



-  Three small squares followed by a horizontal line.



1. Python数据类型

数据类型
整数 / 浮点数
字符串
布尔值
None

Python 数据类型 整数 / 浮点数 字符串 布尔值 None



- **int** 整数 1 100 -5
- **float** 浮点数 3.14 2.0
- **bool** 布尔值 True False
- **str** 字符串 / 文本 "test"
- **NoneType** None



- **list** 列表 []
- **tuple** 元组 ()
- **dict** 字典 {key: value}
- **set** 集合 {}



- 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

ast.literal_eval Python
ast.literal_eval
ast.literal_eval

- ast.literal_eval
- ast.literal_eval
- ast.literal_eval
- ast.literal_eval dict list tuple int str

- ast.literal_eval

ast.literal_eval Excel
ast.literal_eval

re.match/re.findall

11111111

1111

- 11111111 JSON
- 11 Python 11
- 11111111

1111 key value 1111

11111111111111

111

- 111 11111111

111

- 11
- 11111111111111

json.loads vs ast.literal_eval

1. **json.loads**
 1. 1111 111
 2. 11
 3. 11111111 JSON
2. **ast.literal_eval**
 1. 1111111111
 2. 11
 3. 111111
 4. 11111111

11111111 111 ast.literal_eval1111 json.loads1

encode /decode1111 ↔ 1111

11111111111111

11

- 11111111

- `token`
- `token`

`split / join` ↔ `token`

`token`

`token`

- `token`
- `token` URL
- `token`

`yaml` `token`

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

`token` `token`

`token`

`pickle` `token` `Python` `token`

`token` `token` `Python` `token` `token`

`token` `token` `Python` `token`

`token` `token`

`token`

`csv / openpyxl` `token`

`token` `token` `Excel` `CSV` `token` → `token` / `token`

`token` `token`

`token`

base64 /hashlib



base64 encode/decode example



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

base64 encode/decode example



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`
4. `try-except`
5. `try-except`



7. 读取 excel

Python 读取 Excel 的库有 **xlrd/xlwt** **openpyxl** **pandas**

1. 读取 Excel 文件

1. **xlrd / xlwt**
 1. 读取 **.xls** Excel 97-2003
 2. xlrd 读取
 3. 写入
2. **openpyxl**
 1. 读取 **.xlsx** Excel 2007+
 2. 读取
 3. 写入
3. **pandas**
 1. 使用 openpyxl/xlrd 读取
 2. 读取
 3. 写入
 4. 写入

2. 写入 Excel 文件

1. pandas 写入

写入

2. openpyxl 写入

写入



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 & False → False 1 & 0 → 0

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

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]

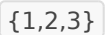
list

tuple

-  
- 
-  
- 
- 

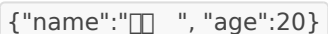


set

-  
-  
- 
- 
- 



dict

-  
- 
-  
- 3.7 
- 







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

“ / ”

11

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

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

“    set  ”

11

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

-

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

“ ”



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

11. 

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



append ()


insert ()


extend ()


□□ +

□□□□□□□□□□

□□□□□□□□

del □□

□ □ □
□□□□□□□□□□

pop (□□)

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

remove (□□)

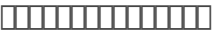
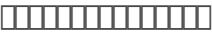
□ □ □□□□□□□□
□□□□□□□□□□

clear()

□□□□
□□□□□□□□□□

□□□□□□□□

□□□□ □□□□
□□□□□□□□□□



index ()



count ()



in 



-  
 -  
 -  
-



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

12.



```
dict[ ]
[ ]
[ ]
```

key

[illegible]

2. update() ☐ ☐ ☐ ☐ / ☐ ☐



1. del

--	--	--	--

 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

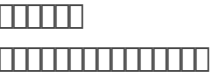
□□□□□□□□□□ **key**□□

13. □□□□□□□□

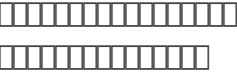
□□ set□□□□□ □□□□□□□□□□
□□
□□□□□□□□□□

□□□

1. add()



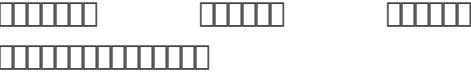
2. update()



1. remove()



2. discard()



3. pop()



4. clear()





△ □□□ “□□” □□



- □□□□□□
- □□□□□□□□□□

□□□□□□ □□□□□□□□ □



□□ □□□□□□ /□□□ □□□□

1. □□□□□□□□

