





1121212121



□□□

□□□

1. Python□□□□□□□

□□
□□□ □ / □□ □□□□
□□□□□□ □□□
□□□□□□□□□□□□

Python □□□□□□ □□□□ □ □ / □□□□□ □□□□□□□□□□□□

□□□□□□□□□□□□□□□□

- **int**□□□□ 1□ 100□ -5
- **float**□□□□□ 3.14□ 2.0
- **bool**□□□□□□ True □ False
- **str**□□□□□□□□ / □□□□□□ "test"
- **NoneType**□□□□□□□□ None

□□ / □□□□□□□□□□□□□□□□□□□□

- **list**□□□□ □□□□□□□□□□□□□□□□ □□ □□
- **tuple**□□□□ □□□□□□□□□□□□□□□□ □□ □□
- **dict**□□□□ □□□□□□□□ key □□□□□□□□ □□ □□
- **set**□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ / □□□□□□□□□□□□

- bytes / bytearray□□□□□
- complex□□□
- □□□□□□□□□□ datetime □ array □ enum □

1. list

- 



- 



- 



2. dict

- 



- 



- 



3. set

- 



- 





int/str/tuple



list/dict/set

3. Python



Python 





8 

- `int()` → 
- `float()` → 
- `str()` → 
- `bool()` → 
- `list()` → 
- `tuple()` → 
- `set()` → 
- `dict()` → 

1. 





□□□□□□□□□□

□□□ □□□□□□□□□□□□□□□□

URL □□

2. □□□□□□□□□□

□□□□□□□□□□

□□□ □□□□□ set□□□□□□□ tuple□

3. □□□□ bool ()□□□□□□

□□□□□□□□□□

□□ □□□□ 0□ None □□□□□□□

4. □□□ dict ()

□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□□□□□

1. □□□□□□□□□□ □□□□□□□ int("abc") □□□
 2. □□□□□□□□□□
 3. **tuple** □ **list** □□□□□
 4. **list** □ **set** □□□□□□□
-

POST

json

config.json

json

token

- `json.loads()` →
- `json.dumps()` →
- `json.load()` json →
- `json.dump()` → json

json 4

- 1.
- 2.
3. json
4. json

json 3

1. `eval`
2. `ast.literal_eval`
3. `re`

eval

eval Python
eval
eval

- eval
- eval
- eval list/tuple/dict

- eval
- eval
- eval

ast.literal_eval

ast.literal_eval

eval Python
eval
eval

- eval
- eval
- eval
- eval dict list tuple int str

- eval

Excel
eval

re.match/re.findall

正则表达式

应用

- 正则表达式 JSON
- Python 正则表达式
- 正则表达式

正则表达式 key value 正则表达式

正则表达式

正则表达式

- 正则表达式 正则表达式

正则表达式

- 正则表达式
- 正则表达式

json.loads vs ast.literal_eval

1. **json.loads**
 1. 正则表达式 正则表达式
 2. 正则表达式
 3. 正则表达式 JSON
2. **ast.literal_eval**
 1. 正则表达式
 2. 正则表达式
 3. 正则表达式
 4. 正则表达式

正则表达式 正则表达式 ast.literal_eval 正则表达式 json.loads

encode /decode 正则表达式 ↔ 正则表达式

正则表达式

正则表达式

- 正则表达式

- `token`
- `token`

`split / join` ↔ `split`

`split`

`split`

- `split`
- `split` URL
- `split`

`yaml` `yaml`

`yaml` `yaml` `.yaml / .yml` → `yaml` / `yaml`

`yaml` `yaml`

`yaml`

`pickle` `Pickle` `Python` `Python`

`Python` `Python` `Python`

`Python` `Python`

`Python` `Python`

`Python`

`csv` `openpyxl` `openpyxl`

`Excel` `CSV` → `Excel` / `CSV`

`Excel` `CSV`

`Excel`

base64 /hashlib



base64 encode/decode
hashlib



- int() str()
- 1. json
 - 2. literal_eval
 - 3. encode/decode
 - 4. split/join
 - 5. yaml
 - 6. openpyxl
 - 7. pickle/base64

5.

Python / 3

(Key)

(Value)

□□□□□□□□□□□□□□

□□ □□□ (□□)

□□□□□□□□□□□□□□

□□□□□□□□□□

□□□□□ **list (□□)**□
□□□□□□□□□□□□□□
□□□□ □□□□□□□□ □ □□□□□□□□□ □ **(Key)**□

□□□□□□□□

1. □□□□□□□□

□□□□□□□□□□□□□□

1. □□□□□□

□□□□□□□□□□□□□□

□□□□

□□□□ / □□□□ □□□□□ □

1. **keys()** —— □□□ □

2. **values()** —— □□□ □

3. **items()** —— □ □□□ □□□□ list(□□) □□□□□□□□□□□□□□

6.



int/float



-  `"123a"`  `"null"`  `""` 
-  `int("123a")`  `ValueError`



1. 
2.  **try-except** 
3. 



json.loads



- Excel /  `{'name': 'test'}` 

- `json.loads`

1. `ast.literal_eval`
2. `json.loads`

`None` `/`

- `int(None)`
- `bool("False")` `True`
- `null`

1. `None` /
2.

- `int([1])` `int({ "a": 1 })`
-



1.  key  
2. 



str/bytes 



- encode / decode ☐ gbk ☐ utf-8 ☐
-



1.  **utf-8**
2.  `UnicodeDecodeError` 



1. `try-except`
2. `ast.literal_eval` `eval` `json` `json`
3. `str to int` `str to dict`



7. 如何读取 excel

Python 如何读取 Excel 文件 `xlrd/xlwt` `openpyxl` `pandas`



1. **xlrd / xlwt**
 1. 如何读取 **.xls** 文件 Excel 97-2003
 2. `xlrd` 如何安装
 3. 如何读取
2. **openpyxl**
 1. 如何读取 **.xlsx** 文件 Excel 2007+
 2. 如何安装
 3. 如何读取 工作簿 工作表
3. **pandas**
 1. 如何安装 `openpyxl/xlrd`
 2. 如何安装 `pandas`
 3. 如何安装 `pandas`
 4. 如何安装



1. pandas 如何安装

如何安装

2. openpyxl 如何安装

如何安装



1. **sheet**

1. pandas `sheet_name=None`

2. openpyxl `wb.sheetnames`

2. pandas



1.



1. openpyxl `Font` `PatternFill`

2.

1. pandas

2. `chunksize`



“ pandas `xlrd/xlwt` openpyxl
 .xls `xlrd/xlwt`

8.



True False

1.

and True True False True and True → True True and False → False

or True True False True or False → True False or False → False

not not True → False not False → True

2. True 1 False 0

True == 1 → True False > 0 → False

3. True > 0 → True False > 0 → False 1 > 0 → True

True + True → 2 False * 5 → 0

9. list tuple

set

dict

Python list tuple set dict

list

- list mutable
 - tuple immutable
 - set mutable
 - dict mutable
- 3.7 + 1.0 = 4.7

list

list []

- list [1, 2, 3]
- list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

list

tuple

-  
- 
-  
- 
- 

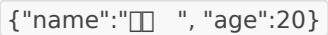


set

-  
- 
- 
- 
- 



dict

-  
- 
- 
- 3.7 
- 









- □ □□□ → list
- □ □□□ → tuple
- □ □ / □□□ → set
- □ □□□□ → dict

10.



list □□□□□□□□

“□□□□”

□□ □□□□□□□□ □□□□ □□□ □□□□□□□□□□ □□□□□□

- □□□□□ test_cases = ["□□ ", "□□ ", "□□ "]
- □□□□□□□□ params = [{"id":1}, {"id":2}]
- □□□□□□□□ user_list = [] □□□□ append
- □□□□□□□□

□□□□ list□

“□□ □□□ □□ □□□□ □

tuple □□□□□□□□

“□□□□”

□□ □□□ □□□□ □□□□ □□□□ □□□ □□□□□□□□ □□□□□□
□□ / □□□□□

- □□□□ HOST = ("192.168.1.1", 8080)
- □□□□□ STATUS = ("□□ ", "□□ ", "□□ ")
- □□□□□□ return code, msg □□□□□

- | | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|

[illegible]

11

--	--	--	--	--	--	--	--

- | | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|

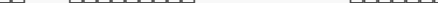
“ set ”

11

--	--	--	--	--	--	--	--	--

-

- | | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|





- **list**  → 
- **tuple**  → 
- **set**  → 
- **dict**  → 

11. 

 `lst = [1, 2, 3, 4]`



append ()


insert ()


extend ()


□□ +

□□□□□□□□□□

□□□□□□□□

del □□

□ □ □
□□□□□□□□□□

pop (□□)

□□□□ □□□□
□□□□□□□□□□
□□□□□□□□
□□□□□□□□□□

remove (□□)

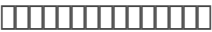
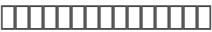
□ □ □□□□□□□□
□□□□□□□□□□

clear()

□□□□
□□□□□□□□□□

□□□□□□□□

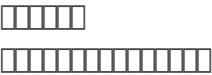
□□□□ □□□□
□□□□□□□□□□



index ()

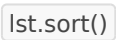


count ()



in 



-  
 -  
 -  
-



- □□ append /insert/extend
- □□ del /pop/remove /clear
- □□□□□□□□
- □□□□□□□□ index□ count□ in

12.



```
dict[ ]
```

--	--	--	--	--

key



1.



2. update() ☐ ☐ ☐ ☐ / ☐ ☐



1.

--	--	--	--

key



2. pop(key) □□□□□

□□□□□□□□□□

3. popitem() □□□□□□

□□□□□□□□□□

Python3.7+□

4. clear() □□□□

□□□□□□□□□□

□□□□□□□□□□

□□ **key** □□□ □□□□□□

□□□□□□□□□□

□□□□ update() □

□□□□□□□□□□

□□□□□□□□□□

1. □□□□ key □□□□□

□□□□□□□□□□

△ □□□ key □□□□ □□ □

2. get() □□□□□□□□

□□□□□□□□□□

3. □□□□ key

□□□□□□□□□□

4. □□□□ value

□□□□□□□□□□

5. □□□□□□□□

□□□□□□□□□□

6. □□ key □□□□

□□□□□□□□□□

□□□□□□

- □ □ dict[key] = □ / update()
- □ □ del / pop() / clear()
- □ □ dict[key] = □□
- □ □ dict[key] / get() / keys/values/items

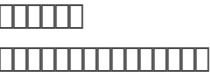
□□□□□□□□□□ **key**□□

13. □□□□□□□□

□□ set□□□□□ □□□□□□□□□□
□□
□□□□□□□□□□

□□□□

1. add()



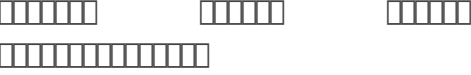
2. update()



1. remove()



2. discard()



3. pop()



4. clear()





△ □□□ □□□ “□□ ” □□
□□

- □□□□□□
- □□□□□□□□□□

□□□□□□ □□□□□□□□ □



□□ □□□□□□ /□□□ □□□

1. □□□□□□□

□□□□□□□□□□