

bytearray

Description

Returns a new array of bytes.

Syntax

`bytearray ([source[, encoding[,errors]]])`

source

Optional. If source is:

- a string, encoding is required.
- an integer, the array will have that size and will be initialized with null bytes.
- an object conforming to the buffer interface, a read-only buffer of the object will be used to initialize the bytes array.
- an iterable, it must be an iterable of integers in the range 0-256, which are used to initialize the array
- without an argument, an array of size 0 is created.

encoding

Optional. Required if source is a string. Typical values: 'ascii', 'utf-8', 'windows-1250', 'windows-1252'. See codecs module for more.

errors

Optional. Possible values for errors are:

- 'strict': raise an exception in case of an encoding error
- 'replace': replace malformed data with a suitable replacement marker, such as '?' or 'ufffd'
- 'ignore': ignore malformed data and continue without further notice

- 'xmlcharrefreplace': replace with the appropriate XML character reference (for encoding only)
- 'backslashreplace': replace with backslashed escape sequences (for encoding only)

Return Value

bytearray

Time Complexity

#TODO

Remarks

The bytearray type is a mutable sequence of integers in the range $0 \leq x < 256$. It can be used to work with low-level binary data such as that inside of images or arriving directly from the network.

Example 1

```
>>> bytearray()  
bytearray(b'')  
>>> bytearray(4)  
bytearray(b'\x00\x00\x00\x00')  
>>> bytearray([0,1,2])  
bytearray(b'\x00\x01\x02')  
>>> bytearray(buffer('hello'))  
bytearray(b'hello')
```

Example 2

```
>>> bytearray('hello', 'ascii')
bytearray(b'hello')
>>> bytearray(u'źdźbło', 'ascii', 'strict')    #'blade of grass' in polish
UnicodeEncodeError: 'ascii' codec can't encode character u'\u017a' in position 0: ordinal not in
range(128)
>>> bytearray(u'źdźbło', 'ascii', 'ignore')
bytearray(b'dbo')
>>> bytearray(u'źdźbło', 'ascii', 'replace')
bytearray(b'?d?b?o')
>>> bytearray(u'źdźbło', 'ascii', 'xmlcharrefreplace')
bytearray(b'&#378;d&#378;b&#322;o')
>>> bytearray(u'źdźbło', 'ascii', 'backslashreplace')
bytearray(b'\\u017ad\\u017ab\\u0142o')
```

See Also

#TODO