

Python Language

Python programming from scratch.

Markdown Shared Library

- [My First python program](#)
- [print](#)
- [comment](#)
- [Variable Naming](#)
 - [Variable and memory](#)
- [Data Type](#)
- [operator](#)
- [Execution Control \(If-else\)](#)
- [Loop](#)
- [Function](#)
- [OOP \(Object Oriented Programming\) and class](#)
 - [class basic](#)
 - [attribute scope](#)
 - [dunder functions](#)
 - [class tricks](#)
 - [class inheritance](#)

My First python program

[hello world](#)

print

[print](#)

- place holder (%s, %d, %f)
- print with tuple
- formatted print: `print(f"x={x}")`

comment

comment

- single line comment: #
- multiple lines comment: """ , ""

Table of Contents

Variable Naming

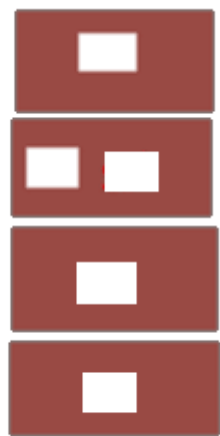
1. variable name cannot start with number
2. variable can be combination of letters and numbers `_`, `a~z`, `A~Z`, `0~9`, ⚡ no other special characters
3. don't use reserved keywords as variable name

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.
5. 🖱class name, function name and attribute name, all of them must follow the rules above🖱.

Variable and memory



Hello

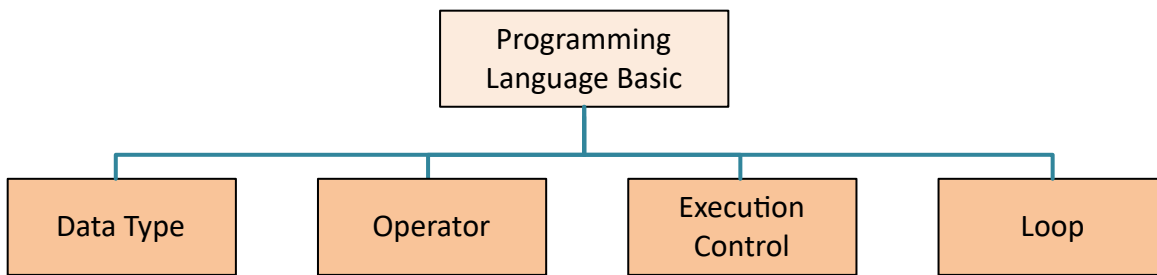


Table of Contents

Data Type

- **Numbers**
 - int: a=4
 - float: a=3.4
 - complex: c=4-3j
- **String**
 - string is iterable
 - string slicing: `start[:end]:step`
 - String operator +, *
 - as function `str(object)`
 - string functions
- **Tuple**
 - tuple is iterable
 - tuple is immutable
 - tuple slicing: `tuple1[start[:end]:step`
 - tuple operator +, *
 - as function: `tuple(iterable)`
 - tuple functions ()
- **List**
 - list is iterable
 - list is mutable
 - list slicing: `list1[start[:end]:step`
 - list operators +, *
 - modify list
 - as function: `list(iterable)`
 - list functions (`append`, `insert`)
- **Set**

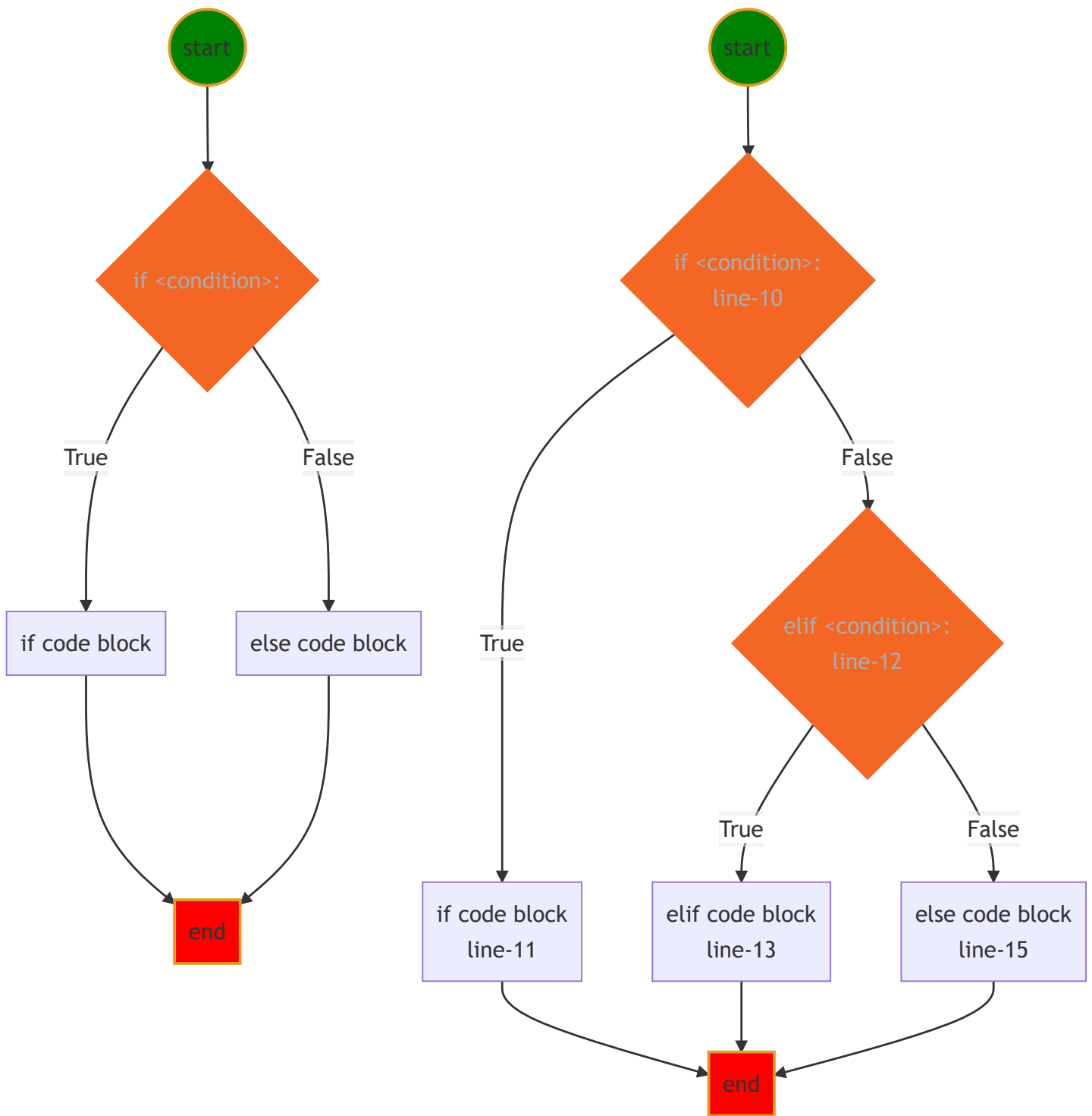
- set is iterable
- set is mutable
- set operators: &, |, <, >, ==
- modify set
- as function: set(iterable)
- set functions ()
- [Dictionary](#)
 - iterable
 - mutable
 - no duplication
 - ** operator
 - function (items, keys, values, clear, pop)

operator

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

Execution Control (If-else)

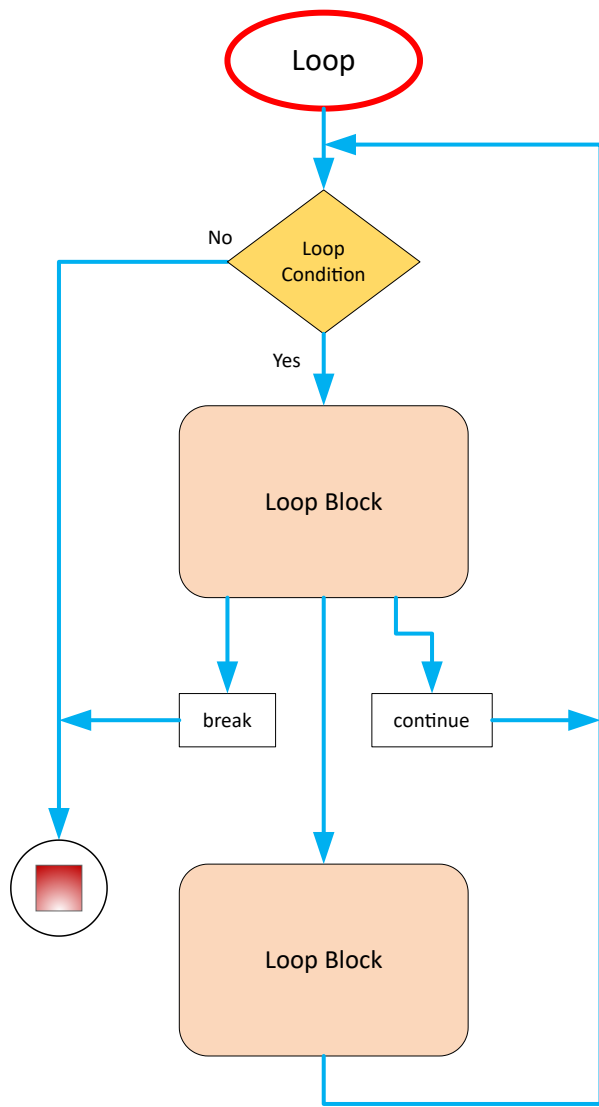
Execution control



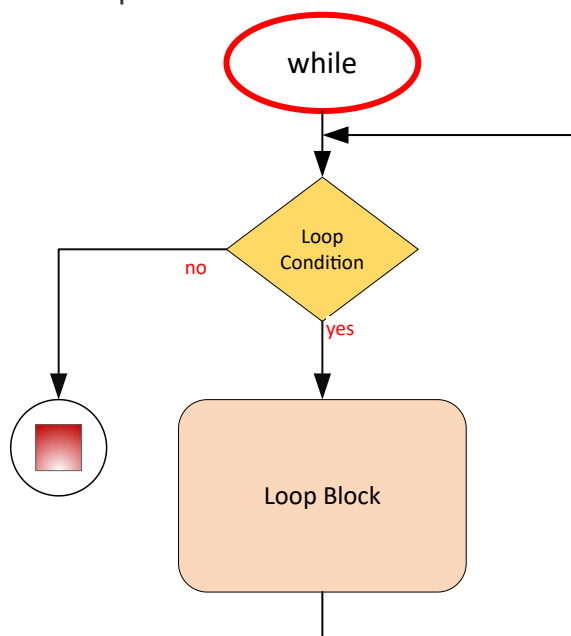
- If without else
- if with elif and else

Loop

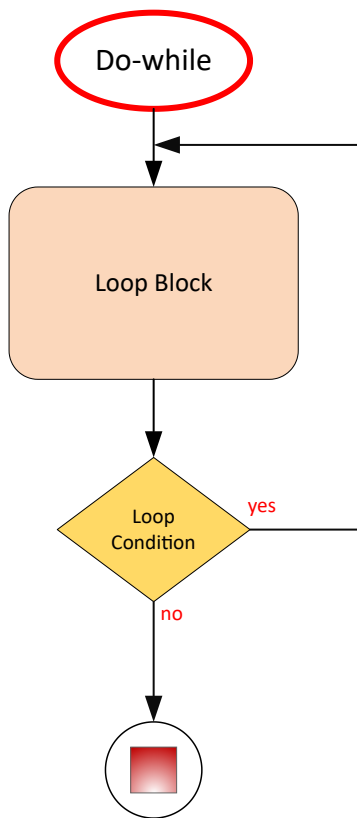
- For loop



- for/while loop
- While loop

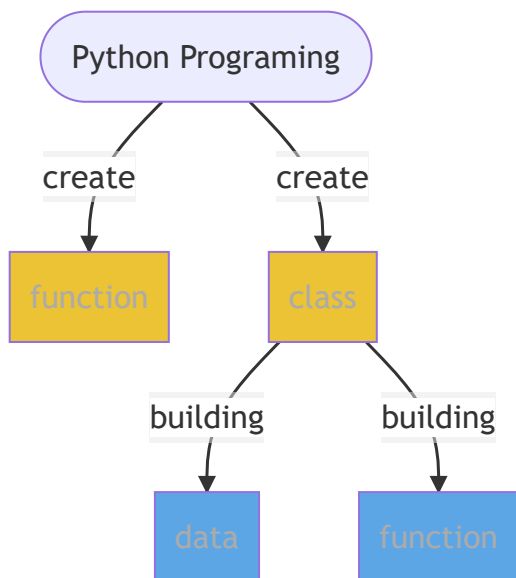


- Python does NOT support do-while loop, but you can simulate do-while.



while loop has 3 part:

1. initialize variable, `a=0`
2. variable condition, `a<10`
3. adjust variable, `a +=1`



Function

A function is a block of organized, reusable code that is used to perform a single, related action.

- def: use Python reserved keyword
- function name: you can name a function whatever you want but follow the variable rules.
- () you have to include () pair in you function definition
- : must end your definition with :.
- the function body must indent
- ⚡ function can be overridden
- 😊 return more than one value
- 💡 Single response, do single thing
- 📞 call a function by function name and () no matter it has arguments or not, and arguments if

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

- [function.py](#)
- [argument.py](#)
- [raise error when radius<0](#)
- [understand if name == 'main':](#)

😊 avoid running test code block from import

- [add try-except block](#)
- [Define inner functions inside outer function](#)
- [return function dynamically](#)

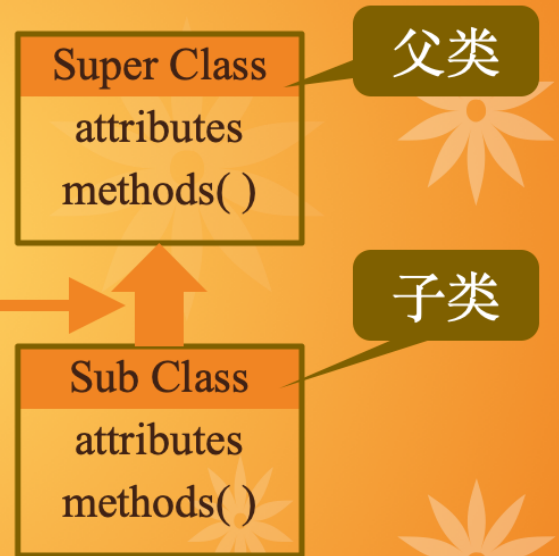
part of Functional programming which focus on goal

- [Functional programming basic](#)

OOP (Object Oriented Programming) and class

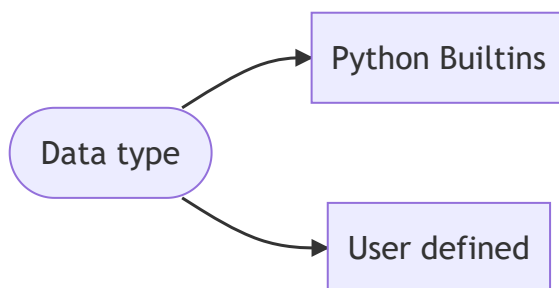
Object Oriented Programming

- Encapsulation 封闭性
- Abstraction 抽象性
- Inheritance 继承性
- Polymorphism 多样性



? What is class?

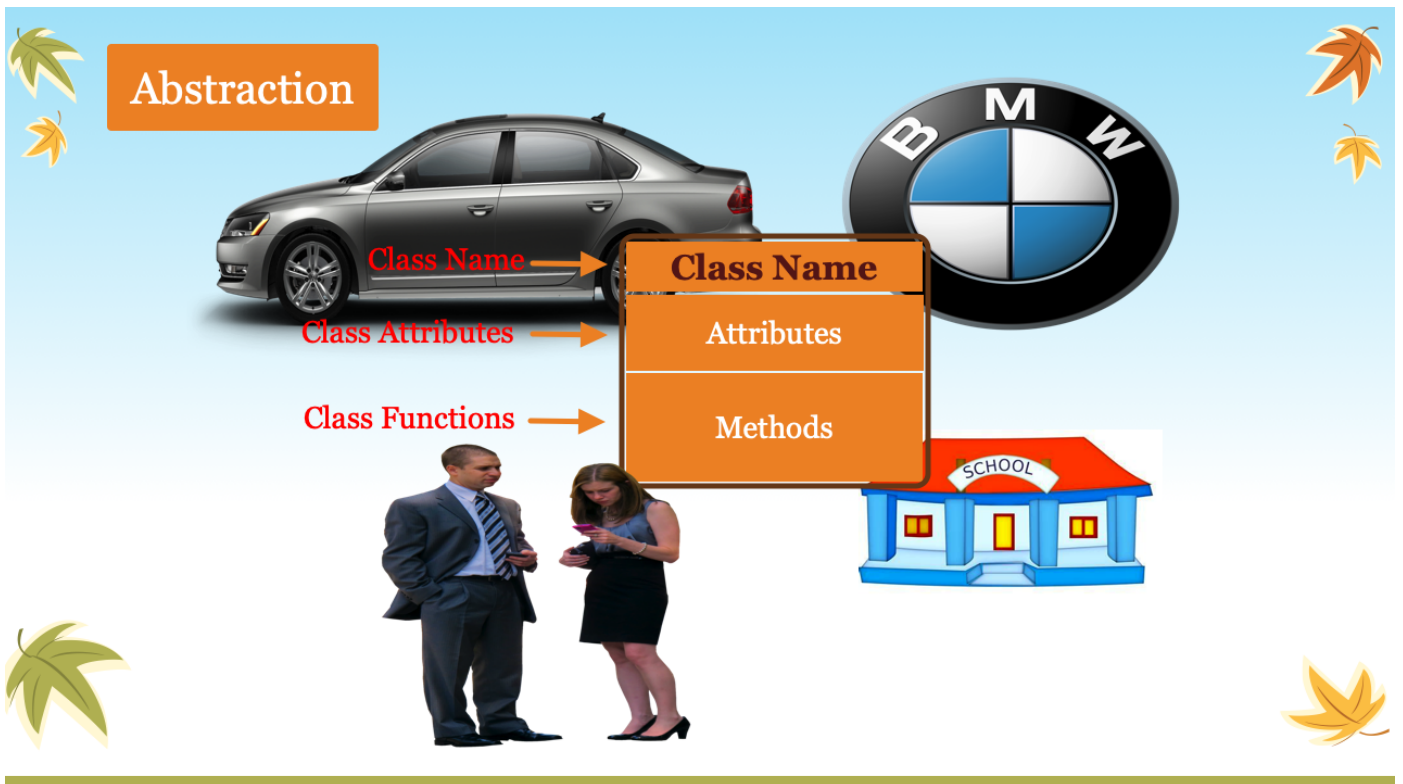
Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its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



There are 4 features in OOP

1. Abstraction: create class which abstract from the object in the real world.
2. Inheritance: a class can inherit from multiple other classes to increase code reuseability.
3. Encapsulation: avoid data or function to be accessed outside unintentionally.
4. Polymorphism: same function behavior differently by different object type which inherit from same class.

- Abstraction



class basic

- Simplest class
- say Hello outside of the class
- Robot now can introduce himself
- Define constructor: **init()** with initial name
- make the inputName as keyword argument
- make private attribute `__energy`
- build getter, setter, and property

attribute scope

- class level attribute vs. instance level attribute
- Solve the class level attribute issue

dunder functions

- compare **repr()** most time, and **str()**
 - **repr()** function can only return str type, or need to convert it to str.
- play **new()** with **init()** is used to initialize object
- return instance from other class (override **new()**)
- override **iter**, **next** function
- start from 1 include stop, override **iter**, **next**

- `class Card`, override `add`, `gt`, `lt`, `eq`

class tricks

- define class function outside, use function in different class
- internal function call another internal function
- class level function and attributes
- inherit from Enum, callable
- multiple inheritance
- Nested class
- Car has Engine
-

class inheritance

? What inherit means?

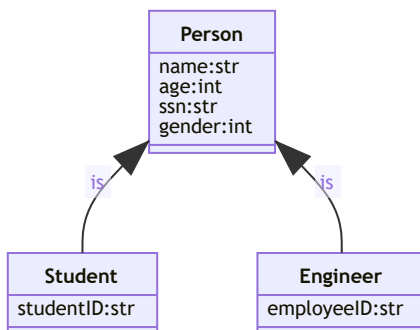
✓ to be born with the same physical or mental characteristics as one of your parents or grandparents:

Definition of Inheritance in Software world: A subclass inherits all of the features including attributes and functions from superclass.

? How do I inherit from a superclass?

✓ see following code below

```
class Subclass(Superclass):
    pass
```



- Person is a super class
- Student is a person
- Engineer is a person

? How do I initialize instance level attribute?

✓ 1. initialize the instance level attribute in `__init__()`; 2. define instance level attribute in setter.

- [class inheritance](#)

1. If subclass define its own **init()**, the superclass **init()** no longer works
2. subclass can override superclass functions.

- [class inheritance](#)

1. If subclass define its own **init()**, the superclass **init()** no longer works
2. subclass can override superclass functions.

🔗What is enumeration?

✓ Enum is a class in python for creating enumerations, which are a set of symbolic names (members) bound to unique, constant values.

- [class inheritance](#)

1. If subclass define its own **init()**, the superclass **init()** no longer works