

1. print
 ① "~~" + "~~" 字符串连接
 ② "~~" - "~~" 跨行
 ③ "Let's go" 转义符
 ④ 'He said "~~"
 ⑤ 变量

2. 变量
 ① my_love = "~~" 赋字符串
 地址?
 135 值 整数/浮点

② 取名 ① 数字、英文字母、下划线
 ② 数字不可开头
 ③ 区分大小写
 ④ 关键字要避开

举例:
 $c = 2^3$
 $py 8 = 2^{**3}$

3. 序列数据

import math
 math.函数名(...)

4. 注释
 单行 # ctrl + /

多行 """ ~~~ """

5. 数据类型
 str 单双引号, "Hello"[5] 从0开始
 int
 float
 bool True, False
 NoneType None
 type(变量) 返回值 <class '类型'>
 len()

6. 简洁模式 & 命令行

>>> #等效命令
 退出 quit() 或 ^D
 cmd 键入 Python3
 命令行点击

7. input()
 一律返回字符串

玻璃→输入(变量)

① 不会保存任何东西

并输出状态

8. 条件
 ① if is-happy
 5 == 5 → 是 == 同C
 [语句] 宽裕缩进(4个)
 else:
 if [条件]:
 [执行]
 else:
 [执行]

② 多条件同时判断

if [-]:
 [A]
 elif [=]:
 [B]
 elif [=]:
 [C]
 else:
 [D]

9. 逻辑运算

and 与 or 或 not 非

10. 列表 → 称为方法
 特性: 1. 有列表.append(~) 或 .remove(~) 用法增加
 2. 可变的 → 用方法时会直接改变原列表
 3. 被很多类型的数据

函数: max(), min(), sorted() ↔ reverse

打印对称值

11. 字典 = {}
 "键": "值"
 另: 元组 = () 不可变 → 可作为键使用

名“键”在字典会返回一个值

12. for

for 变量 in 可迭代对象:

range(5, 10, 2)
 始终步长

13. while

while 条件A:

行动B

14. 函数定义
 def 名(): → 参数又可写数据结构
 缩进 #内容
 返回值 return A

15. 测试

try:

| 代码块

except ValueError:

| 异常后代码

except ZeroDivisionError:

except: C所有

else: 没错时会运行

finally: 无论如何都会运行

16. 测试

assert bool(应该为true)

| True 无事
 False AssertionError → 直接中止

unittest 库(Python)

另: 测试代码使用独立文件 from 文件名 import *

unittest 会优先用 test_ 开头的

要测的文件

1. 创建类, 定义对象的属性和方法

class NameOfClass: #驼峰命名法

① 构造函数 并定义属性
 | def __init__(self): 并前后两个下划线
 | self.name = "Lambten"

2) 创建对象: cat1 = NameOfClass()

3) 传递 class CuteCat:
 def __init__(self, cat_name)
 self.name = cat_name

cat1 = CuteCat("Jojo")

1. 绝对路径

windows: C:\home
 ↓ C:\home
 ↓ C:\home\data

2) 相对路径 (win 中 \ 替代 /)

① ...
 ... temp ← temp
 ... temping ← temping
 ... temping → temping
 ... temping → temp
 ... temping → home
 ... temping → data
 ... temping → ./data
 a.py → ./data/a.py

4) 定义方法
 class CuteCat:
 def __init__(self)

使用方法: if 子类有:
 调用子类
 else 调用父类

另:
 class Mammal:
 def __init__(self):
 ...

5) 类的继承 有嵌套类
 class Cat(Mammal):
 def __init__(self):
 super().__init__(name, sex)

并调用父类构造函数(不用重写)

~~~~~

2) 操作文件 读取 编码  
 open("路径", 模式, encoding = "utf-8")  
 ↳ 不写默认 "r" 只读

↓  
 返回一个文件对象

方便操作 ← txt

如: f = open("...")  
 print(f.read()) → 大文件会爆  
 除非 f.read(10)  
 ↳ 会记录阅读位置  
 不够连读

另:  
 f.readline() + while

或 f.readlines() + for

最后一定 f.close()

with open("路径") as f:  
 print(f) ← 自动关闭

"rt" (同时读写, r+ a)

"a" (不覆盖, 同下)

2) 写模式 "w" (不存在会自动创建, 会覆盖)

方法 f.write("") 不换行

① print 中 end 和 sep 参数, end: 决定以什么内容结尾, sep: 决定多对象之间

② int 整型转换 int(16, 16) → 16进制转其它, 默认十进制

③ 小写字母换 .upper 全部 .title 第一个字母大写  
 .lower

④ 对字符串:  
 .strip() 删两边空格  
 .lstrip() 删左空格  
 .rstrip() 删右空格  
 .replace(" ", "") 删所有空格  
 .split() 分割, "".join() 聚合并

限制字符串: print(name[0:10])  
 [start: end: step]

⑤ 循环  
 for i in 列表:  
 print("~~".format(i))

⑥ 生成列表  
 print([str1.split()]) #空格拆分字符串  
 str2 = str1.split(" ") 并根据结果为字符串拆分列表

#大写转换  
 text.upper() #文本  
 text.lower() #文本  
 text.title() #字符串大写  
 text.capitalize() #首字母大写  
 text.swapcase() #将大写转小写, 小写转大写  
 text.lstrip() #删除左侧空格  
 text.rstrip() #删除右侧空格  
 text.replace(" ", "") #将字符串替换成另一个字符串  
 text.partition(" ") #将字符串按指定分隔符分成三个部分  
 text.splitlines() #将字符串按行进行分割  
 text.split(" ") #将字符串按空格进行分割  
 text.startswith("abc") #判断是否以abc开头  
 text.endswith("abc") #判断是否以abc结尾  
 text.replace("abc", "def") #将字符串中的abc替换成def

⑦ list 排序式

前置代码: for i in list:  
 i = int(i)  
 list2.append(i)

↳ list1 = [int(i) for i in input().split()]

list 扩展  
 list 可变  
 操作: 内置函数、method、非导式

{ 改变 list → method list.sort  
 不变 → 内置函数 sorted(list)}

常见内置 { len(): 个数  
 max(): min(): sum(): sorted():  
 list():  
 any(): 全 True all(): 都是 True  
 ↳ 返回 True  
 enumerate(): 配合 for loop 和元素位置

方法 { .append(): 末尾加  
 .insert(): 特定位置加  
 .pop(): 删特定  
 .remove(): 删具体一个  
 .extend(): 合并  
 .copy(): 复制, 值修改, 列表  
 .clear(): 清空  
 .index(): 报位置  
 .count(): 出现次数  
 .reverse(): 反转排序 sort(): 对应  
 .sort(reverse=True): 从大到小

ydel(元素)  
 非导式: list=[语句 for i in list ]  
 可省略 append 和 建立 list 步骤