



# 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 值
- **set** 集合 {}



- bytes / bytearray
- complex
- datetime array enum



- 

[illegible]

- 

- 

- 

list

/dict/set

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. 数据序列化

2. 数据反序列化

3. json 数据格式

4. POST 请求

5. json 数据格式 → 数据

6. config.json 文件

7. 数据序列化

8. json 数据格式

9. token 数据

10. 数据序列化

11. 数据反序列化

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

12. json 数据格式 4 数据

1. 数据序列化
2. 数据反序列化
3. json 数据格式
4. 数据 json 数据



□□ json □□□□□□ 3 □□ □□□□□□□□□□

1. **eval**
2. **ast.literal\_eval**
3.  **re**

[illegible]

Python 

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

- 
- 
- 

III

- 
- 
- 

# ast.literal\_eval

☐ eval ☐☐☐☐

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

- `[]` `list`
- `{}` `dict`
- `()` `tuple`
- `int` `int`
- `str` `str`

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

- 

The image shows two tens rods and fifteen ones units. The tens rods are placed side-by-side, and the ones units are placed in two rows of seven and one row of eight, totaling 15 units.

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

-  JSON
-  Python 
- 

- 
- 

4. 

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

|                                                                                                     |                                              |                         |                                                         |                   |
|-----------------------------------------------------------------------------------------------------|----------------------------------------------|-------------------------|---------------------------------------------------------|-------------------|
| <div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div> | <div><div></div><div></div><div></div></div> | <b>ast.literal_eval</b> | <div><div></div><div></div><div></div><div></div></div> | <b>json.loads</b> |
|-----------------------------------------------------------------------------------------------------|----------------------------------------------|-------------------------|---------------------------------------------------------|-------------------|

encode / decode  ↔ 





**(Value)**

$\begin{bmatrix} \square & \square \end{bmatrix}$        $\begin{bmatrix} \square & \square & \square \end{bmatrix}$        $\left( \begin{bmatrix} \square & \square \end{bmatrix} \right)$

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

`list ( )`

☐ ☐ ☐ ☐      ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐      ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐      ☐ (Key) ☐

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

1. 

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

1. 

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

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

$$\begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline \end{array} \quad / \quad \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline \end{array} \quad \begin{array}{|c|c|c|c|c|c|} \hline & & & & & \\ \hline \end{array} \quad \begin{array}{|c|} \hline \\ \hline \end{array}$$

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 })` `int()`
- `int()`

`int()`

1. `int()` `key` `int()` `int()`
2. `int()`

`int()`

`int()`

str/bytes `int()`

`int()`

- `encode / decode` `gbk` `utf-8` `int()`
- `int()`

`int()`

1. `int()` **utf-8**
2. `int()` `UnicodeDecodeError` `int()`

`int()`

`int()`

1. `int()` **try-except** `int()`
2. `int()` `int()`
3. `int()` **ast.literal\_eval** `int()` **eval** `int()` **json** `int()` **json** `int()`
4. `int()` `int()` `int()`
5. `int()` `str_to_int` `str_to_dict` `int()`

`int()`



7.  excel

Python  Excel  **xlrd/xlwt** **openpyxl** **pandas**



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



 1  pandas 



 2  openpyxl 

xxxxxxxxxxxx

□□□□

1. `sheet`
  1. pandas `sheet_name=None`
  2. openpyxl `wb.sheetnames`
2. pandas

xxxxxxxxxxxx

- 1.

xxxxxxxxxxxx

1. openpyxl `Font` `PatternFill`
2. pandas `chunksize`

□□□□□

“ pandas openpyxl  
 .xls xlrd/xlwt

8. □□□□□□□□□□□□

True False

- 1.

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

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

`not` `not True` → `False` `not False` → `True`

2. `True` `1` `False` `0`

`True == 1` → `True` `False > 0` → `False`

3. `1` `0`

`True + True` → `2` `False * 5` → `0`

9. `list` `tuple`

`set`

`dict`

Python `list` `tuple` `set` `dict`

`list`

- `list` `list`
- `tuple` `tuple` `tuple`
- `set` `set` `set`
- `dict` `dict` `3.7 + dict`

`list`

`list` `[]`

- `list` `[1, 2, 3]`
- `list` `list`
- `list` `list`
- `list`
- `list`

`list`

---

## tuple `tuple` `()`

- `tuple` `(1, 2, 3)`
- `tuple`
- `tuple` `list`
- `tuple`
- `tuple`

`tuple`

---

## set `set` `{}`

- `set` `{1,2,3}`
- `set` `list`
- `set`
- `set`
- `set`

`set`

---

## dict `dict` `{key:value}`

- `dict` `{"name": " ", "age": 20}`
- `dict`
- `dict` `list`
- `3.7` `list`
- `dict`

`dict`



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

10.



list □□□□□□□□

“□□□□”

□□ □□□□□□□□ □□□□ □□□ □□□□□□□□□□□□ □

- □□□□□□ test\_cases = ["□□ ", "□□ ", "□□ "]
- □□□□□□□□ params = [{"id":1}, {"id":2}]
- □□□□□□□□ user\_list = [] □□□□ append
- □□□□□□□□

□□□□ list□

“□□ □□□ □□ □□□□ □

# tuple 不可变序列 “只读”

tuple 序列 有序 不可变 元素 可任意类型 不可变序列 不可变

tuple / 元组

- 创建 HOST = ("192.168.1.1", 8080)
- 访问 STATUS = ("OK", "200", "OK")
- 遍历 return code, msg 遍历
- 遍历 key 列表 tuple 遍历

创建 tuple

```
“ ” / 不可变序列
```

## set 集合 + 集合

set 集合 无序 不可变 元素 可任意类型 不可变集合 不可变

集合

- 创建 user\_ids = list(set(old\_ids))
- 遍历 list 遍历 if name in black\_set:
- 遍历
  - A 与 B set(A) - set(B)
  - 交集 set(A) & set(B)
- 遍历 token id 遍历

创建 set

```
“ ” 不可变集合 集合 set
```

## dict 字典

dict 键值对 key 键 值 不可变字典 不可变

字典

- 创建

字典

- `dict`
- `case = {"id":1, "title":" ", "expect":200}`
- `header` `cookies`

`dict`

`dict`

`list`

- `list` `list` → `list`
- `tuple` `tuple` → `tuple`
- `set` `set` → `set`
- `dict` `dict` → `dict`

11.

`list`

`list`

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

`list`

`append ( )`

`list` `list`

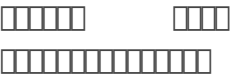
`list`

`insert ( )`

`list` `list`

`list`

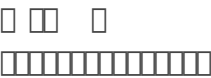
extend ( [ ] )



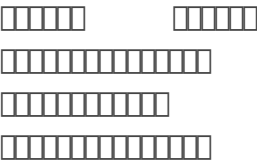
[ ] +



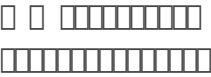
del [ ]



pop ( [ ] )

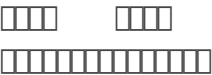


remove ( [ ] )

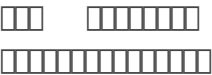


clear()

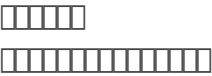




index (  )



count (  )



in 



- `lst.sort()`
- `lst.reverse()`
- `len(lst)`



- `lst.append()` / `lst.insert()` / `lst.extend()`
- `lst.del()` / `lst.pop()` / `lst.remove()` / `lst.clear()`
- `lst.index()`
- `lst.count()`      `lst.index()`   `lst.count()`   in

12. 

`dict` `{}`  
`dict`  
`dict`

`key` `value`



1. 

`dict`

2. `update()` `dict` / `dict`

`dict`



1. del [ ] key



2. pop(key) [ ]

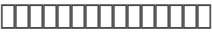


3. popitem() [ ]



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.

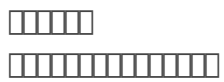


set    
  


---



1. add(  )



2. update(  )



1. remove(  )



2. discard(  )



□□□□□□□□□□

3. pop()

□□□□□□□□□□  
□□□□□□□□□□

4. clear()

□□□□  
□□□□□□□□□□

---

□□□□

- △ □□□□      “□□” □□  
□□
- □□□□□□
  - □□□□□□□□□□

□□□□□□      □□□□□□□□      □

---

□□□□

□□   □□□□□□      /□□□   □□□□

1. □□□□□□

□□□□□□□□□□

---