§ week 0 Func. & Variable.

1. code hello.py 创建新文件.
2. python hello.py 运行 hello.py
   (py, py -3) python = interpreter.
3. str = input ("ask sth") ⇒ expect str.

Java I/O
```
import java.util.Scanner
Scanner input = new Scanner (System.in);
int num = input.nextInt();
System.out.println ("You entered " + num);
```

succinctly 言简意赅.

4. print ("hello," , name , name)
   ⇒ hello, name name
   串联. 直接 concatenation 用空格.
   print ("hello,", end = " ")
   print (name)
   sep = ' ', end = '\n'. could be overwritten
   ⇒ hello, name
5. print (f"hello, {name}")
   f-string : 打印变量.
   ⇒ f'This is an f-string {var1} and {var2}'
   f'{True if num%2==0 else False}'
6. str.strip() 去头尾空格.
   - lstrip() 去左
   - rstrip() 去右

   - str.capitalize() 第一个大写
   - str.title() 全大写. yang li ⇒ Yang Li

   first, last = str.split(" ") 以空格分开.
7. % remainder, modulo sign.
8. py. 运行 python interactive mode
9. num = int (input ("What's the num?"))

⇒ { Integer.valueOf( )
Integer.parseInt( )

float (...) .

round(num,2) 留两位. round (...) 约到 nearest Int.

⇒ Math.round (...)

f"{num:,}" ⇒ eg. 1,000
用逗号为隔每4位. NOT 1000.

:format

'{:.2f}'.format (5.3910) ⇒ 5.39
两位小数.

'{:.2f}%'.format ⇒ 5.39%.
百分比.

truncate (8.499) ⇒ 8.49
只留两位小数.

Padding with zeros :

左 '{:0>9}'.format (3.499) ⇒ '00003.499'

右 '{:0<11}'.format ⇒ '3.499000000'

长度 = 给的数. 包括 .

10. def ⇒ define func.

```
def funcName ( ) :
    print ("return")
def funcName ( arg = "Default") :
```

Default value.

11. def main ( ) :

⇒ public static void main (String[] args) {...}

再 main ( )

12. `n ** 2 = pow(n, 2)`

⇒ $n^2 = n * n$.

* command : `cp hello.py to.py` (copy)
`mv hello.py to.py` (移动到 …)

substring : `str[0:]` 全部
`str[0:-1]` 去尾.
`str[1:]` 去头.

---

§ Week 1 Conditionals

connote 言外之意. 意味.
parity 平等. 相同

1.
```
if x < y :
    print("x is less than y")
elif x > y :  ...
else :        ...
```

2. `if x > y or x < y :`
3. `if 80 <= score < 90 :`
4. `if n % 2 == 0` even (remainder)
   `== 1` odd
5.
```
if is_even(n) :
    print('Even')
def is_even(n) :
    if n % 2 == 0 ...
        return True / False
```
Func. call.

⇒ `return True if n % 2 == 0 else False`
`return (n % 2 == 0)` 可省略.

6.
```
match name:
    case 'Harry' : ...
```

* string immutable.
所以 built-in 都 return new string.
list mutable.
所以直接修改. return None

case 'Draco' : ...

case _ : ...

下划线：default

or : Bool Expr

⇒ Java : switch . break

```
switch (name) {
    case value : ...
                 break;
    default : ...
}
```

match pattern.

※ match msg : 模式匹配. 而不是 "or"

```
case 'Hello' | 'hello' : ...
case msg if (msg[0] == 'H') |
            (msg[0] == 'h') :
```

用 | , if 要 case ele if ...

※ if __name__ == '__main__':

    main()

⇒ 可以在 Script 运行该文件.

在 Module (引用) 中 define function. 不运行整个文件. (main())

※ Mac 上有一点奇怪. input (...) 可能直接 return value.

⇒ Windows : "1 + 2"

Mac : 3 直接计算

isinstance (obj , str) ⇒ 询问是否是这个类形的实例

---

§ Week 2 Loops

1. i = 3

```
while i != 0:
```

2. range (3) ⇒ [0, 1, 2] list

```
for i in [0, 1, 2]:
    print (...)
```

print (…)
i -= 1

3. range (int)
从 0 到 int - 1
range (start , stop , step)
⇒ range (3, 20, 2)
从 3 到 19 , increment 2
3, 5, 7 … 19

4. print ("meow\n" * 3 , end = " ")

5. while True :
    n = int (input ("What's n?")
    if n < 0 : continue
    else : break
任意符、不用 i . (指向被忽略的值).
for _ in range (n) :
    print (…)

4种 下划线 _ :
1° 交互器 ILDE 中保存最后表达式的值.
2° 指向被忽略的值.
3° 分割数字
4° 赋予属性和方法含义.

6. Python Built-in data types : Arrays
- List : [1, 2, 3] ordered, changeable, √重复. 不变成list就可以.
- Tuple : (1, 2, 3) ordered, unchangeable. √重复. change了.
- Set : {1, 2, 3} unordered, unchangeable, unindexed. X重复. 可加减.
- Dictionary : { "1" : "one" , "2" : "two" }.
  store key : value pairs , ordered . changeable. X重复.

※: type(range(3)) ⇒ class 'range'
不是 list !!

Python 的 scope : 从 define 开始到 script 结束.
而不是块级 scope (作用域).

- List : 列表 ⇒ Python 内置数据类型.
- Array : 数组 . ⇒ 数值 . 计算 Math.

Abstraction: A simplification of a potentially more complicated idea.

※ if int[num] not in [5, 10, 25]:
查 punctuation: import string.
if _ in string.punctuation.

Week 3 Exceptions

wry 歪斜的. glean 收集.
Value Error eg. int(str)

{ proactively 主动地
passively
preach 宣扬.

try:
except:

eg.
```
except ValueError:
    print("error")
```
※ 哨指.

type: SyntaxError (只能自己排除 fix)
ValueError (Runtime)
NameError (NOT Defined)

※
```
try:
    x = int(...)   # input a string → error
except:
    ...
print(x) ⇒ NameError
```
先 execute 右边. 所以 X NOT Defined.

※ 用 else:

{ try: ① except: else: ② → No error

```
eg. while True:
        try:
        except:
        else:
            break
```

pass → execute. 过.
raise raise exception.

※ Debugging
- step over : 直接得出该行结果. ↷ 跳过到下一行
- step into : 进入这个 function. 逐行看.

Dict.keys ⇒ class dict.keys
用 sort 要先 list( )

padding 0 ⇒ f'{var:02}' pad 0 before. length = 2.

Week 4 Library

modules : a library has one / more func.
eg. random : random.choice(seq) usually a list
import library : load a library.
from library import func : access to specific func
eg. from random import choice
coin = choice(seq) 可能会有 conflict name

random.choice(seq).
.randint(a,b) # [a,b].
.randrange(a, b+1) # [a, b+1) ⇒ [a,b]
.shuffle(sth.eg. list) shuffle in place, return None

statistics.median(seq)
.average(seq)

Command-line arguments

eg. python hello.py sth, sth, sth.

sys.argv[idx] 一般 sys.argv[0] 是 script.py

argument vector

※ 如果 idx 位置为空 ⇒ IndexError (out of range).

len(sys.argv)

type 'list'

for arg in sys.argv: print(arg)

sys.exit([arg])

退出 # print arg (error msg).

package: func in folder, third-part library that can be installed on PC PYPI Python package index

eg. cowsay ⇒ pip install cowsay

package manager

允许直接从命令行运行模块代码.不加.py

✡ py -m pip install package

看对 Python 版本!!✡✡

APIs

Application Programming Interface

requests.get(url, auth=(..)) # method

- .status_code
- .headers['content-type']
- .encoding
- .text
- .json() # dict

} attr.

agnostic 不可知的

json.dumps(json_data, indent = int)
more readability

※ __name__ = "__main__"
Python 自动设置 __name__
run 的 script 是 __main__，其他还是 script

regiment 受管制

PEP 8 : Python Enhanced Proposal
8 ⇒ standardize code appearance.

※ Linter : 自动检查代码错误. 编码风格.
- pylint : linter
- pycodestyle : reformat the code.
- black
  eg. black script.py 自动 format

HW :
1. match pattern，用 if sth. not in [...] 更好
2. crtl-d 对应 except EOFError
3. loop 的时候控制变量.
4. requests.get ⇒ return <Response 200>
   要 json 得用 .json()

Week 5 Unit Test = Test units = Test functions.
assert True
False ⇒ Error

AssertionError

pytest automatically do unit testing
※ 类似 JUnit. 每个 assert 最好放在不同的 function ⇒ 单独的不会同时运行. Error 会 stop execute remaining.

※ import pytest
with pytest.raises(TypeError):
    func(sth).

※ 无 return value ⇒ assert ~~None~~ = sth.
side effects ⇒ print, visual, audio ...

__init__.py: treat the folder as a package
空的也行. NOT module or file

※ pytest 可以 用一个 folder + __init__.py ⇒ package
多个 file 测试 func
※ mac 上注意: python3 -m pytest test.py

HW.
1. 可以 raise ValueError(err_msg)

Week 6 File I/O
open file: open(fileName, method)
※ return file handle
io.TextIOWrapper.

eg. file = open('name.txt', 'w')
file.write(name) write mode
file.close()

※ 'w' : write mode (overwritten)
'a' : append mode (ND overwritten)

with (Automatically close the file after usage)

```
with open( fileName, mode) as file:
```

※ 'r' : read mode

```
filehandler.readlines() ⇒ list
for line in lines:
    print (line.rstrip())   remove \n.
```

sorted (iterable, /, *, key = None, reverse = False)
eg. list, dict.keys, set

CSV Comma, separated values.

※ 也用 open as handle.
line.rstrip()

※ Python 和人一样. double quote 也用 single quote

※ sorted (list, key = get_name) → func
Sort on a specific field

(list, key = lambda student : student['name'])
另用 define func.
Input : Output

Lambda: create a func anonymously & allow you to pass it in. as value

List

CSV module: csv.reader(file) piecemeal

⇒ for row in reader: 要迭代.

Dict
. DictReader(file) 在第一行要把 attribute 写上. eg. name

⇒ name : row["name"]

. writer (file)

⇒ . writerow(content)

. DictWriter(file, fieldnames=[..])

⇒ . write row (dict)

Binary Files

PIL pillow : navigate img file (filters, etc)

eg. from PIL import Image.

image = Image.open(file).

image 1 . save (return_file, save_all = True, append_images [...], duration = 200, loop = 0)

save together

有所有的.

pause 200 ms (+s) in between

loop infty.

HW.

1. File Not Found Error
2. 多用 enumerate ⇒ for idx, item in enumerate (sth):
3. 多用 nested Expr.

[key, value for idx, value in enumerated (sth)]

4. Dict Writer. write header ()

write a row with the field names

5. PIL. ImageOps. fit (input, size) # resize. crop

. Image. paste (shirt, shirt)

overlay ing mask

Week 7 Regular Expression (regexs)

※ if username : print (True)

anything Except None or "" (empty)

- .startswith( )
- .endswith( )

param to modify func behavior

re . search (pattern , string , flags = 0)

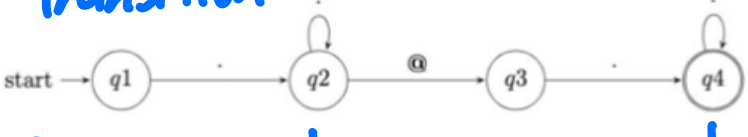
wildcard .

.	any character except a newline
*	0 or more repetitions
+	1 or more repetitions
?	0 or 1 repetition
{m}	m repetitions
{m,n}	m–n repetitions
^	matches the start of the string
$	matches the end of the string or just before the newline at the end of the string
[]	set of characters
[^]	complementing the set

\d	decimal digit
\D	not a decimal digit
\s	whitespace characters
\S	not a whitespace character
\w	word character … as well as numbers and the underscore
\W	not a word character

No dot

transition



Consume char.

accept State

raw string : r "pattern" treat literally

⇒ passed in exact as it is .

eg. \ , . , … 不会 misinterpret

caret ^ : 插入符号.

[^] : eg. [^@] Anything not @ .

{m} : eg. @{1} Only one times.

[ ] : eg. [a-z A-Z 0-9]

(...|...) : eg. (edu|gov) group & or.

flags

- re.IGNORECASE
- re.MULTILINE — input spans multiple lines.
- re.DOTALL — configue dot identifies as new line

A|B	either *A* or *B*
(...)	a group
(?:...)	non-capturing version

will be returned

分组. 但不捕获.

和 group 有关.

re.match(pattern, string, flags=0) match beginning

.fullmatch(...) match till end (start & end)

re.search(...)

⇒ sth = matches.groups() # all

matches.group(1), ...

从 group(1) 开始. 1, 2, 3 ...

0 指的是全部. = group()

walrus (海象)

if var := expr : both assign & exe. bool. expr.

string.removeprefix(prefix)

re.sub(pattern, replace, string_where, count=0, flag)

substitute.

※ 注意 group(num) stringent 严谨的.

不要的用 (?:…)

.split(pattern, string, maxsplit=0, flags=0)

.findall(pattern, string, flags=0)

HW

1. 注意 ^ : .startswith()
   $: .endswith()

2. re.search
   => matches.groups() # return tuple.
(…,…)= 可以用 len(..) 看长度.

3. 可以用 var = var or newVal
   排除 None

4. f"{int(sth):02}" ✓ # 9 => 09

   直接用 str:02 # 9 => 90

5. re. \b boundary. 单词边界

Week 8 Object-Oriented Programming

tuple : immutable collection of data

eg. return (name, house)

only return one value(tuple)

```
tuple[0] = name
tuple[1] = house
```

TypeError: tuple obj doesn't support item assign.

eg. return [name, house]

list is mutable.

... (implement later)

class : blueprint of data type

object : create object from classes.

method : func inside. behave in a special way.

Dunder : 双下划线 __

eg. __init__

__str__ see obj. as string

raise : raise Errors

property : attr. w/ more defense mechanism

※ decorators : func modify other func.

```
# Getter
@property
def attr_name(self):
# Setter
@attr_name.setter
def attr_name(self, attr):
```

PRIVATE

return self._attr_name.

不能一样 ⇒ collide.

加 underscore

self._attr_name = attr

☆ __init__ 会自动 call setter (如果有的话)
__init__ 不用 self._attr_name
↘ call setter.
D/W circumvent setter

☆ self._var # Don't touch this
setter 直接修改
self.__var

@classmethod NOT instance method
类相关. def func_name(cls, param)
@staticmethod ↓cls.variable 等.
不与 self, class 相关.
def func_name(param)
封装与类相关但不需要访问实例.

※ singleton 全局可访问. 一局一实例. 首次就创建.
self ⇒ specific object

Inheritance hierarchical design. inherit all func
class subclass(superclass):
☆☆ super().__init__(name)
access superclass

```
BaseException
 +-- KeyboardInterrupt
 +-- Exception
      +-- ArithmeticError
      |    +-- ZeroDivisionError
      +-- AssertionError
      +-- AttributeError
      +-- EOFError
      +-- ImportError
      |    +-- ModuleNotFoundError
      +-- LookupError
      |    +-- KeyError
      +-- NameError
      +-- SyntaxError
      |    +-- IndentationError
      +-- ValueError
```

operator overloading ✡✡ denomination 教派

eg. '+' : plus, concatenate, ...

object.__add__(self, other)

Week 9 Et Cetera

global outside all func.

※ UnboundLocalError: Can't write to global var. ONLY READ.

↳ 在 local func 里用 global variable (modify)

global global_var # CAN EDIT.

✡
- self.variable 公有实例变量.
- self._variable 受保护. 类似 private. 只在内部.
- self.__variable 私有实例变量 private.

sparingly 俭省地

用 getter return self._variable

可以防止 accidentally change variable.

例如 account.balance = 1000.

constants ALL CAPS convention

type hints eg. mypy file.py ⇒ help find error

```
name: str = "String".
def add(a: int, b: int) -> int:
```

docstrings

eg.

```
func document
"""
comment (func purpose)
```

```
:param n: what is n?
:type n: n's type.
:raise Error: happen how?
:return: return what
:rtype: return type
"""
```

argparse [no need for sys.argv].

```
parser = argparse.ArgumentParser()
parser.add_argument("-n", help="explain")
args = parser.parse_args()
print(args.n)
```

※ { python file.py -h / --help ⇒ argument manuel

先确认 args.sth is not None.

unpacking eg. * list = item, item, ...
work for tuple.
set ⇒ wield !!!

** dict unpack key & value

※ * args, ** kwargs variadic 可变参数

↓ positional tuple

↓ keyword (named) dict

eg. func(name = sth, name = sth)

print (*objects)

map map(func, iterable, ...)

不需要() [NOT call it now]

list comprehensions [x if expr else y].

[x for i in iterable if expr]

filter (类似 map) = list comprehension

filter(func, iterable)

write it yourself, Bool. 不加()

可以写 lambda func input:output

dictionary comprehension Like list comprehension.

list = [{key:value} for i in iterators].

dict = {key:value for i in iterators}

可以是 variable

enumerate [idx, value]

for idx, var in enumerate(iterators):

generators + **yield** ☆ return Iterator

生成器函数: 处理大型数据集/逐个处理.

NOT Run it All ⇒ One by One.

(游戏存档)

每次都 execute 一行. 而不是全 output 出来

※ suspend the func. remember the state.
⇒ execute next iteration

Final Project:

1. pytest 에서 input()

⇒ from io import StringIO

```
input_stream = StringIO(user_input_str)
monkeypatch.setattr('sys.stdin', input_stream)
result = func().
assert result == str.
```

pytest.MonkeyPatch