

# PYTHON

## 编程基础

# list数据类型

# 概述

List（列表）是Python中一种非常重要的数据类型。

列表中可以包含多个元素，且元素类型可以不相同。

每一元素可以是任一数据类型，包括列表（即列表嵌套）及后面要介绍的元组、集合、字典。

所有元素都写在一对方括号“[]”中，每两个元素之间用逗号分隔。

对于不包含任何元素的列表，即[]，称为空列表。

# 列表元素索引

列表中元素的索引方式与字符串中元素的索引方式完全相同，也支持从前向后索引和从后向前索引两种方式



与字符串相同，利用下标 “[]” 可以从已有列表中取出其中部分元素形成一个新列表，其语法格式为：

```
ls[beg:end]
```

其中，beg是要取出的部分元素在ls中的起始下标，end是要取出的部分元素在ls中的结束下标。

# 列表元素索引



## 提示

省略beg，则表示从ls中的第一个元素开始，等价于ls[0:end]；省略end，则表示要取出的部分元素从beg位置开始一直到最后一个元素（包括最后一个元素）；beg和end都省略则取出ls中的所有元素。

# 列表元素索引

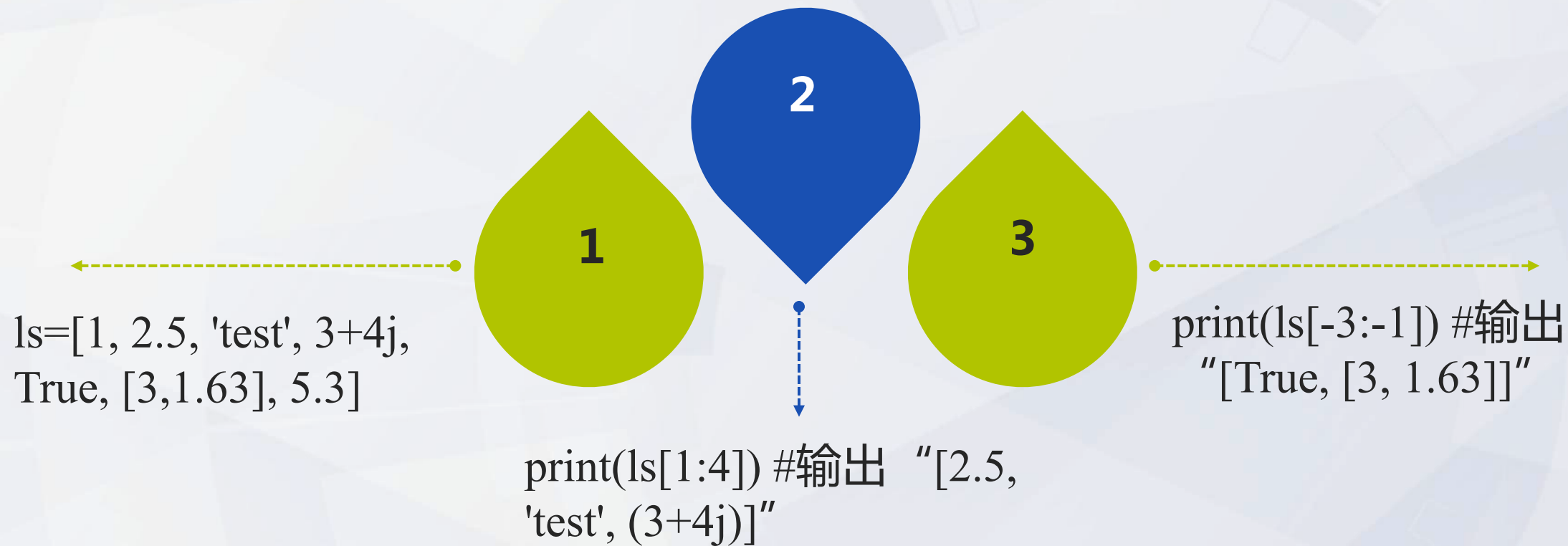
例如



对于ls=[1, 2.5, 'test', 3+4j, True, [3,1.63], 5.3]这个列表，其各元素的下标为：

列表	1	2.5	'test'	3+4j	True	[3,1.63]	5.3
从前向后索引	0	1	2	3	4	5	6
从后向前索引	-7	-6	-5	-4	-3	-2	-1

# 列表元素索引



# 列表元素索引

4

▶ `print(ls[2:-1])` #输出 `"['test', (3+4j), True, [3, 1.63]]"`

5

▶ `print(ls[:3])` #输出 `"[1, 2.5, 'test']"`

6

▶ `print(ls[-2:])` #输出 `"[[3, 1.63], 5.3]"`

7

▶ `print(ls[:])` #输出 `"[1, 2.5, 'test', (3+4j), True, [3, 1.63], 5.3]"`



# 访问单个元素



如果只访问列表ls中的某一个元素，则可以使用下面的写法：

```
ls[idx]
```

其中，idx是要访问的元素的下标。



例如：

```
1      ls=[1, 2.5, 'test', 3+4j, True, [3,1.63], 5.3]
```

```
2      print(ls[2]) #输出 "test"
```

```
3      print(ls[-3]) #输出 "True"
```

# 访问单个元素

例如

注意

`ls[beg:end]`返回的仍然是一个列表；而`ls[idx]`返回的是列表中的一个元素。

可见

`ls[2:3]`返回的是只有一个字符串元素'`test`'的列表，而`ls[2]`返回的则是`ls`中第3个元素的值（即字符串'`test`'）。

对于`ls=[1, 2.5, 'test', 3+4j, True, [3,1.63], 5.3]`，通过“`print(ls[2:3])`”和“`print(ls[2])`”输出的结果分别是“`['test']`”和“`test`”。

# 元素修改

通过下标 “[]” 不仅可以访问列表中的某个元素，还可以对元素进行修改。例如：

1

▶ `ls=[1, 2.5, 'test', 3+4j, True, [3,1.63], 5.3]`

2

▶ `print(ls)` #输出 “[1, 2.5, 'test', (3+4j), True, [3, 1.63], 5.3]”

3

▶ `ls[2]=15` #将列表ls中第3个元素的值改为15

## 元素修改

`print(ls)` #输出 “[1, 2.5, 15, (3+4j), True, [3, 1.63], 5.3]”

4

5

`ls[1:4]=['python',20]` #将列表ls中第2至4个元素替换为['python',20]中的元素

# 元素修改



6

```
print(ls) #输出 "[1, 'python', 20, True, [3, 1.63], 5.3]"
```



7

```
ls[2]=['program',23.15] #将列表ls中第3个元素替换为['program',23.15]
```



8

```
print(ls) #输出 "[1, 'python', ['program', 23.15], True, [3, 1.63], 5.3]"
```

# 元素修改



9

`ls[0:2]=[]` #将列表ls中前两个元素替换为空列表[], 即将前两个元素删除



10

`print(ls)` #输出 `"['program', 23.15], True, [3, 1.63], 5.3]"`

## 元素修改



注意：

在对列表中的元素赋值时，既可以通过“`ls[idx]=a`”这种方式修改单个元素的值，也可以通过“`ls[beg:end]=b`”这种方式修改一个元素或同时修改连续多个元素的值。

但需要注意，在通过“`ls[beg:end]=b`”这种方式赋值时，`b`是另一个列表，其功能是用`b`中各元素替换`ls`中`beg`至`end-1`这些位置上的元素，赋值前后列表元素数量允许发生变化。

# 元素修改



例如

上面所示的代码中，第3行和第7行都是修改列表ls中某一个元素的值，在为单个元素赋值时，可以使用任意类型的数据（包括列表，如第7行）；第5行是将列表ls中第2至4个元素修改为另一个列表['python',20]中的两个元素；第9行是将列表ls中前两个元素修改为另一个空列表[]中的元素，相当于将ls中前两个元素删除。