Python Class Notes

python language basics

calculate triangle area

My First python program

hello world

getting start

🤔 How do I open python playground?
✔ open new terminal

```
C:\Users\12818\workspace\Rodney\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.
>>>
```

🤔 How do I read help document in python
✔ document for **print** function

```
>>> help(print)
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

print

print

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

- \n is escape sequence, which means a new line character is added
- \t is escape sequence, which means a tab character is inserted

comment

comment

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

Variable Naming

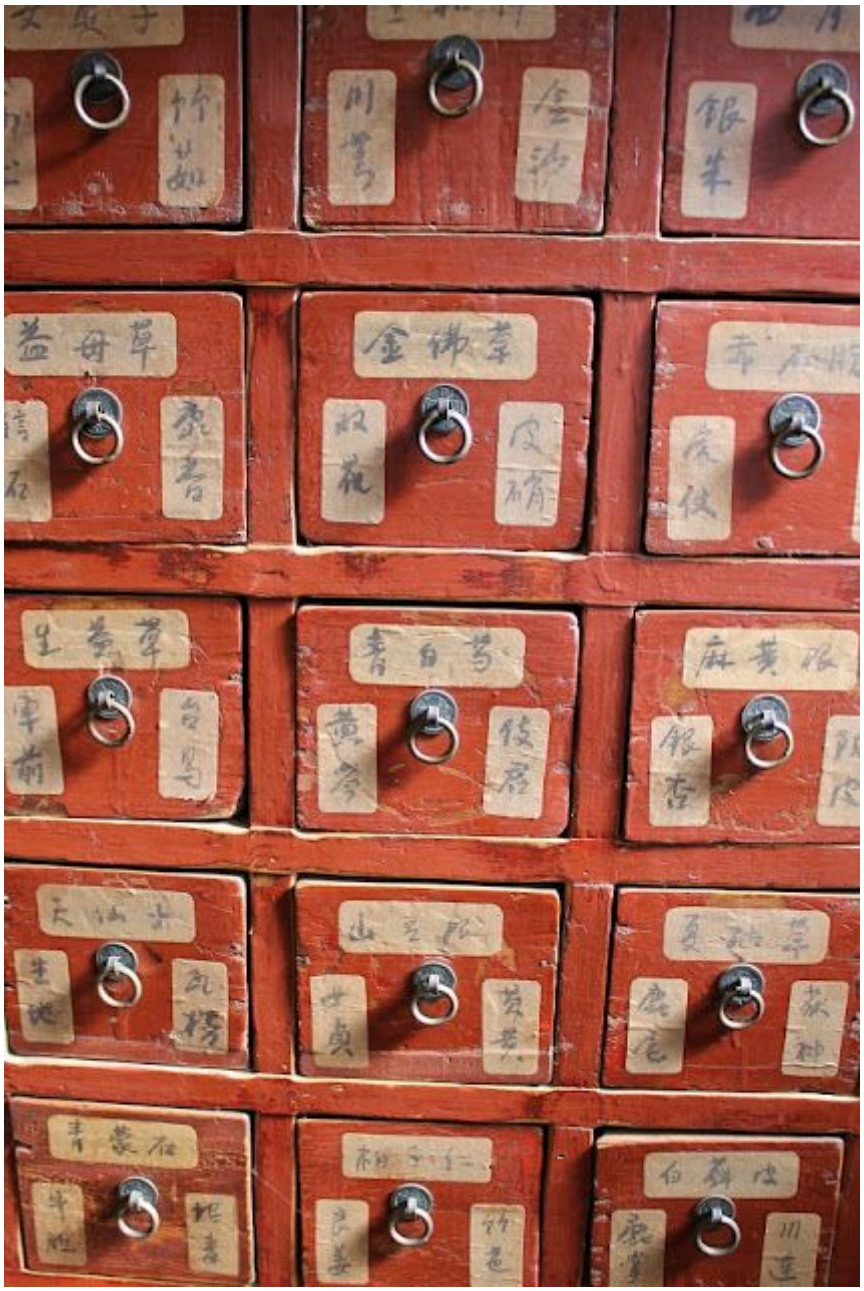
1. variable name cannot start with number
2. variable can be combination of letters and numbers _, a~z, A~Z, 0~9, ❗⚡no other special characters
3. don't use reserved keywords as variable name

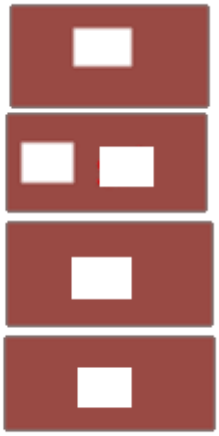
Keywords in Python programming language				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Python Keywords

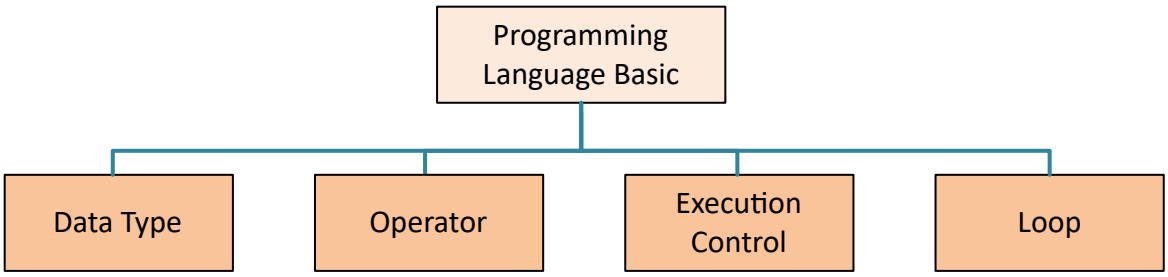
4. Avoid using existing function name as your variable name.
   otherwise, your python builtins functions no longer works the way you expected.
5. ☝class name, function name and attribute name, all of them must follow the rules above☝.

Variable and memory





Hello



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: tuple1start]:[end]:[step
  - tupler operator +, *
  - as function: tuple(iterable)
  - tuple functions ()
- List
  - list is iterable
  - list is mutable
  - list slicing: list1start]:[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

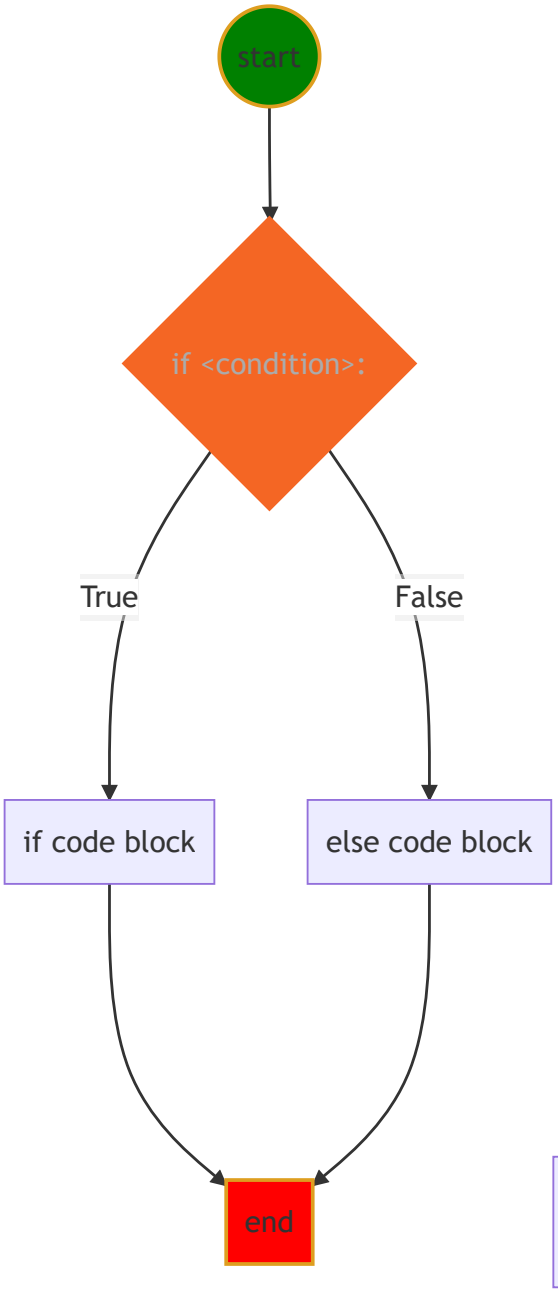
    - as function: set(iterable)
    - set functions ()
- Dictionary
    - iterable
    - mutable
    - no duplication
    - ** operator
    - function (items, keys, values, clear, pop)

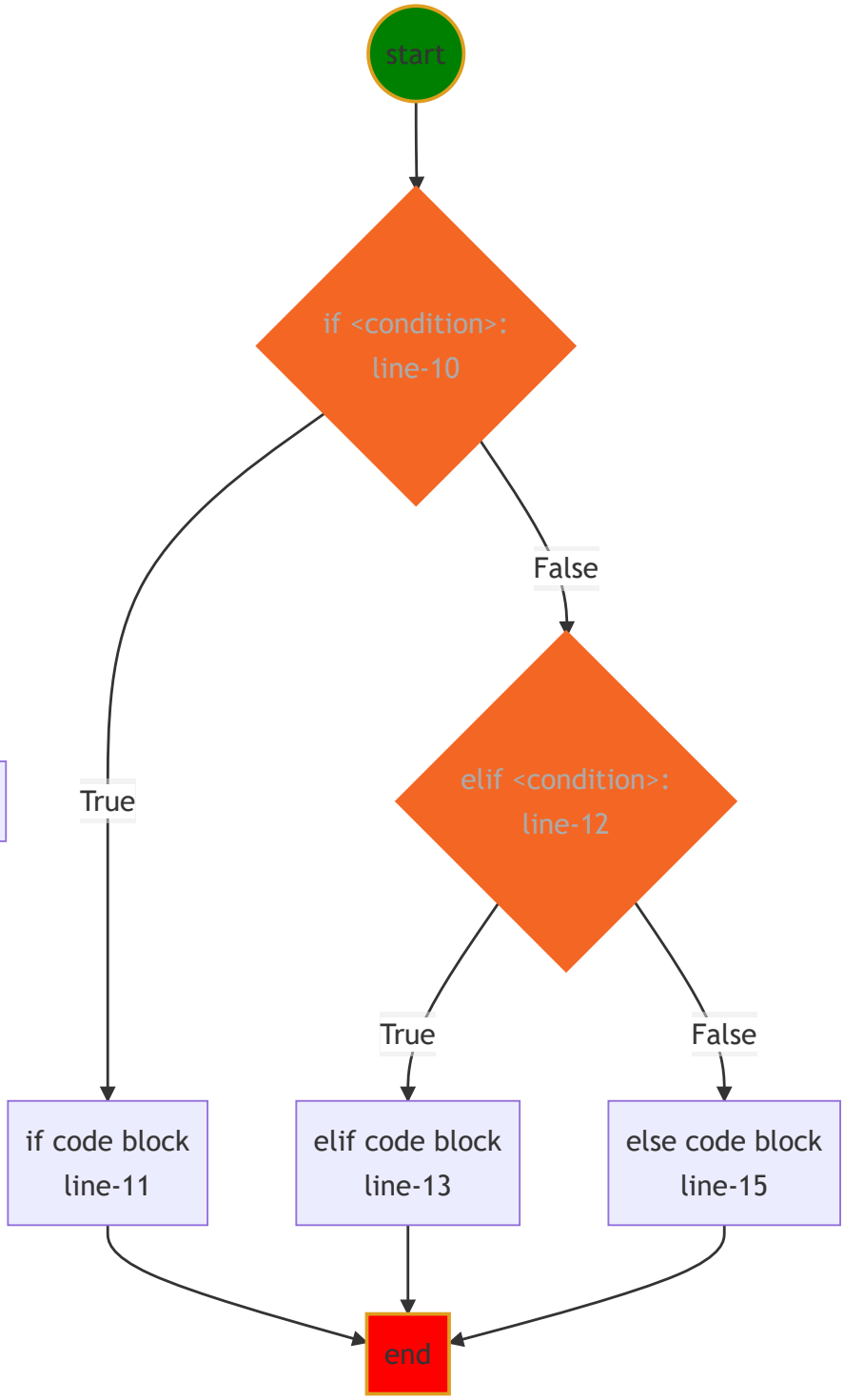
operator

- Arithmatic Operator: +; -; *; /: %; **;//(floor divisor)
  arithmatic.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
```

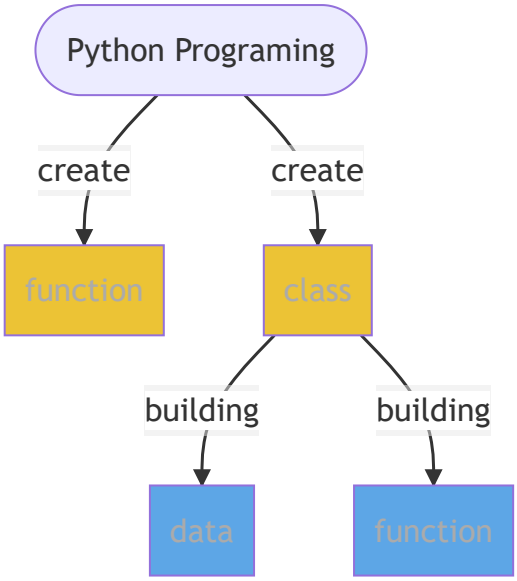


Table of Contents

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
```

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

- 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