

Python内置数据结构

讲师：Wayne

从业十余载，漫漫求知路

bytes、 bytearray

- Python3引入两个新类型

- bytes

- **不可变**字节序列

- bytearray

- 字节数组

- **可变**

bytes、bytearray

□ 字符串与bytes

- 字符串是字符组成的有序序列，字符可以使用编码来理解
- bytes是字节组成的有序的**不可变**序列
- bytearray是字节组成的有序的**可变**序列

□ 编码与解码

- 字符串按照不同的字符集编码encode返回字节序列bytes
 - `encode(encoding='utf-8', errors='strict') -> bytes`
- 字节序列按照不同的字符集解码decode返回字符串
 - `bytes.decode(encoding="utf-8", errors="strict") -> str`
 - `bytearray.decode(encoding="utf-8", errors="strict") -> str`

ASCII

- ASCII (American Standard Code for Information Interchange , 美国信息交换标准代码) 是基于拉丁字母的一套单字节编码系统

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

bytes定义

□ 定义

- `bytes()` 空bytes
- `bytes(int)` 指定字节的bytes , 被0填充
- `bytes(iterable_of_ints)` -> bytes [0,255]的int组成的可迭代对象
- `bytes(string, encoding[, errors])` -> bytes 等价于`string.encode()`
- `bytes(bytes_or_buffer)` -> immutable copy of bytes_or_buffer 从一个字节序列或者buffer复制出一个新的不可变的bytes对象
- 使用b前缀定义
 - 只允许基本ASCII使用字符形式`b'abc9'`
 - 使用16进制表示`b"\x41\x61"`

bytes操作

- 和str类型类似，都是不可变类型，所以方法很多都一样。只不过bytes的方法，输入是bytes，输出是bytes

```
b'abcdef'.replace(b'f',b'k')
```

```
b'abc'.find(b'b')
```

- 类方法 bytes.fromhex(string)

- string必须是2个字符的16进制的形式，'6162 6a 6b'，空格将被忽略

```
bytes.fromhex('6162 09 6a 6b00')
```

- hex()

- 返回16进制表示的字符串

```
'abc'.encode().hex()
```

- 索引

```
b'abcdef'[2] 返回该字节对应的数，int类型
```

bytearray定义

□ 定义

- bytearray() 空bytearray
 - bytearray(int) 指定字节的bytearray，被0填充
 - bytearray(iterable_of_ints) -> bytearray [0,255]的int组成的可迭代对象
 - bytearray(string, encoding[, errors]) -> bytearray 近似string.encode()，不过返回可变对象
 - bytearray(bytes_or_buffer) 从一个字节序列或者buffer复制出一个新的可变的bytearray对象
-
- 注意，b前缀定义的类型是bytes类型

bytearray操作

- 和bytes类型的方法相同

`bytearray(b'abcdef').replace(b'f',b'k')`

`bytearray(b'abc').find(b'b')`

- 类方法 `bytearray.fromhex(string)`

- string必须是2个字符的16进制的形式，'6162 6a 6b'，空格将被忽略

`bytearray.fromhex('6162 09 6a 6b00')`

- `hex()`

- 返回16进制表示的字符串

`bytearray('abc'.encode()).hex()`

- 索引

`bytearray(b'abcdef')[2]` 返回该字节对应的数，int类型

bytearray操作

- ❑ `append(int)` 尾部追加一个元素
- ❑ `insert(index, int)` 在指定索引位置插入元素
- ❑ `extend(iterable_of_ints)` 将一个可迭代的整数集合追加到当前bytearray
- ❑ `pop(index=-1)` 从指定索引上移除元素，默认从尾部移除
- ❑ `remove(value)` 找到第一个value移除，找不到抛 `ValueError`异常
- ❑ 注意：上述方法若需要使用int类型，值在[0, 255]
- ❑ `clear()` 清空bytearray
- ❑ `reverse()` 翻转bytearray，就地修改

```
b = bytearray()
b.append(97)
b.append(99)
b.insert(1,98)
b.extend([65,66,67])
b.remove(66)
b.pop()
b.reverse()
b.clear()
```

int和bytes

- `int.from_bytes(bytes, byteorder)`

- 将一个字节数组表示成整数

- `int.to_bytes(length, byteorder)`

- `byteorder`字节序

- 将一个整数表达成一个指定长度的字节数组

```
i = int.from_bytes(b'abc', 'big')
```

```
print(i, hex(i)) # 6382179 0x616263
```

```
print(i.to_bytes(3, 'big')) # b'abc'
```

谢谢

咨询热线 400-080-6560