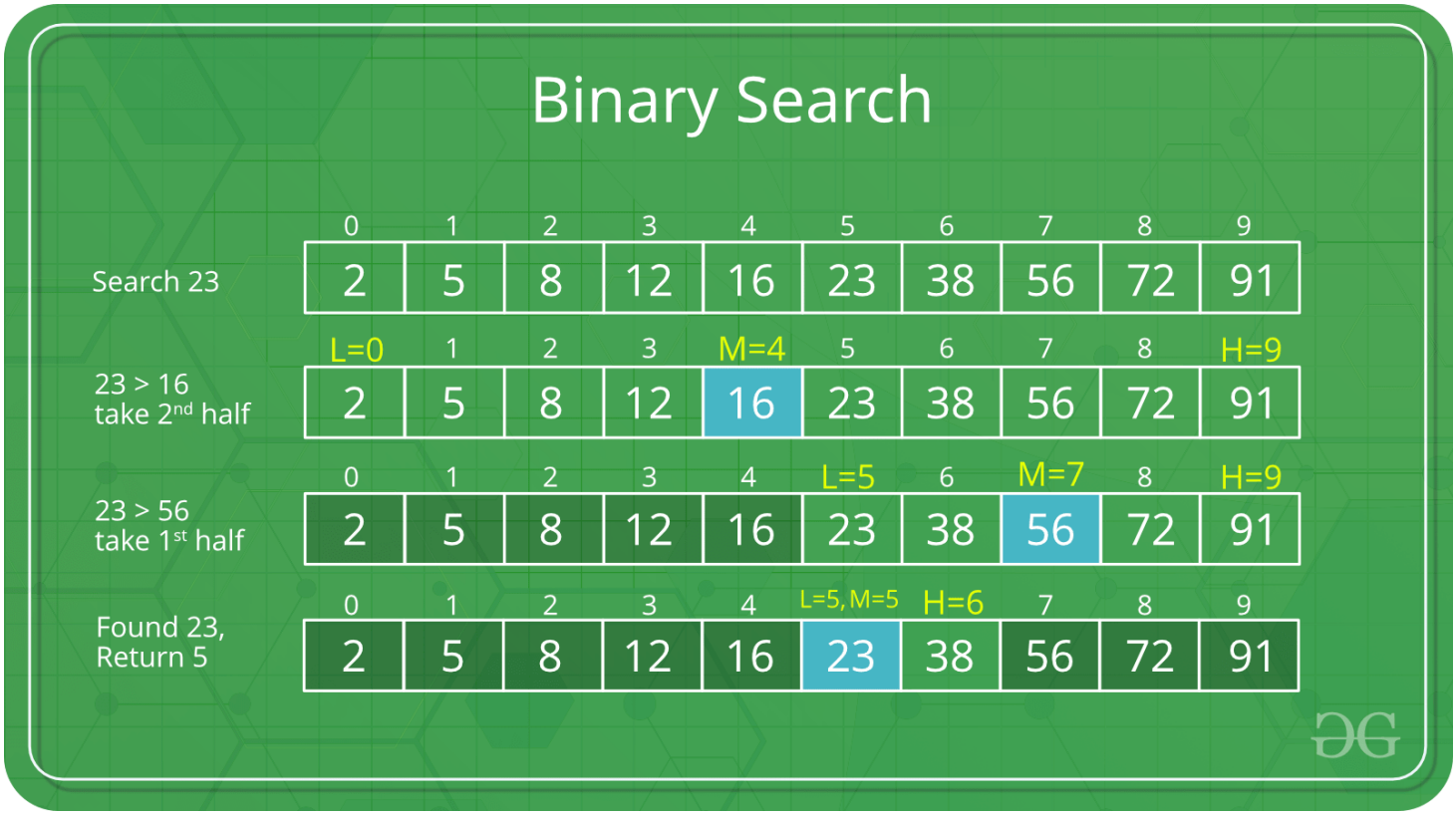
❓Why do I need use linked list?

✔Insert new node in list is expensive because all elements on right-hand side need to be shifted.

doubly linked list

doubly linked data list

binary search



- binary search recursive
- binary search while loop

insertion sort

Insertion Sort

selection sort

Selection Sort

quick sort

quick sort

Functional Programing

❓What is functional programming?
✔ functional programming is a programming paradigm where programs are constructed by applying and composing functions. functional programming is a programming paradigm where programs are constructed by applying and composing functions. Goal oriented

Function decorator(timer)

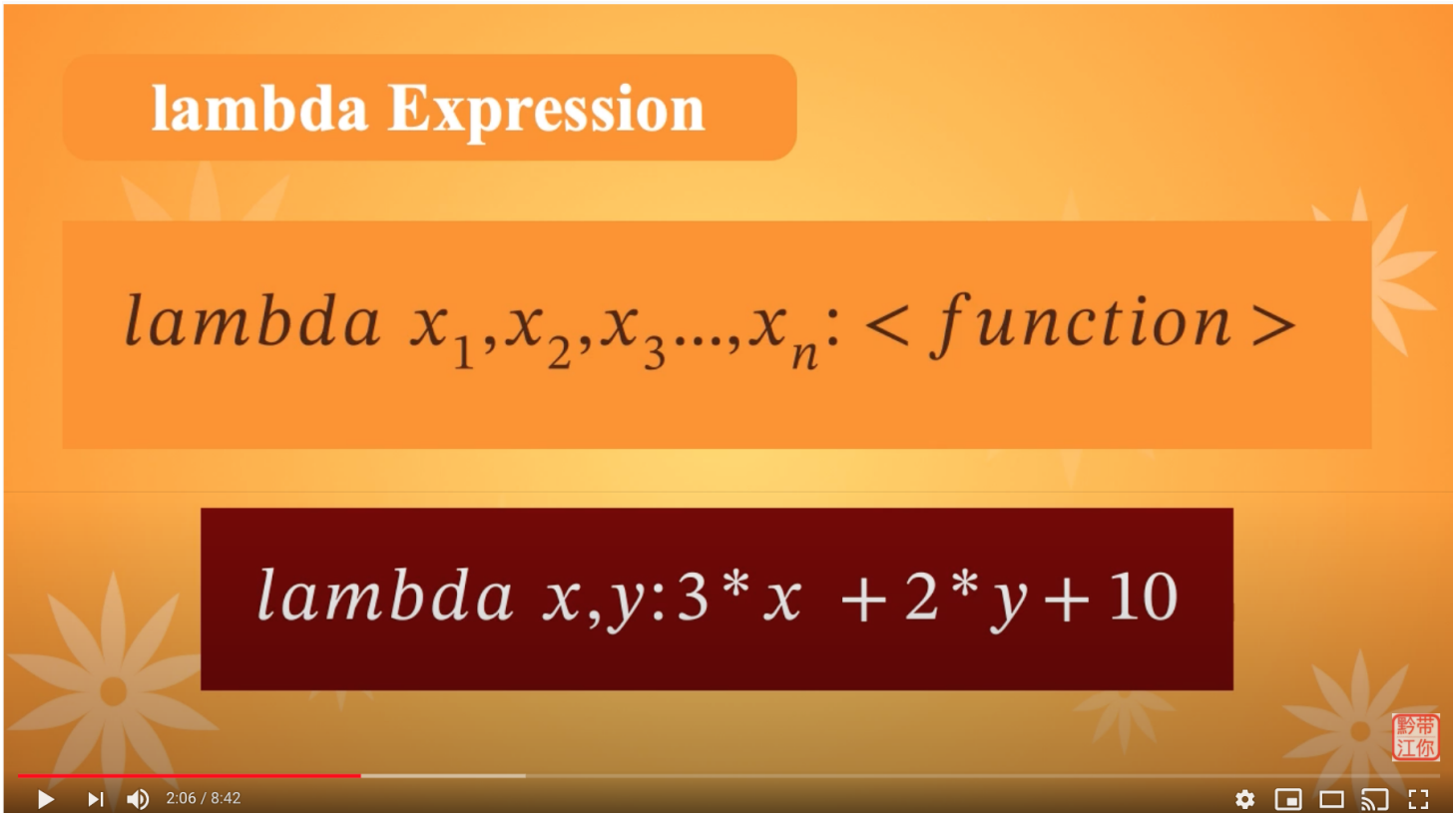
- Understand my wrapper function
- add decorator to any function
- pass function to class
- timer

Lambda expression

❓What is lambda function?
✔A Lambda Function in Python programming is an anonymous function or a function having no name.

- Syntax

```
lambda <variable list separated by comma>: expression
print(lambda x, y: x + y)
```

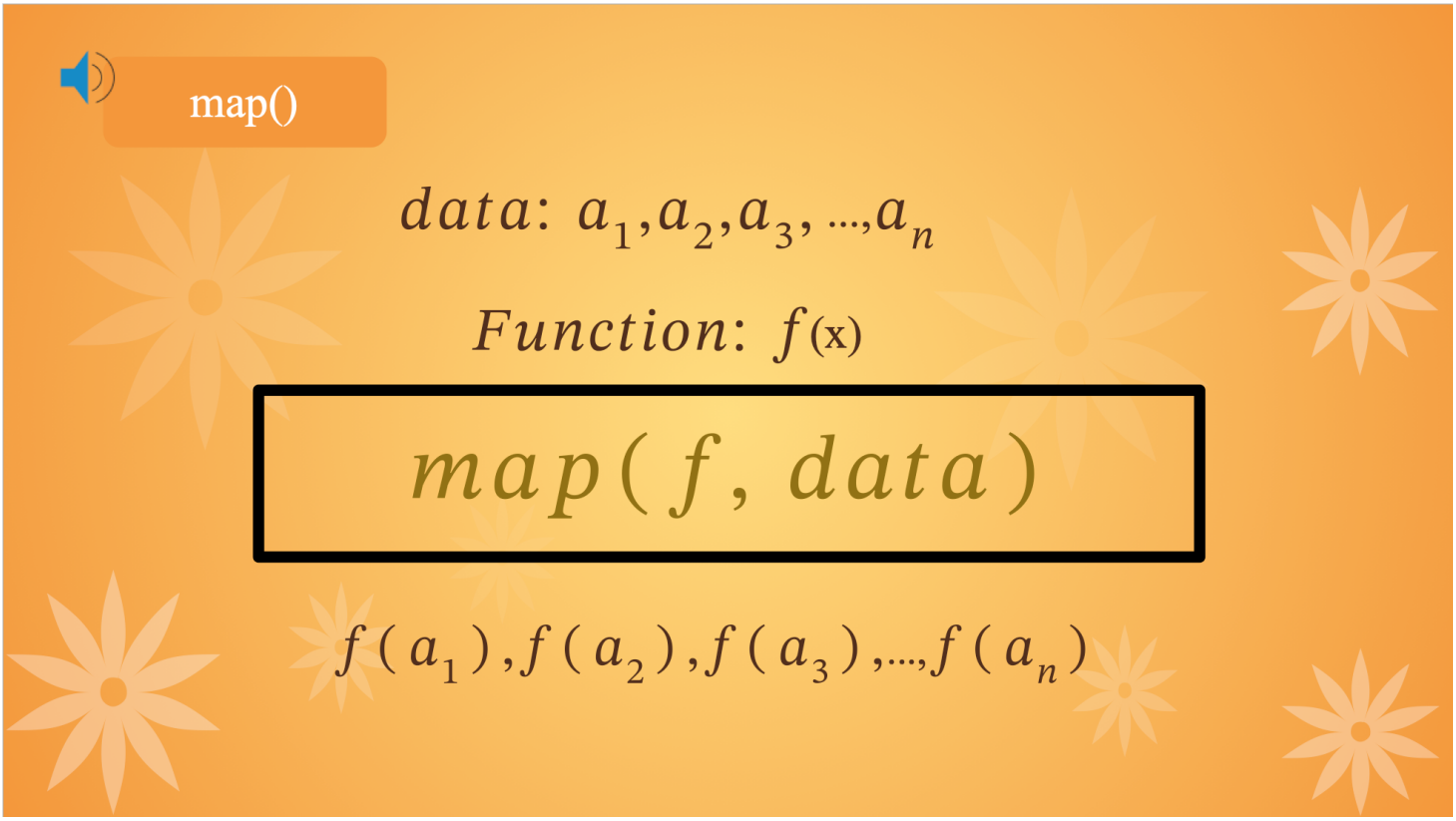


- map
-
-

map() function

❓What is map() function?

> ✔ the map() function is processing iterables without loop.




```
map(func, *iterables) --> map object
map(lambda x:x*x, list1)
map(lambda x,y:x+y, tuple1, tuple2)
```

1. map() takes two argument, a function and an iterable data
2. map() applys the given function to each element in the iterable
3. map() return a map object which is iterable and can be converted to list or tuple

map 2 list to one
convert temperature
understand map() function
map with multiple iterables
convert temperature
map with multiple iterables
map vs. for-loop

filter() function

❓What is filter() function?
✔Return an iterator yielding those items of iterable for which function(item) | is true. If function is None, return the items that are true.

```
filter(function or None, iterable) --> filter object
filter(lambda x:x%2==0, list1)
```

- filter object is iterable and can be converted to list or tuple
  get all prime number from 2 to n
  func.py
  lambda02.py

reduce() function

❓What is reduce() function?
✔The reduce(fun,seq) function is used to apply a particular function passed in its argument to all of the list elements

- sum over iterable
- find max and min
- multiply all elements

sort() function

❓How to use sort function?

✔sort dict

- sort car dict by year

✔sort list by lambda expression

- sort temperature by city
- reverse sort by temperature

✔sort list with tuple element

- sort list

zip() function

- Combine city with temperatur
- combine list
- matrix operation
- understand zip() function

Calculate Square root

$$a_{i+1} = \frac{(a_i + \frac{n}{a_i})}{2}$$

$$a_1 = f(a_0)$$

$$a_2 = f(f(a_0))$$

- Function chain

Non-strict evaluation

- Lasy or non-strict evaluation
- pure function
  ❓ What is a pure function?

> ✔A pure function is a function whose output value follows solely from its input values, without any observable **side effects**.

❓ Why use functional programming?
✔ The functional paradigm is popular because it offers several advantages over other programming paradigms.

1. **High level**: You're describing the result you want rather than explicitly specifying the steps required to get there. Single statements tend to be concise but pack a lot of punch.
2. **Transparent**: The behavior of a pure function depends only on its inputs and outputs, without intermediary values. That eliminates the possibility of side effects, which facilitates debugging.
3. **Parallelizable**: Routines that don't cause side effects can more easily run in parallel with one another.

❓ How Python support functional programming?
✔two features:

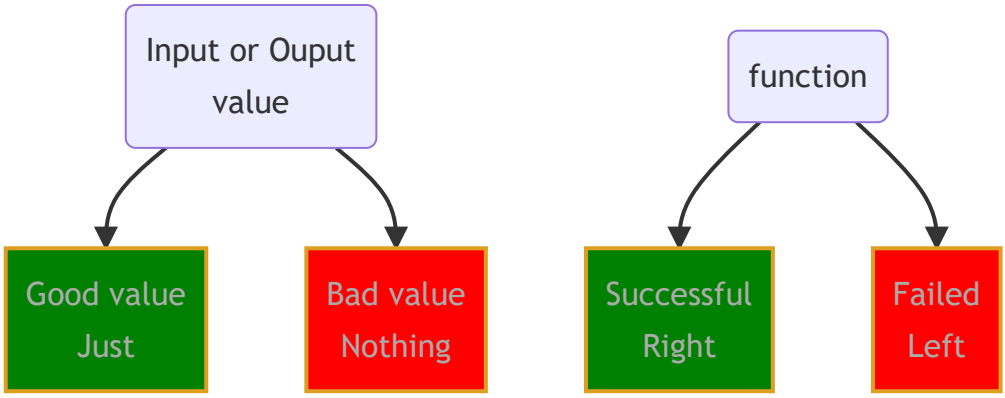
1. To take another function as an argument
2. To return another function to its caller

monad

❓ Why use monad?

✔ to solve very common program issues

1. Null pointer exception.
2. function call failure.



- None value and function failure

```
def neg(num):
  return -num

x = 10
y = str(neg(int(x)))
```

if x is not an intege string, cause software blows up.

- avoid function call failure
- blows up
- List monad

❓What is monad?

> ✔ A monad is a design pattern that allows us to add a context to data values, and also allows us to easily compose existing functions so that they execute in a context aware manner.
> ✔a monad is an abstraction that allows structuring programs generically.

- In English dictionary, monad is
  ✔Unit, one; Atom sence;
  ✔an elementary individual substance which reflects the order of the world and from which material properties are derived

❓ How does the monad solve those two issues?

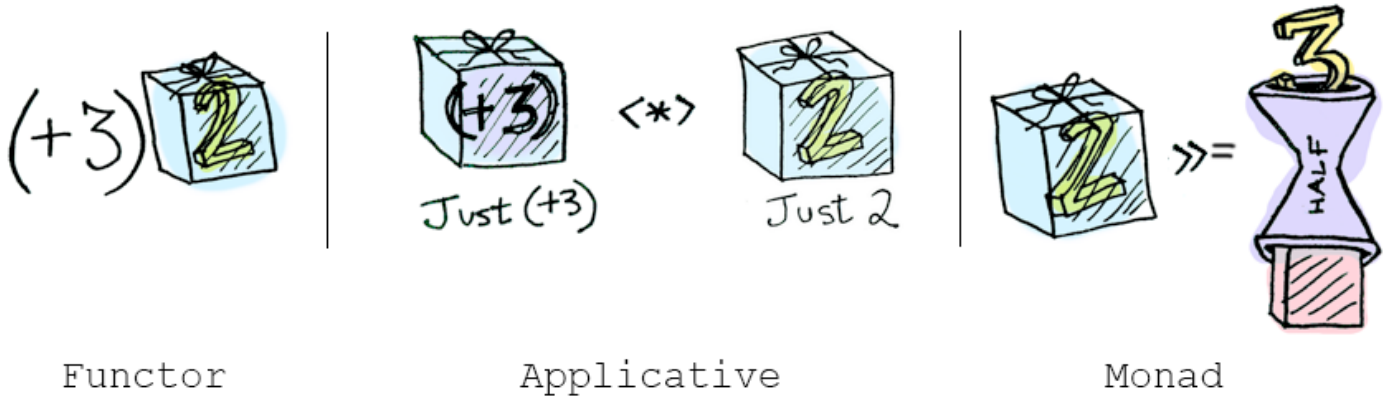
✔ box value, return either

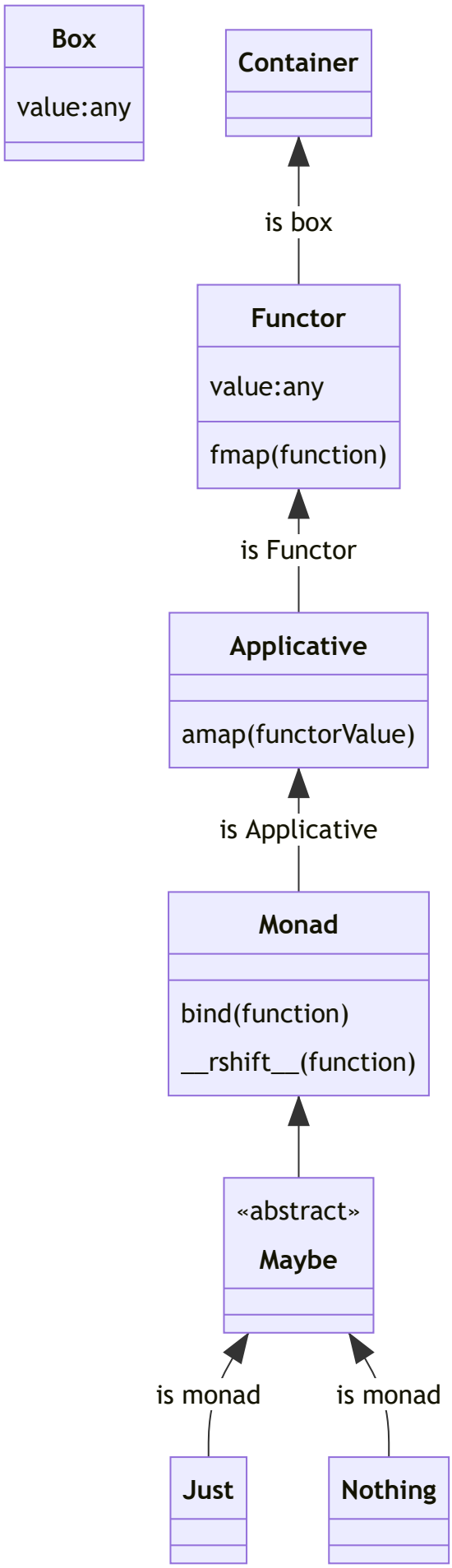
- kleisli Compose

Definition of Monad

✔Wrapper Class type with implementation of fmap(), amap() and bind() functions.

- One to wrap values of any basic type within the monad (yielding a monadic value);
- Another to compose functions that output monadic values (called monadic functions).




Box
value:any
Container
is box
Functor
value:any
fmap(function)
is Functor
Applicative
amap(functorValue)
is Applicative
Monad
bind(function)
__rshift__(function)
«abstract»
Maybe
is monad
is monad
Just
Nothing

- Functor.py

❓ What is functors?
✔️you apply a function to a wrapped value using map
implement fmap() function: functor map

❓what is applicatives?
✔️you apply a wrapped function to a wrapped value using apply, if defined
implement amap() function: applicative map

❓what is monads
✔️you apply a function that returns a wrapped value, to a wrapped value using flatMap.
implement bind() function: binding function
✔️ Override binding operator (*, >>)

- Understand functor, applicative, monad
- Functor >> bind
- Applicative
- Applicative
- Applicative
- .then() fuction chain for applicative
- ReactiveX
- 😟monad01.py
- ✔️operate a founction to a boxed value
- Understand Functor, applicative, monad
- Monad >> bind
- Compose funtion chain
- Compose function chain head tail
- @curry function decorator
- Use ListMonad create list, then do then() or map()
- applicative handle error
- ListMode and monad bind()
- Monad bind
- Use Maybe solve the None issue
- Maybe.maybe() function
- How applicative works
- process to write code in VSCode
- Shopping function list
- observer pattern

- map shopping functions

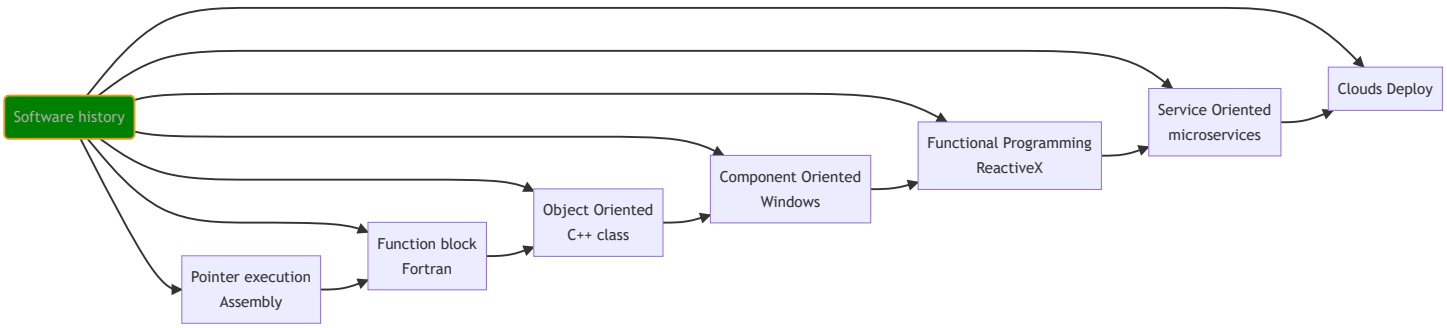
❓ what is fmap?

> ✔Applies 'function' to the wrapper box value and returns a new wrapper box value
> ✔We have a polymorphic type f, and fmap gives us the ability to:

1. Liberate a pure value from the type constructor that refers to it
2. Call a function on it, which could return a result of a different type
3. Have the type constructor refer to the type of the result

❓ what is amap?

> ✔Applies the function stored in the functor to the value of 'functorValue', returning a new wrapper box value.



Either

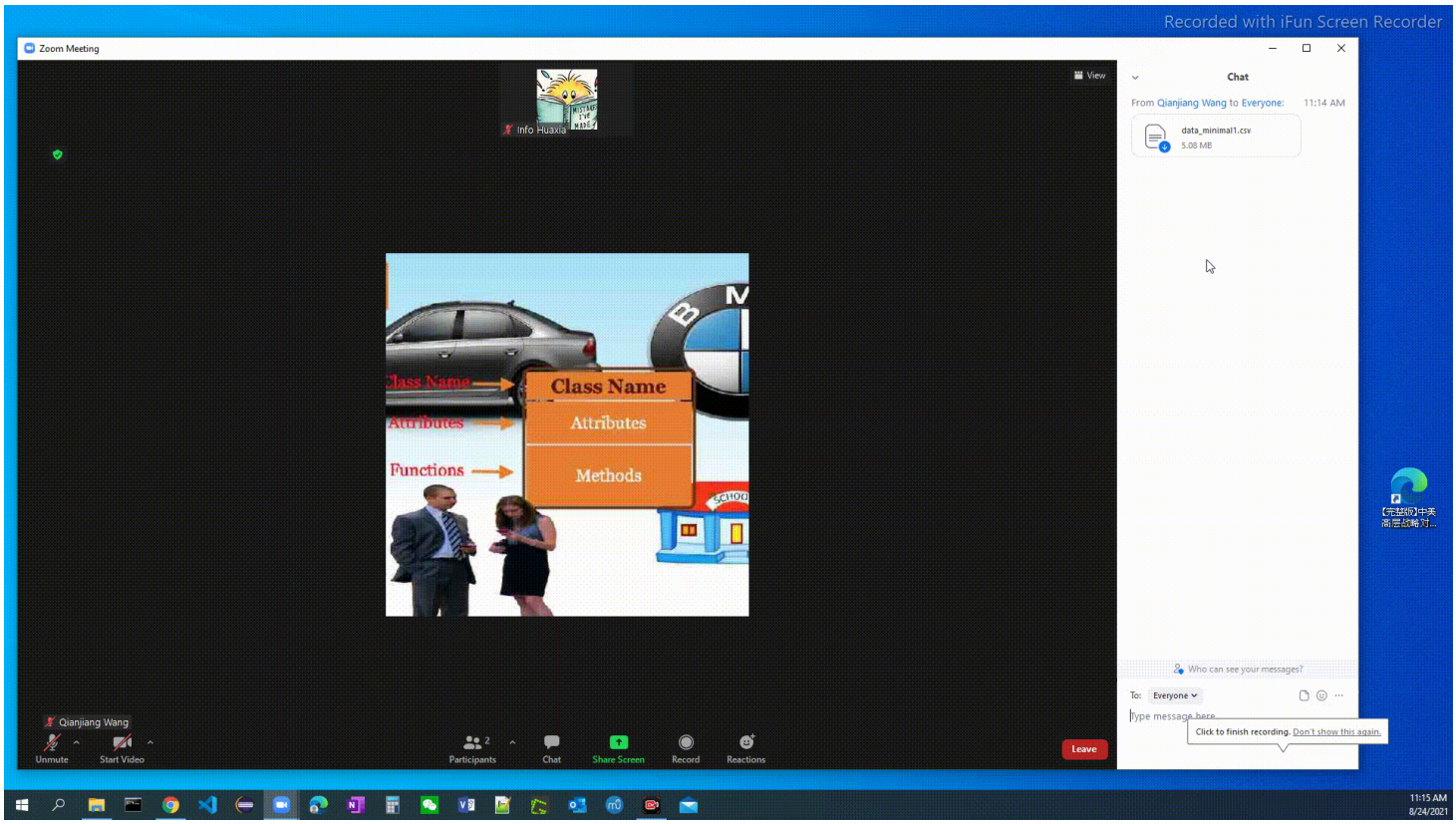
❓ Any application service
✔

-
- check a even number
- return Right/Left
- bind functions
- circle area Either

❓ what is flatmap?

✔FlatMap differs from Map in a key way. Instead of the Map operation returning an Option automatically, it instead requires that the function passed to it return an Option type result itself. That is, while Map returns an Option, FlatMap returns a value (or None value) for each option, regardless of whether it is a “something” or a “nothing” and requires that the input method applied return an Option type response.

- 👎Understnd map vs flatmap
- Understand monad map vs flatmap
- mymonad: Some or Nil
- pymonad implementation
- map() function as Functor
- Understand Either and flat_map
- simulate aireline tickets



Design pattern

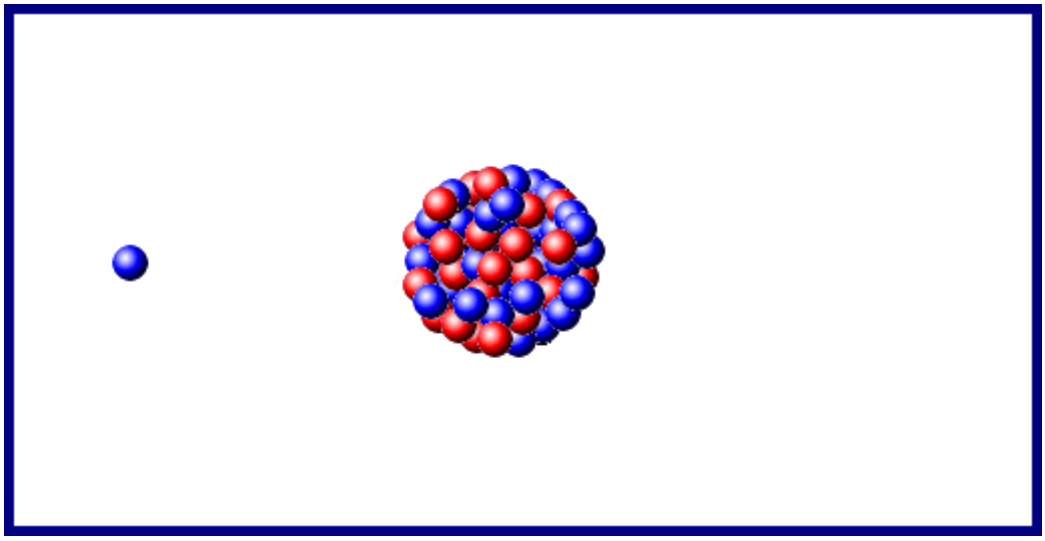
Reactivex design pattern

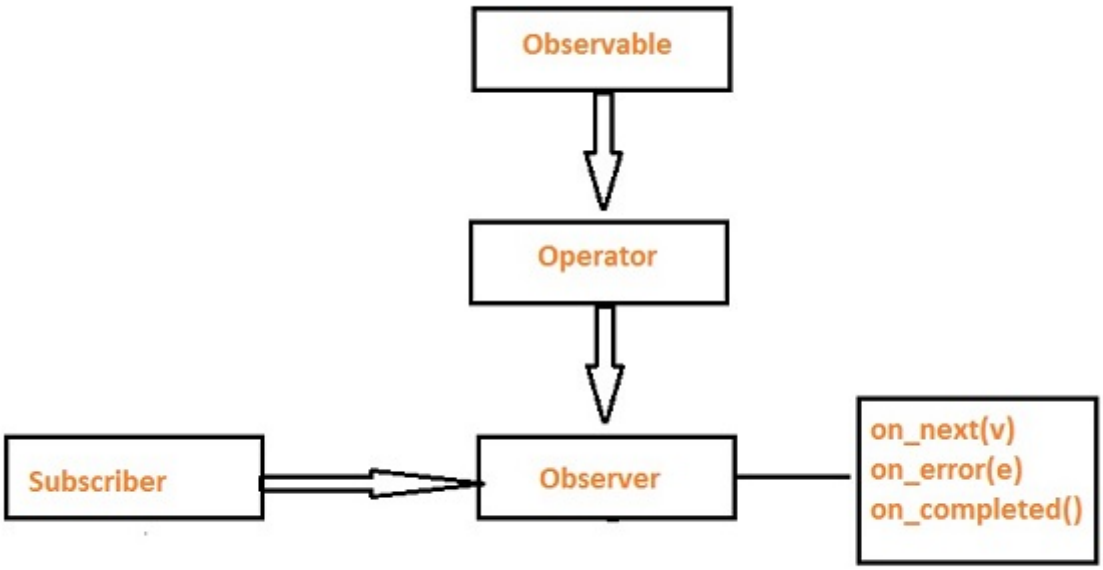
❓ What is ReactiveX?

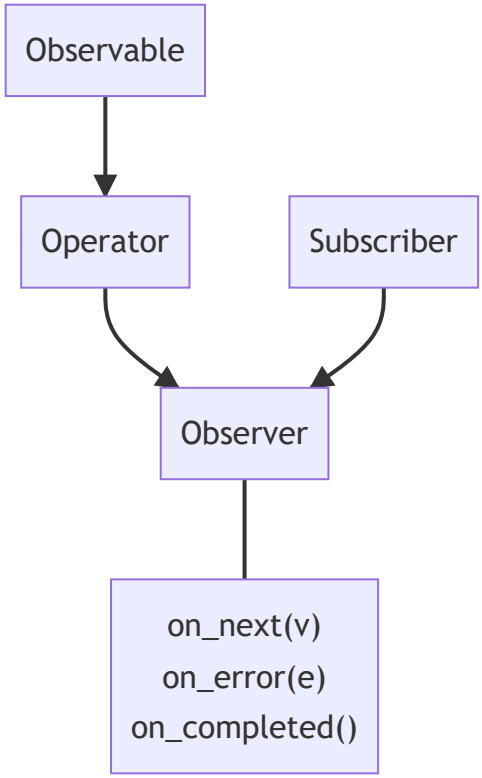
✔Reactive programming is a programming paradigm, that deals with data flow and the propagation

of change. It means that, when a data flow is emitted by one component, the change will be propagated to other components by a reactive programming library. The propagation of change will continue until it reaches the final receiver.

ReactiveX design pattern







- use 'of' to create observable
- use 'create' to create observable
- use 'pipe' function and operator
- Define processor first, pass data later

where RxPY offers operators such as

- Mathematical operators (average, concat, count, max, min, reduce,sum)
- Transformation operators (buffer, ground_by, map, scan)
- Filtering operators (distinct, element_at, filter, first, ignore_element, last, skip, take)
- Error handling operators (catch, retry)
- Utility operators (delay, time_interval, timeout, timestamp)
- Conditional operators
- Creation operators
- Connectable operators
- Combining Operators

❓What is Observable?

> ✔ An observable is a function that creates an observer and attaches it to the source having data streams that are expected from, for example, Tweets, computer−related events, etc.
> ✔ Data Stream

❓ What is Observer?

> ✔ It is an object with on_next(), on_error() and on_completed() methods, that will get called when there is interaction with the observable i.e. the source interacts for an example incoming Tweets, etc.
> ✔ Who process Data Stream

❓ What is Subscription?

> ✔ When the observable is created, to execute the observable we need to subscribe to it.
> ✔ trigger above process

❓ What are Operators?

> ✔ An operator is a pure function that takes in observable as input and the output is also an observable. You can use multiple operators on an observable data by using the pipe operator.

❓ What is Subject?

> ✔ A subject is an observable sequence as well as an observer that can multicast, i.e. talk to many observers that have subscribed. The subject is a cold observable, i.e. the values will be shared between the observers that have been subscribed.

❓ What is Subscription?

> ✔ When the observable is created, to execute the observable we need to subscribe to it.

❓ Advantages of using RxPy?

> ✔the following
>
> - RxPY is an awesome library when it comes to the handling of async data streams and events. RxPY uses observables to work with reactive programming that deals with asynchronous data calls, callbacks and event-based programs.
> - RxPY offers a huge collection of operators in mathematical, transformation, filtering, utility, conditional, error handling, join categories that makes life easy when used with reactive programming.
> - Concurrency i.e. working of multiple tasks together is achieved using schedulers in RxPY.
> - The performance is improved using RxPY as handling of async task and parallel processing is made easy.

❓ Disadvantage of using RxPY

> ✔ Debugging the code with observables is a little difficult.

18 modules

web development

1. requests
2. [flask]
3. django

```
pip install django==4.0
```

1. [twisted](realtime game)
2. [BeautifulSoup]
3. selenium

Data Science

7. numpy YouTube
8. [pandas](great for reading data) YouTube
9. [matpotlib](data visualization)Matplotlib Tutorial (Part 1)
10. [nltk](natural link tool kit)
11. [opencv](image data)
12. [tensorflow]
13. [keras]
14. [pytouch]
15. [scikit]
16. [kivy](platform independent app)

GUI Window Application

17. [pyqt5]
18. [tkinter]

Game

19. [pygame]