Built-in Types

The following sections describe the standard types that are built into the interpreter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 13)

Unknown directive type "index".

```
.. index:: pair: built-in; types
```

The principal built-in types are numerics, sequences, mappings, classes, instances and exceptions.

Some collection classes are mutable. The methods that add, subtract, or rearrange their members in place, and don't return a specific item, never return the collection instance itself but `None`.

Some operations are supported by several object types; in particular, practically all objects can be compared for equality, tested for truth value, and converted to a string (with the :func:`repr` function or the slightly different :func:`str` function). The latter function is implicitly used when an object is written by the :func:`print` function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 22); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 22); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 22); *backlink*

Unknown interpreted text role "func".

Truth Value Testing

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 34)

Unknown directive type "index".

```
.. index::
   statement: if
   statement: while
   pair: truth; value
   pair: Boolean; operations
   single: false
```

Any object can be tested for truth value, for use in an :keyword:`if` or :keyword:`while` condition or as operand of the Boolean operations below.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 41); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 41); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 44)

Unknown directive type "index".

```
.. index:: single: true
```

By default, an object is considered true unless its class defines either a :meth:`__bool__` method that returns `False` or a :meth:`__len__` method that returns zero, when called with the object. [1] Here are most of the built-in objects considered false:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 46); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 46); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 51)

Unknown directive type "index".

```
.. index::
   single: None (Built-in object)
   single: False (Built-in object)
```

- constants defined to be false: `None` and `False`.
- zero of any numeric type: `0`, `0.0`, `0j`, `Decimal(0)`, `Fraction(0, 1)`
- empty sequences and collections: `''`, `()`, `[]`, `{}`, `set()`, `range(0)`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 63)

Unknown directive type "index".

```
.. index::
   operator: or
   operator: and
   single: False
   single: True
```

Operations and built-in functions that have a Boolean result always return `0` or `False` for false and `1` or `True` for true, unless otherwise stated. (Important exception: the Boolean operations `or` and `and` always return one of their operands.)

Boolean Operations --- :keyword:`!and`, :keyword:`!or`, :keyword:`!not`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 77); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 77); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 77); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 80)

Unknown directive type "index".

```
.. index:: pair: Boolean; operations
```

These are the Boolean operations, ordered by ascending priority:

Operation	Result	Notes
`x or y`	if *x* is false, then *y*, else *x*	(1)
`x and y`	if *x* is false, then *x*, else *y*	(2)
`not x`	if *x* is false, then `True`, else `False`	(3)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 97)

Unknown directive type "index".

```
.. index::
   operator: and
   operator: or
   operator: not
```

Notes:

1. This is a short-circuit operator, so it only evaluates the second argument if the first one is false.
2. This is a short-circuit operator, so it only evaluates the second argument if the first one is true.
3. `not` has a lower priority than non-Boolean operators, so `not a == b` is interpreted as `not (a == b)`, and `a == not b` is a syntax error.

Comparisons

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 122)

Unknown directive type "index".

```
.. index::
   pair: chaining; comparisons
   pair: operator; comparison
   operator: ==
   operator: < (less)
   operator: <=
   operator: > (greater)
   operator: >=
   operator: !=
   operator: is
   operator: is not
```

There are eight comparison operations in Python. They all have the same priority (which is higher than that of the Boolean operations). Comparisons can be chained arbitrarily; for example, `x < y <= z` is equivalent to `x < y and y <= z`, except that *y* is evaluated only once (but in both cases *z* is not evaluated at all when `x < y` is found to be false).

This table summarizes the comparison operations:

Operation	Meaning
`<`	strictly less than
`<=`	less than or equal
`>`	strictly greater than
`>=`	greater than or equal
`==`	equal
`!=`	not equal
`is`	object identity
`is not`	negated object identity

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 162)

Unknown directive type "index".

```
.. index::
   pair: object; numeric
   pair: objects; comparing
```

Objects of different types, except different numeric types, never compare equal. The == operator is always defined but for some object types (for example, class objects) is equivalent to :keyword:`is`. The <, <=, > and >= operators are only defined where they make sense; for example, they raise a :exc:`TypeError` exception when one of the arguments is a complex number.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 166); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 166); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 172)

Unknown directive type "index".

```
.. index::
   single: __eq__() (instance method)
   single: __ne__() (instance method)
   single: __lt__() (instance method)
   single: __le__() (instance method)
   single: __gt__() (instance method)
   single: __ge__() (instance method)
```

Non-identical instances of a class normally compare as non-equal unless the class defines the :meth:`~object.__eq__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 180); *backlink*

Unknown interpreted text role "meth".

Instances of a class cannot be ordered with respect to other instances of the same class, or other types of object, unless the class defines enough of the methods :meth:`~object.__lt__`, :meth:`~object.__le__`, :meth:`~object.__gt__`, and :meth:`~object.__ge__` (in general, :meth:`~object.__lt__` and :meth:`~object.__eq__` are sufficient, if you want the conventional meanings of the comparison operators).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 183); *backlink*

Unknown interpreted text role "meth".

The behavior of the :keyword:`is` and :keyword:`is not` operators cannot be customized; also they can be applied to any two objects and never raise an exception.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 190); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 190); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 194)

Unknown directive type "index".

```
.. index::
   operator: in
   operator: not in
```

Two more operations with the same syntactic priority, :keyword:`in` and :keyword:`not in`, are supported by types that are :term:`iterable` or implement the :meth:`__contains__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 198); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 198); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 198); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 198); *backlink*

Unknown interpreted text role "meth".

Numeric Types --- :class:`int`, :class:`float`, :class:`complex`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 204); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 204); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 204); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 207)

Unknown directive type "index".

```
.. index::
   object: numeric
   object: Boolean
```

```
object: integer
object: floating point
object: complex number
pair: C; language
```

There are three distinct numeric types: :dfn:`integers`, :dfn:`floating point numbers`, and :dfn:`complex numbers`. In addition, Booleans are a subtype of integers. Integers have unlimited precision. Floating point numbers are usually implemented using :c:type:`double` in C; information about the precision and internal representation of floating point numbers for the machine on which your program is running is available in :data:`sys.float_info`. Complex numbers have a real and imaginary part, which are each a floating point number. To extract these parts from a complex number *z*, use `z.real` and `z.imag`. (The standard library includes the additional numeric types :mod:`fractions.Fraction`, for rationals, and :mod:`decimal.Decimal`, for floating-point numbers with user-definable precision.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 215); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 228)

Unknown directive type "index".

```
.. index::
   pair: numeric; literals
   pair: integer; literals
   pair: floating point; literals
   pair: complex number; literals
   pair: hexadecimal; literals
   pair: octal; literals
   pair: binary; literals
```

Numbers are created by numeric literals or as the result of built-in functions and operators. Unadorned integer literals (including hex, octal and binary numbers) yield integers. Numeric literals containing a decimal point or an exponent sign yield floating point numbers. Appending `'j'` or `'J'` to a numeric literal yields an imaginary number (a complex number with a zero real part) which you can add to an integer or float to get a complex number with real and imaginary parts.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 245)

Unknown directive type "index".

```
.. index::
   single: arithmetic
   builtin: int
   builtin: float
   builtin: complex
   single: operator; + (plus)
   single: + (plus); unary operator
   single: + (plus); binary operator
   single: operator; - (minus)
   single: - (minus); unary operator
   single: - (minus); binary operator
   operator: * (asterisk)
   operator: / (slash)
   operator: //
   operator: % (percent)
   operator: **
```

Python fully supports mixed arithmetic: when a binary arithmetic operator has operands of different numeric types, the operand with the "narrower" type is widened to that of the other, where integer is narrower than floating point, which is narrower than complex. A comparison between numbers of different types behaves as though the exact values of those numbers were being compared. [2]

The constructors :func:`int`, :func:`float`, and :func:`complex` can be used to produce numbers of a specific type.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 268); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 268); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 268); *backlink*

Unknown interpreted text role "func".

All numeric types (except complex) support the following operations (for priorities of the operations, see :ref:`operator-summary`):

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 271); *backlink*

Unknown interpreted text role "ref".

Operation	Result	Notes	Full documentation
`x + y`	sum of *x* and *y*		
`x - y`	difference of *x* and *y*		
`x * y`	product of *x* and *y*		
`x / y`	quotient of *x* and *y*		
`x // y`	floored quotient of *x* and *y*	(1)	
`x % y`	remainder of `x / y`	(2)	
`-x`	*x* negated		
`+x`	*x* unchanged		

Operation	Result	Notes	Full documentation
`abs(x)`	absolute value or magnitude of *x*		:func:`abs` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 295); *backlink* Unknown interpreted text role "func".
`int(x)`	*x* converted to integer	(3)(6)	:func:`int` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 298); *backlink* Unknown interpreted text role "func".
`float(x)`	*x* converted to floating point	(4)(6)	:func:`float` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 300); *backlink* Unknown interpreted text role "func".
`complex(re, im)`	a complex number with real part *re*, imaginary part *im*. *im* defaults to zero.	(6)	:func:`complex` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 302); *backlink* Unknown interpreted text role "func".
`c.conjugate()`	conjugate of the complex number *c*		

Operation	Result	Notes	Full documentation
`divmod(x, y)`	the pair `(x // y, x % y)`	(2)	:func:`divmod` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 309); *backlink*** Unknown interpreted text role "func".
`pow(x, y)`	*x* to the power *y*	(5)	:func:`pow` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 311); *backlink*** Unknown interpreted text role "func".
`x ** y`	*x* to the power *y*	(5)	

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 315)

Unknown directive type "index".

```
.. index::
   triple: operations on; numeric; types
   single: conjugate() (complex number method)
```

Notes:

1. Also referred to as integer division. The resultant value is a whole integer, though the result's type is not necessarily int. The result is always rounded towards minus infinity: `1//2` is `0`, `(-1)//2` is `-1`, `1//(-2)` is `-1`, and `(-1)//(-2)` is `0`.
2. Not for complex numbers. Instead convert to floats using :func:`abs` if appropriate.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 328); *backlink*

   Unknown interpreted text role "func".

3. **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 332)**

   Unknown directive type "index".

   ```
   .. index::
      module: math
      single: floor() (in module math)
      single: ceil() (in module math)
      single: trunc() (in module math)
      pair: numeric; conversions
      pair: C; language
   ```

Conversion from floating point to integer may round or truncate as in C; see functions :func:`math.floor` and :func:`math.ceil` for well-defined conversions.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 340); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 340); *backlink*

Unknown interpreted text role "func".

4. float also accepts the strings "nan" and "inf" with an optional prefix "+" or "-" for Not a Number (NaN) and positive or negative infinity.
5. Python defines `pow(0, 0)` and `0 ** 0` to be `1`, as is common for programming languages.
6. The numeric literals accepted include the digits `0` to `9` or any Unicode equivalent (code points with the `Nd` property).

   See https://www.unicode.org/Public/14.0.0/ucd/extracted/DerivedNumericType.txt for a complete list of code points with the `Nd` property.

All :class:`numbers.Real` types (:class:`int` and :class:`float`) also include the following operations:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 360); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 360); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 360); *backlink*

Unknown interpreted text role "class".

Operation	Result
:func:`math.trunc(\ x) <math.trunc>` System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 367); *backlink* Unknown interpreted text role "func".	*x* truncated to :class:`~numbers.Integral` System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 367); *backlink* Unknown interpreted text role "class".

Operation	Result
:func:`round(x[, n]) <round>` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 370); *backlink*** Unknown interpreted text role "func".	*x* rounded to *n* digits, rounding half to even. If *n* is omitted, it defaults to 0.
:func:`math.floor(\ x) <math.floor>` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 374); *backlink*** Unknown interpreted text role "func".	the greatest :class:`~numbers.Integral` <= *x* **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 374); *backlink*** Unknown interpreted text role "class".
:func:`math.ceil(x) <math.ceil>` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 377); *backlink*** Unknown interpreted text role "func".	the least :class:`~numbers.Integral` >= *x* **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 377); *backlink*** Unknown interpreted text role "class".

For additional numeric operations see the :mod:`math` and :mod:`cmath` modules.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 380); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 380); *backlink*

Unknown interpreted text role "mod".

Bitwise Operations on Integer Types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 391)

Unknown directive type "index".

```
.. index::
   triple: operations on; integer; types
   pair: bitwise; operations
   pair: shifting; operations
   pair: masking; operations
   operator: | (vertical bar)
   operator: ^ (caret)
   operator: & (ampersand)
   operator: <<
   operator: >>
   operator: ~ (tilde)
```

Bitwise operations only make sense for integers. The result of bitwise operations is calculated as though carried out in two's complement with an infinite number of sign bits.

The priorities of the binary bitwise operations are all lower than the numeric operations and higher than the comparisons; the unary operation ~ has the same priority as the other unary numeric operations (+ and -).

This table lists the bitwise operations sorted in ascending priority:

Operation	Result	Notes
`x | y`	bitwise :dfn:`or` of *x* and *y* **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 417); *backlink* Unknown interpreted text role "dfn".	(4)
`x ^ y`	bitwise :dfn:`exclusive or` of *x* and *y* **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 420); *backlink* Unknown interpreted text role "dfn".	(4)
`x & y`	bitwise :dfn:`and` of *x* and *y* **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 423); *backlink* Unknown interpreted text role "dfn".	(4)
`x << n`	*x* shifted left by *n* bits	(1)(2)
`x >> n`	*x* shifted right by *n* bits	(1)(3)
`~x`	the bits of *x* inverted	

Notes:

1. Negative shift counts are illegal and cause a :exc:`ValueError` to be raised.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 435); *backlink*
   Unknown interpreted text role "exc".

2. A left shift by *n* bits is equivalent to multiplication by `pow(2, n)`.
3. A right shift by *n* bits is equivalent to floor division by `pow(2, n)`.
4. Performing these calculations with at least one extra sign extension bit in a finite two's complement representation (a working bit-width of `1 + max(x.bit_length(), y.bit_length())` or more) is sufficient to get the same result as if there were an infinite number of sign bits.

Additional Methods on Integer Types

The int type implements the :class:`numbers.Integral` :term:`abstract base class`. In addition, it provides a few more methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 453); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 453); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 456)

Unknown directive type "method".

```
.. method:: int.bit_length()

    Return the number of bits necessary to represent an integer in binary,
    excluding the sign and leading zeros::

        >>> n = -37
        >>> bin(n)
        '-0b100101'
        >>> n.bit_length()
        6

    More precisely, if ``x`` is nonzero, then ``x.bit_length()`` is the
    unique positive integer ``k`` such that ``2**(k-1) <= abs(x) < 2**k``.
    Equivalently, when ``abs(x)`` is small enough to have a correctly
    rounded logarithm, then ``k = 1 + int(log(abs(x), 2))``.
    If ``x`` is zero, then ``x.bit_length()`` returns ``0``.

    Equivalent to::

        def bit_length(self):
            s = bin(self)       # binary representation:  bin(-37) --> '-0b100101'
            s = s.lstrip('-0b') # remove leading zeros and minus sign
            return len(s)       # len('100101') --> 6

    .. versionadded:: 3.1
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 482)

Unknown directive type "method".

```
.. method:: int.bit_count()

    Return the number of ones in the binary representation of the absolute
    value of the integer. This is also known as the population count.
    Example::

        >>> n = 19
        >>> bin(n)
        '0b10011'
        >>> n.bit_count()
        3
        >>> (-n).bit_count()
        3

    Equivalent to::

        def bit_count(self):
            return bin(self).count("1")

    .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 503)

Unknown directive type "method".

```
.. method:: int.to_bytes(length=1, byteorder='big', *, signed=False)

    Return an array of bytes representing an integer.

        >>> (1024).to_bytes(2, byteorder='big')
        b'\x04\x00'
        >>> (1024).to_bytes(10, byteorder='big')
        b'\x00\x00\x00\x00\x00\x00\x00\x00\x04\x00'
        >>> (-1024).to_bytes(10, byteorder='big', signed=True)
        b'\xff\xff\xff\xff\xff\xff\xff\xff\xfc\x00'
        >>> x = 1000
        >>> x.to_bytes((x.bit_length() + 7) // 8, byteorder='little')
        b'\xe8\x03'

    The integer is represented using *length* bytes, and defaults to 1.  An
    :exc:`OverflowError` is raised if the integer is not representable with
    the given number of bytes.

    The *byteorder* argument determines the byte order used to represent the
    integer, and defaults to ``"big"``.  If *byteorder* is
    ``"big"``, the most significant byte is at the beginning of the byte
    array.  If *byteorder* is ``"little"``, the most significant byte is at
    the end of the byte array.

    The *signed* argument determines whether two's complement is used to
    represent the integer.  If *signed* is ``False`` and a negative integer is
    given, an :exc:`OverflowError` is raised. The default value for *signed*
    is ``False``.

    The default values can be used to conveniently turn an integer into a
    single byte object.  However, when using the default arguments, don't try
    to convert a value greater than 255 or you'll get an :exc:`OverflowError`::

        >>> (65).to_bytes()
        b'A'

    Equivalent to::

        def to_bytes(n, length=1, byteorder='big', signed=False):
            if byteorder == 'little':
                order = range(length)
            elif byteorder == 'big':
                order = reversed(range(length))
            else:
                raise ValueError("byteorder must be either 'little' or 'big'")

            return bytes((n >> i*8) & 0xff for i in order)

    .. versionadded:: 3.2
    .. versionchanged:: 3.11
       Added default argument values for ``length`` and ``byteorder``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 555)

Unknown directive type "classmethod".

```
.. classmethod:: int.from_bytes(bytes, byteorder='big', *, signed=False)

    Return the integer represented by the given array of bytes.

        >>> int.from_bytes(b'\x00\x10', byteorder='big')
        16
        >>> int.from_bytes(b'\x00\x10', byteorder='little')
        4096
        >>> int.from_bytes(b'\xfc\x00', byteorder='big', signed=True)
        -1024
        >>> int.from_bytes(b'\xfc\x00', byteorder='big', signed=False)
        64512
        >>> int.from_bytes([255, 0, 0], byteorder='big')
        16711680

    The argument *bytes* must either be a :term:`bytes-like object` or an
    iterable producing bytes.

    The *byteorder* argument determines the byte order used to represent the
    integer, and defaults to ``"big"``.  If *byteorder* is
    ``"big"``, the most significant byte is at the beginning of the byte
    array.  If *byteorder* is ``"little"``, the most significant byte is at
    the end of the byte array.  To request the native byte order of the host
    system, use :data:`sys.byteorder` as the byte order value.
```

```
The *signed* argument indicates whether two's complement is used to
represent the integer.

Equivalent to::

    def from_bytes(bytes, byteorder='big', signed=False):
        if byteorder == 'little':
            little_ordered = list(bytes)
        elif byteorder == 'big':
            little_ordered = list(reversed(bytes))
        else:
            raise ValueError("byteorder must be either 'little' or 'big'")

        n = sum(b << i*8 for i, b in enumerate(little_ordered))
        if signed and little_ordered and (little_ordered[-1] & 0x80):
            n -= 1 << 8*len(little_ordered)

        return n

.. versionadded:: 3.2
.. versionchanged:: 3.11
   Added default argument value for ``byteorder``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 603)

Unknown directive type "method".

```
.. method:: int.as_integer_ratio()

   Return a pair of integers whose ratio is exactly equal to the original
   integer and with a positive denominator. The integer ratio of integers
   (whole numbers) is always the integer as the numerator and ``1`` as the
   denominator.

   .. versionadded:: 3.8
```

Additional Methods on Float

The float type implements the :class:`numbers.Real` :term:`abstract base class`. float also has the following additional methods.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 615); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 615); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 618)

Unknown directive type "method".

```
.. method:: float.as_integer_ratio()

   Return a pair of integers whose ratio is exactly equal to the
   original float and with a positive denominator.  Raises
   :exc:`OverflowError` on infinities and a :exc:`ValueError` on
   NaNs.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 625)

Unknown directive type "method".

```
.. method:: float.is_integer()

   Return ``True`` if the float instance is finite with integral
   value, and ``False`` otherwise::

      >>> (-2.0).is_integer()
      True
```

```
>>> (3.2).is_integer()
False
```

Two methods support conversion to and from hexadecimal strings. Since Python's floats are stored internally as binary numbers, converting a float to or from a *decimal* string usually involves a small rounding error. In contrast, hexadecimal strings allow exact representation and specification of floating-point numbers. This can be useful when debugging, and in numerical work.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 644)

Unknown directive type "method".

```
.. method:: float.hex()

   Return a representation of a floating-point number as a hexadecimal
   string.  For finite floating-point numbers, this representation
   will always include a leading ``0x`` and a trailing ``p`` and
   exponent.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 652)

Unknown directive type "classmethod".

```
.. classmethod:: float.fromhex(s)

   Class method to return the float represented by a hexadecimal
   string *s*.  The string *s* may have leading and trailing
   whitespace.
```

Note that :meth:`float.hex` is an instance method, while :meth:`float.fromhex` is a class method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 659); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 659); *backlink*

Unknown interpreted text role "meth".

A hexadecimal string takes the form:

```
[sign] ['0x'] integer ['.' fraction] ['p' exponent]
```

where the optional `sign` may by either `+` or `-`, `integer` and `fraction` are strings of hexadecimal digits, and `exponent` is a decimal integer with an optional leading sign. Case is not significant, and there must be at least one hexadecimal digit in either the integer or the fraction. This syntax is similar to the syntax specified in section 6.4.4.2 of the C99 standard, and also to the syntax used in Java 1.5 onwards. In particular, the output of :meth:`float.hex` is usable as a hexadecimal floating-point literal in C or Java code, and hexadecimal strings produced by C's `%a` format character or Java's `Double.toHexString` are accepted by :meth:`float.fromhex`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 666); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 666); *backlink*

Unknown interpreted text role "meth".

Note that the exponent is written in decimal rather than hexadecimal, and that it gives the power of 2 by which to multiply the coefficient. For example, the hexadecimal string `0x3.a7p10` represents the floating-point number `(3 + 10./16 + 7./16**2) * 2.0**10`, or `3740.0`:

```
>>> float.fromhex('0x3.a7p10')
3740.0
```

Applying the reverse conversion to `3740.0` gives a different hexadecimal string representing the same number:

```
>>> float.hex(3740.0)
'0x1.d380000000000p+11'
```

Hashing of numeric types

For numbers `x` and `y`, possibly of different types, it's a requirement that `hash(x) == hash(y)` whenever `x == y` (see the :meth:`~object.__hash__` method documentation for more details). For ease of implementation and efficiency across a variety of numeric types (including :class:`int`, :class:`float`, :class:`decimal.Decimal` and :class:`fractions.Fraction`) Python's hash for numeric types is based on a single mathematical function that's defined for any rational number, and hence applies to all instances of :class:`int` and :class:`fractions.Fraction`, and all finite instances of :class:`float` and :class:`decimal.Decimal`. Essentially, this function is given by reduction modulo `P` for a fixed prime `P`. The value of `P` is made available to Python as the :attr:`modulus` attribute of :data:`sys.hash_info`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 701); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 714)

Unknown directive type "impl-detail".

```
.. impl-detail::

   Currently, the prime used is ``P = 2**31 - 1`` on machines with 32-bit C
   longs and ``P = 2**61 - 1`` on machines with 64-bit C longs.
```

Here are the rules in detail:

- If `x = m / n` is a nonnegative rational number and `n` is not divisible by `P`, define `hash(x)` as `m * invmod(n, P) % P`, where `invmod(n, P)` gives the inverse of `n` modulo `P`.
- If `x = m / n` is a nonnegative rational number and `n` is divisible by `P` (but `m` is not) then `n` has no inverse modulo `P` and the rule above doesn't apply; in this case define `hash(x)` to be the constant value `sys.hash_info.inf`.
- If `x = m / n` is a negative rational number define `hash(x)` as `-hash(-x)`. If the resulting hash is `-1`, replace it with `-2`.
- The particular values `sys.hash_info.inf` and `-sys.hash_info.inf` are used as hash values for positive infinity or negative infinity (respectively).
- For a :class:`complex` number `z`, the hash values of the real and imaginary parts are combined by computing `hash(z.real) + sys.hash_info.imag * hash(z.imag)`, reduced modulo `2**sys.hash_info.width` so that it lies in `range(-2**(sys.hash_info.width - 1), 2**(sys.hash_info.width - 1))`. Again, if the result is `-1`, it's replaced with `-2`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 738); *backlink*

Unknown interpreted text role "class".

To clarify the above rules, here's some example Python code, equivalent to the built-in hash, for computing the hash of a rational number, :class:`float`, or :class:`complex`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 746); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 746); *backlink*

Unknown interpreted text role "class".

```
import sys, math

def hash_fraction(m, n):
    """Compute the hash of a rational number m / n.

    Assumes m and n are integers, with n positive.
    Equivalent to hash(fractions.Fraction(m, n)).

    """
    P = sys.hash_info.modulus
    # Remove common factors of P.  (Unnecessary if m and n already coprime.)
    while m % P == n % P == 0:
        m, n = m // P, n // P

    if n % P == 0:
        hash_value = sys.hash_info.inf
    else:
        # Fermat's Little Theorem: pow(n, P-1, P) is 1, so
        # pow(n, P-2, P) gives the inverse of n modulo P.
        hash_value = (abs(m) % P) * pow(n, P - 2, P) % P
    if m < 0:
        hash_value = -hash_value
    if hash_value == -1:
        hash_value = -2
    return hash_value

def hash_float(x):
```

```
        """Compute the hash of a float x."""

        if math.isnan(x):
            return object.__hash__(x)
        elif math.isinf(x):
            return sys.hash_info.inf if x > 0 else -sys.hash_info.inf
        else:
            return hash_fraction(*x.as_integer_ratio())

    def hash_complex(z):
        """Compute the hash of a complex number z."""

        hash_value = hash_float(z.real) + sys.hash_info.imag * hash_float(z.imag)
        # do a signed reduction modulo 2**sys.hash_info.width
        M = 2**(sys.hash_info.width - 1)
        hash_value = (hash_value & (M - 1)) - (hash_value & M)
        if hash_value == -1:
            hash_value = -2
        return hash_value
```

Iterator Types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 803)

Unknown directive type "index".

```
.. index::
   single: iterator protocol
   single: protocol; iterator
   single: sequence; iteration
   single: container; iteration over
```

Python supports a concept of iteration over containers. This is implemented using two distinct methods; these are used to allow user-defined classes to support iteration. Sequences, described below in more detail, always support the iteration methods.

One method needs to be defined for container objects to provide :term:`iterable` support:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 814); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 819)

Unknown directive type "method".

```
.. method:: container.__iter__()

   Return an :term:`iterator` object.  The object is required to support the
   iterator protocol described below.  If a container supports different types
   of iteration, additional methods can be provided to specifically request
   iterators for those iteration types.  (An example of an object supporting
   multiple forms of iteration would be a tree structure which supports both
   breadth-first and depth-first traversal.)  This method corresponds to the
   :c:member:`~PyTypeObject.tp_iter` slot of the type structure for Python
   objects in the Python/C API.
```

The iterator objects themselves are required to support the following two methods, which together form the :dfn:`iterator protocol`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 830); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 834)

Unknown directive type "method".

```
.. method:: iterator.__iter__()

   Return the :term:`iterator` object itself.  This is required to allow both
   containers and iterators to be used with the :keyword:`for` and
```

```
:keyword:`in` statements.  This method corresponds to the
:c:member:`~PyTypeObject.tp_iter` slot of the type structure for Python
objects in the Python/C API.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 843)

Unknown directive type "method".

```
.. method:: iterator.__next__()

   Return the next item from the :term:`iterator`.  If there are no further
   items, raise the :exc:`StopIteration` exception.  This method corresponds to
   the :c:member:`~PyTypeObject.tp_iternext` slot of the type structure for
   Python objects in the Python/C API.
```

Python defines several iterator objects to support iteration over general and specific sequence types, dictionaries, and other more specialized forms. The specific types are not important beyond their implementation of the iterator protocol.

Once an iterator's :meth:`~iterator.__next__` method raises :exc:`StopIteration`, it must continue to do so on subsequent calls. Implementations that do not obey this property are deemed broken.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 855); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 855); *backlink*

Unknown interpreted text role "exc".

Generator Types

Python's :term:`generator`s provide a convenient way to implement the iterator protocol. If a container object's :meth:`__iter__` method is implemented as a generator, it will automatically return an iterator object (technically, a generator object) supplying the :meth:`__iter__` and :meth:`~generator.__next__` methods. More information about generators can be found in :ref:`the documentation for the yield expression <yieldexpr>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 865); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 865); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 865); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 865); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 865); *backlink*

Unknown interpreted text role "ref".

Sequence Types --- :class:`list`, :class:`tuple`, :class:`range`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 876); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 876); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 876); *backlink*

Unknown interpreted text role "class".

There are three basic sequence types: lists, tuples, and range objects. Additional sequence types tailored for processing of :ref:`binary data <binaryseq>` and :ref:`text strings <textseq>` are described in dedicated sections.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 879); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 879); *backlink*

Unknown interpreted text role "ref".

Common Sequence Operations

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 890)

Unknown directive type "index".

```
.. index:: object: sequence
```

The operations in the following table are supported by most sequence types, both mutable and immutable. The :class:`collections.abc.Sequence` ABC is provided to make it easier to correctly implement these operations on custom sequence types.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 892); *backlink*

Unknown interpreted text role "class".

This table lists the sequence operations sorted in ascending priority. In the table, *s* and *t* are sequences of the same type, *n*, *i*, *j* and *k* are integers and *x* is an arbitrary object that meets any type and value restrictions imposed by *s*.

The `in` and `not in` operations have the same priorities as the comparison operations. The `+` (concatenation) and `*` (repetition) operations have the same priority as the corresponding numeric operations. [3]

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 906)

Unknown directive type "index".

```
.. index::
   triple: operations on; sequence; types
   builtin: len
   builtin: min
   builtin: max
   pair: concatenation; operation
   pair: repetition; operation
   pair: subscript; operation
   pair: slice; operation
   operator: in
   operator: not in
   single: count() (sequence method)
   single: index() (sequence method)
```

Operation	Result	Notes
`x in s`	`True` if an item of *s* is equal to *x*, else `False`	(1)
`x not in s`	`False` if an item of *s* is equal to *x*, else `True`	(1)
`s + t`	the concatenation of *s* and *t*	(6)(7)
`s * n` or `n * s`	equivalent to adding *s* to itself *n* times	(2)(7)
`s[i]`	*i*th item of *s*, origin 0	(3)
`s[i:j]`	slice of *s* from *i* to *j*	(3)(4)
`s[i:j:k]`	slice of *s* from *i* to *j* with step *k*	(3)(5)
`len(s)`	length of *s*	
`min(s)`	smallest item of *s*	
`max(s)`	largest item of *s*	
`s.index(x[, i[, j]])`	index of the first occurrence of *x* in *s* (at or after index *i* and before index *j*)	(8)
`s.count(x)`	total number of occurrences of *x* in *s*	

Sequences of the same type also support comparisons. In particular, tuples and lists are compared lexicographically by comparing corresponding elements. This means that to compare equal, every element must compare equal and the two sequences must be of the same type and have the same length. (For full details see :ref:`comparisons` in the language reference.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 956); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 962)

Unknown directive type "index".

```
.. index::
   single: loop; over mutable sequence
   single: mutable sequence; loop over
```

Forward and reversed iterators over mutable sequences access values using an index. That index will continue to march forward (or backward) even if the underlying sequence is mutated. The iterator terminates only when an :exc:`IndexError` or a :exc:`StopIteration` is encountered (or when the index drops below zero).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 966); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 966); *backlink*

Unknown interpreted text role "exc".

Notes:

1. While the `in` and `not in` operations are used only for simple containment testing in the general case, some specialised sequences (such as :class:`str`, :class:`bytes` and :class:`bytearray`) also use them for subsequence testing:

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 975); *backlink*

   Unknown interpreted text role "class".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 975); *backlink*

   Unknown interpreted text role "class".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line

975); *backlink*

Unknown interpreted text role "class".

```
>>> "gg" in "eggs"
True
```

2. Values of *n* less than `0` are treated as `0` (which yields an empty sequence of the same type as *s*). Note that items in the sequence *s* are not copied; they are referenced multiple times. This often haunts new Python programmers; consider:

   ```
   >>> lists = [[]] * 3
   >>> lists
   [[], [], []]
   >>> lists[0].append(3)
   >>> lists
   [[3], [3], [3]]
   ```

   What has happened is that `[[]]` is a one-element list containing an empty list, so all three elements of `[[]] * 3` are references to this single empty list. Modifying any of the elements of `lists` modifies this single list. You can create a list of different lists this way:

   ```
   >>> lists = [[] for i in range(3)]
   >>> lists[0].append(3)
   >>> lists[1].append(5)
   >>> lists[2].append(7)
   >>> lists
   [[3], [5], [7]]
   ```

   Further explanation is available in the FAQ entry :ref:`faq-multidimensional-list`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1008); *backlink*

   Unknown interpreted text role "ref".

3. If *i* or *j* is negative, the index is relative to the end of sequence *s*: `len(s) + i` or `len(s) + j` is substituted. But note that `-0` is still `0`.

4. The slice of *s* from *i* to *j* is defined as the sequence of items with index *k* such that `i <= k < j`. If *i* or *j* is greater than `len(s)`, use `len(s)`. If *i* is omitted or `None`, use `0`. If *j* is omitted or `None`, use `len(s)`. If *i* is greater than or equal to *j*, the slice is empty.

5. The slice of *s* from *i* to *j* with step *k* is defined as the sequence of items with index `x = i + n*k` such that `0 <= n < (j-i)/k`. In other words, the indices are `i`, `i+k`, `i+2*k`, `i+3*k` and so on, stopping when *j* is reached (but never including *j*). When *k* is positive, *i* and *j* are reduced to `len(s)` if they are greater. When *k* is negative, *i* and *j* are reduced to `len(s) - 1` if they are greater. If *i* or *j* are omitted or `None`, they become "end" values (which end depends on the sign of *k*). Note, *k* cannot be zero. If *k* is `None`, it is treated like `1`.

6. Concatenating immutable sequences always results in a new object. This means that building up a sequence by repeated concatenation will have a quadratic runtime cost in the total sequence length. To get a linear runtime cost, you must switch to one of the alternatives below:

   - if concatenating :class:`str` objects, you can build a list and use :meth:`str.join` at the end or else write to an :class:`io.StringIO` instance and retrieve its value when complete

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1040); *backlink*

     Unknown interpreted text role "class".

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1040); *backlink*

     Unknown interpreted text role "meth".

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1040); *backlink*

     Unknown interpreted text role "class".

- if concatenating :class:`bytes` objects, you can similarly use :meth:`bytes.join` or :class:`io.BytesIO`, or you can do in-place concatenation with a :class:`bytearray` object. :class:`bytearray` objects are mutable and have an efficient overallocation mechanism

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1044); *backlink*

  Unknown interpreted text role "class".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1044); *backlink*

  Unknown interpreted text role "meth".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1044); *backlink*

  Unknown interpreted text role "class".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1044); *backlink*

  Unknown interpreted text role "class".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1044); *backlink*

  Unknown interpreted text role "class".

- if concatenating :class:`tuple` objects, extend a :class:`list` instead

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1049); *backlink*

  Unknown interpreted text role "class".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1049); *backlink*

  Unknown interpreted text role "class".

- for other types, investigate the relevant class documentation

7. Some sequence types (such as :class:`range`) only support item sequences that follow specific patterns, and hence don't support sequence concatenation or repetition.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1055); *backlink*

   Unknown interpreted text role "class".

8. `index` raises :exc:`ValueError` when *x* is not found in *s*. Not all implementations support passing the additional arguments *i* and *j*. These arguments allow efficient searching of subsections of the sequence. Passing the extra arguments is roughly equivalent to using `s[i:j].index(x)`, only without copying any data and with the returned index being relative to the start of the sequence rather than the start of the slice.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1060); *backlink*

Unknown interpreted text role "exc".

Immutable Sequence Types

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1073)

Unknown directive type "index".

```
.. index::
   triple: immutable; sequence; types
   object: tuple
   builtin: hash
```

The only operation that immutable sequence types generally implement that is not also implemented by mutable sequence types is support for the :func:`hash` built-in.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1078); *backlink*

Unknown interpreted text role "func".

This support allows immutable sequences, such as :class:`tuple` instances, to be used as :class:`dict` keys and stored in :class:`set` and :class:`frozenset` instances.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1082); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1082); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1082); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1082); *backlink*

Unknown interpreted text role "class".

Attempting to hash an immutable sequence that contains unhashable values will result in :exc:`TypeError`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1086); *backlink*

Unknown interpreted text role "exc".

Mutable Sequence Types

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1095)

Unknown directive type "index".

```
.. index::
   triple: mutable; sequence; types
   object: list
   object: bytearray
```

The operations in the following table are defined on mutable sequence types. The :class:`collections.abc.MutableSequence` ABC is provided to make it easier to correctly implement these operations on custom sequence types.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1100); *backlink*

Unknown interpreted text role "class".

In the table *s* is an instance of a mutable sequence type, *t* is any iterable object and *x* is an arbitrary object that meets any type and value restrictions imposed by *s* (for example, :class:`bytearray` only accepts integers that meet the value restriction `0 <= x <= 255`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1104); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1110)

Unknown directive type "index".

```
.. index::
   triple: operations on; sequence; types
   triple: operations on; list; type
   pair: subscript; assignment
   pair: slice; assignment
   statement: del
   single: append() (sequence method)
   single: clear() (sequence method)
   single: copy() (sequence method)
   single: extend() (sequence method)
   single: insert() (sequence method)
   single: pop() (sequence method)
   single: remove() (sequence method)
   single: reverse() (sequence method)
```

Operation	Result	Notes
`s[i] = x`	item *i* of *s* is replaced by *x*	
`s[i:j] = t`	slice of *s* from *i* to *j* is replaced by the contents of the iterable *t*	
`del s[i:j]`	same as `s[i:j] = []`	
`s[i:j:k] = t`	the elements of `s[i:j:k]` are replaced by those of *t*	(1)
`del s[i:j:k]`	removes the elements of `s[i:j:k]` from the list	
`s.append(x)`	appends *x* to the end of the sequence (same as `s[len(s):len(s)] = [x]`)	
`s.clear()`	removes all items from *s* (same as `del s[:]`)	(5)
`s.copy()`	creates a shallow copy of *s* (same as `s[:]`)	(5)
`s.extend(t)` or `s += t`	extends *s* with the contents of *t* (for the most part the same as `s[len(s):len(s)] = t`)	
`s *= n`	updates *s* with its contents repeated *n* times	(6)
`s.insert(i, x)`	inserts *x* into *s* at the index given by *i* (same as `s[i:i] = [x]`)	
`s.pop()` or `s.pop(i)`	retrieves the item at *i* and also removes it from *s*	(2)
`s.remove(x)`	remove the first item from *s* where `s[i]` is equal to *x*	(3)
`s.reverse()`	reverses the items of *s* in place	(4)

Notes:

1. *t* must have the same length as the slice it is replacing.
2. The optional argument *i* defaults to `-1`, so that by default the last item is removed and returned.
3. :meth:`remove` raises :exc:`ValueError` when *x* is not found in *s*.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1186); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line

1186); *backlink*

Unknown interpreted text role "exc".

4. The :meth:`reverse` method modifies the sequence in place for economy of space when reversing a large sequence. To remind users that it operates by side effect, it does not return the reversed sequence.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1189); *backlink*

Unknown interpreted text role "meth".

5. :meth:`clear` and :meth:`!copy` are included for consistency with the interfaces of mutable containers that don't support slicing operations (such as :class:`dict` and :class:`set`). :meth:`!copy` is not part of the :class:`collections.abc.MutableSequence` ABC, but most concrete mutable sequence classes provide it.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1194); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1200)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
   :meth:`clear` and :meth:`!copy` methods.
```

6. The value *n* is an integer, or an object implementing :meth:`~object.__index__`. Zero and negative values of *n* clear the sequence. Items in the sequence are not copied; they are referenced multiple times, as explained for `s * n` under :ref:`typesseq-common`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1204); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1204); *backlink*

Unknown interpreted text role "ref".

Lists

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1215)

Unknown directive type "index".

```
.. index:: object: list
```

Lists are mutable sequences, typically used to store collections of homogeneous items (where the precise degree of similarity will vary by application).

Lists may be constructed in several ways:

- Using a pair of square brackets to denote the empty list: `[]`
- Using square brackets, separating items with commas: `[a]`, `[a, b, c]`
- Using a list comprehension: `[x for x in iterable]`
- Using the type constructor: `list()` or `list(iterable)`

The constructor builds a list whose items are the same and in the same order as *iterable*'s items. *iterable* may be either a sequence, a container that supports iteration, or an iterator object. If *iterable* is already a list, a copy is made and returned, similar to `iterable[:]`. For example, `list('abc')` returns `['a', 'b', 'c']` and `list( (1, 2, 3) )` returns `[1, 2, 3]`. If no argument is given, the constructor creates a new empty list, `[]`.

Many other operations also produce lists, including the :func:`sorted` built-in.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1239); *backlink*

Unknown interpreted text role "func".

Lists implement all of the :ref:`common <typesseq-common>` and :ref:`mutable <typesseq-mutable>` sequence operations. Lists also provide the following additional method:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1242); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1242); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1246)

Unknown directive type "method".

```
.. method:: list.sort(*, key=None, reverse=False)

   This method sorts the list in place, using only ``<`` comparisons
   between items. Exceptions are not suppressed - if any comparison operations
   fail, the entire sort operation will fail (and the list will likely be left
   in a partially modified state).

   :meth:`sort` accepts two arguments that can only be passed by keyword
   (:ref:`keyword-only arguments <keyword-only_parameter>`):

   *key* specifies a function of one argument that is used to extract a
   comparison key from each list element (for example, ``key=str.lower``).
```

```
The key corresponding to each item in the list is calculated once and
then used for the entire sorting process. The default value of ``None``
means that list items are sorted directly without calculating a separate
key value.

The :func:`functools.cmp_to_key` utility is available to convert a 2.x
style *cmp* function to a *key* function.

*reverse* is a boolean value.  If set to ``True``, then the list elements
are sorted as if each comparison were reversed.

This method modifies the sequence in place for economy of space when
sorting a large sequence.  To remind users that it operates by side
effect, it does not return the sorted sequence (use :func:`sorted` to
explicitly request a new sorted list instance).

The :meth:`sort` method is guaranteed to be stable.  A sort is stable if it
guarantees not to change the relative order of elements that compare equal
--- this is helpful for sorting in multiple passes (for example, sort by
department, then by salary grade).

For sorting examples and a brief sorting tutorial, see :ref:`sortinghowto`.

.. impl-detail::

   While a list is being sorted, the effect of attempting to mutate, or even
   inspect, the list is undefined.  The C implementation of Python makes the
   list appear empty for the duration, and raises :exc:`ValueError` if it can
   detect that the list has been mutated during a sort.
```

Tuples

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1294)

Unknown directive type "index".

```
.. index:: object: tuple
```

Tuples are immutable sequences, typically used to store collections of heterogeneous data (such as the 2-tuples produced by the :func:`enumerate` built-in). Tuples are also used for cases where an immutable sequence of homogeneous data is needed (such as allowing storage in a :class:`set` or :class:`dict` instance).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1296); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1296); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1296); *backlink*

Unknown interpreted text role "class".

Tuples may be constructed in a number of ways:

- Using a pair of parentheses to denote the empty tuple: `()`
- Using a trailing comma for a singleton tuple: `a,` or `(a,)`
- Separating items with commas: `a, b, c` or `(a, b, c)`
- Using the :func:`tuple` built-in: `tuple()` or `tuple(iterable)`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1309); *backlink*

Unknown interpreted text role "func".

The constructor builds a tuple whose items are the same and in the same order as *iterable*'s items. *iterable* may be either a sequence, a container that supports iteration, or an iterator object. If *iterable* is already a tuple, it is returned unchanged. For example, `tuple('abc')` returns `('a', 'b', 'c')` and `tuple( [1, 2, 3] )` returns `(1, 2, 3)`. If no argument is given, the constructor creates a new empty tuple, `()`.

Note that it is actually the comma which makes a tuple, not the parentheses. The parentheses are optional, except in the empty tuple case, or when they are needed to avoid syntactic ambiguity. For example, `f(a, b, c)` is a function call with three arguments, while `f((a, b, c))` is a function call with a 3-tuple as the sole argument.

Tuples implement all of the :ref:`common <typesseq-common>` sequence operations.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1325); *backlink***
>
> Unknown interpreted text role "ref".

For heterogeneous collections of data where access by name is clearer than access by index, :func:`collections.namedtuple` may be a more appropriate choice than a simple tuple object.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1328); *backlink***
>
> Unknown interpreted text role "func".

Ranges

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1338)**
>
> Unknown directive type "index".
>
> ```
> .. index:: object: range
> ```

The :class:`range` type represents an immutable sequence of numbers and is commonly used for looping a specific number of times in :keyword:`for` loops.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1340); *backlink***
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1340); *backlink***
>
> Unknown interpreted text role "keyword".

The arguments to the range constructor must be integers (either built-in :class:`int` or any object that implements the :meth:`~object.__index__` special method). If the *step* argument is omitted, it defaults to `1`. If the *start* argument is omitted, it defaults to `0`. If *step* is zero, :exc:`ValueError` is raised.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1347); *backlink***
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1347); *backlink***
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1347); *backlink***
>
> Unknown interpreted text role "exc".

For a positive *step*, the contents of a range `r` are determined by the formula `r[i] = start + step*i` where `i >= 0` and `r[i] < stop`.

For a negative *step*, the contents of the range are still determined by the formula `r[i] = start + step*i`, but the constraints are

`i >= 0` and `r[i] > stop`.

A range object will be empty if `r[0]` does not meet the value constraint. Ranges do support negative indices, but these are interpreted as indexing from the end of the sequence determined by the positive indices.

Ranges containing absolute values larger than :data:`sys.maxsize` are permitted but some features (such as :func:`len`) may raise :exc:`OverflowError`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1366); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1366); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1366); *backlink*

Unknown interpreted text role "exc".

Range examples:

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list(range(1, 11))
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> list(range(0, 30, 5))
[0, 5, 10, 15, 20, 25]
>>> list(range(0, 10, 3))
[0, 3, 6, 9]
>>> list(range(0, -10, -1))
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
>>> list(range(0))
[]
>>> list(range(1, 0))
[]
```

Ranges implement all of the :ref:`common <typesseq-common>` sequence operations except concatenation and repetition (due to the fact that range objects can only represent sequences that follow a strict pattern and repetition and concatenation will usually violate that pattern).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1387); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1392)

Unknown directive type "attribute".

```
.. attribute:: start

   The value of the *start* parameter (or ``0`` if the parameter was
   not supplied)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1397)

Unknown directive type "attribute".

```
.. attribute:: stop

   The value of the *stop* parameter
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1401)

Unknown directive type "attribute".

```
.. attribute:: step

   The value of the *step* parameter (or ``1`` if the parameter was
   not supplied)
```

The advantage of the :class:`range` type over a regular :class:`list` or :class:`tuple` is that a :class:`range` object will always take the same (small) amount of memory, no matter the size of the range it represents (as it only stores the `start`, `stop` and `step` values, calculating individual items and subranges as needed).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1406); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1406); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1406); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1406); *backlink*

Unknown interpreted text role "class".

Range objects implement the :class:`collections.abc.Sequence` ABC, and provide features such as containment tests, element index lookup, slicing and support for negative indices (see :ref:`typesseq`):

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1412); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1412); *backlink*

Unknown interpreted text role "ref".

```
>>> r = range(0, 20, 2)
>>> r
range(0, 20, 2)
>>> 11 in r
False
>>> 10 in r
True
>>> r.index(10)
5
>>> r[5]
10
>>> r[:5]
range(0, 10, 2)
>>> r[-1]
18
```

Testing range objects for equality with `==` and `!=` compares them as sequences. That is, two range objects are considered equal if they represent the same sequence of values. (Note that two range objects that compare equal might have different :attr:`~range.start`, :attr:`~range.stop` and :attr:`~range.step` attributes, for example `range(0) == range(2, 1, 3)` or `range(0, 3, 2) == range(0, 4, 2)`.)

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 1432); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1432); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1432); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1439)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.2
   Implement the Sequence ABC.
   Support slicing and negative indices.
   Test :class:`int` objects for membership in constant time instead of
   iterating through all items.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1445)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   Define '==' and '!=' to compare range objects based on the
   sequence of values they define (instead of comparing based on
   object identity).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1450)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
   The :attr:`~range.start`, :attr:`~range.stop` and :attr:`~range.step`
   attributes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1454)

Unknown directive type "seealso".

```
.. seealso::

   * The `linspace recipe <http://code.activestate.com/recipes/579000/>`_
     shows how to implement a lazy version of range suitable for floating
     point applications.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1460)

Unknown directive type "index".

```
.. index::
   single: string; text sequence type
   single: str (built-in class); (see also string)
   object: string
```

Text Sequence Type --- :class:`str`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1467); *backlink*

Unknown interpreted text role "class".

Textual data in Python is handled with :class:`str` objects, or :dfn:`strings`. Strings are immutable :ref:`sequences <typesseq>` of Unicode code points. String literals are written in a variety of ways:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1470); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1470); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1470); *backlink*

Unknown interpreted text role "ref".

- Single quotes: `'allows embedded "double" quotes'`
- Double quotes: `"allows embedded 'single' quotes"`
- Triple quoted: `'''Three single quotes'''`, `"""Three double quotes"""`

Triple quoted strings may span multiple lines - all associated whitespace will be included in the string literal.

String literals that are part of a single expression and have only whitespace between them will be implicitly converted to a single string literal. That is, `("spam " "eggs") == "spam eggs"`.

See :ref:`strings` for more about the various forms of string literal, including supported escape sequences, and the `r` ("raw") prefix that disables most escape sequence processing.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1486); *backlink*

Unknown interpreted text role "ref".

Strings may also be created from other objects using the :class:`str` constructor.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1490); *backlink*

Unknown interpreted text role "class".

Since there is no separate "character" type, indexing a string produces strings of length 1. That is, for a non-empty string *s*, `s[0] == s[0:1]`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1496)

Unknown directive type "index".

```
.. index::
   object: io.StringIO
```

There is also no mutable string type, but :meth:`str.join` or :class:`io.StringIO` can be used to efficiently construct strings from multiple fragments.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1499); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1499); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1503)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   For backwards compatibility with the Python 2 series, the ``u`` prefix is
```

```
once again permitted on string literals. It has no effect on the meaning
of string literals and cannot be combined with the ``r`` prefix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1509)

Unknown directive type "index".

```
.. index::
   single: string; str (built-in class)
```

Return a :ref:`string <textseq>` version of *object*. If *object* is not provided, returns the empty string. Otherwise, the behavior of `str()` depends on whether *encoding* or *errors* is given, as follows.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1515); *backlink*

Unknown interpreted text role "ref".

If neither *encoding* nor *errors* is given, `str(object)` returns :meth:`type(object).__str__(object) <object.__str__>`, which is the "informal" or nicely printable string representation of *object*. For string objects, this is the string itself. If *object* does not have a :meth:`~object.__str__` method, then :func:`str` falls back to returning :meth:`repr(object) <repr>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1519); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1519); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1519); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1519); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1527)

Unknown directive type "index".

```
.. index::
   single: buffer protocol; str (built-in class)
   single: bytes; str (built-in class)
```

If at least one of *encoding* or *errors* is given, *object* should be a :term:`bytes-like object` (e.g. :class:`bytes` or :class:`bytearray`). In this case, if *object* is a :class:`bytes` (or :class:`bytearray`) object, then `str(bytes, encoding, errors)` is equivalent to :meth:`bytes.decode(encoding, errors) <bytes.decode>`. Otherwise, the bytes object underlying the buffer object is obtained before calling :meth:`bytes.decode`. See :ref:`binaryseq` and :ref:`bufferobjects` for information on buffer objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1531); *backlink*

Unknown interpreted text role "ref".

Passing a :class:`bytes` object to :func:`str` without the *encoding* or *errors* arguments falls under the first case of returning the informal string representation (see also the :option:`-b` command-line option to Python). For example:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1540); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1540); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1540); *backlink*

Unknown interpreted text role "option".

```
>>> str(b'Zoot!')
"b'Zoot!'"
```

For more information on the `str` class and its methods, see :ref:`textseq` and the :ref:`string-methods` section below. To output formatted strings, see the :ref:`f-strings` and :ref:`formatstrings` sections. In addition, see the :ref:`stringservices` section.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1548); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1548); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1548); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1548); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1548); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1554)

Unknown directive type "index".

```
.. index::
   pair: string; methods
```

String Methods

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1562)

Unknown directive type "index".

```
.. index::
   module: re
```

Strings implement all of the :ref:`common <typesseq-common>` sequence operations, along with the additional methods described below.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1565); *backlink*

Unknown interpreted text role "ref".

Strings also support two styles of string formatting, one providing a large degree of flexibility and customization (see :meth:`str.format`, :ref:`formatstrings` and :ref:`string-formatting`) and the other based on C `printf` style formatting that handles a narrower range of types and is slightly harder to use correctly, but is often faster for the cases it can handle (:ref:`old-string-formatting`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1568); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1568); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1568); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1568); *backlink*

Unknown interpreted text role "ref".

The :ref:`textservices` section of the standard library covers a number of other modules that provide various text related utilities (including regular expression support in the :mod:`re` module).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1575); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1575); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1579)

Unknown directive type "method".

```
.. method:: str.capitalize()

   Return a copy of the string with its first character capitalized and the
   rest lowercased.

   .. versionchanged:: 3.8
      The first character is now put into titlecase rather than uppercase.
      This means that characters like digraphs will only have their first
      letter capitalized, instead of the full character.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1589)

Unknown directive type "method".

```
.. method:: str.casefold()

   Return a casefolded copy of the string. Casefolded strings may be used for
   caseless matching.

   Casefolding is similar to lowercasing but more aggressive because it is
   intended to remove all case distinctions in a string. For example, the German
   lowercase letter ``'ÃŸ'`` is equivalent to ``"ss"``. Since it is already
   lowercase, :meth:`lower` would do nothing to ``'ÃŸ'``; :meth:`casefold`
   converts it to ``"ss"``.

   The casefolding algorithm is described in section 3.13 of the Unicode
   Standard.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1606)

Unknown directive type "method".

```
.. method:: str.center(width[, fillchar])

   Return centered in a string of length *width*. Padding is done using the
   specified *fillchar* (default is an ASCII space). The original string is
   returned if *width* is less than or equal to ``len(s)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1614)

Unknown directive type "method".

```
.. method:: str.count(sub[, start[, end]])

   Return the number of non-overlapping occurrences of substring *sub* in the
```

```
range [*start*, *end*].  Optional arguments *start* and *end* are
interpreted as in slice notation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1621)

Unknown directive type "method".

```
.. method:: str.encode(encoding="utf-8", errors="strict")

   Return an encoded version of the string as a bytes object. Default encoding
   is ``'utf-8'``. *errors* may be given to set a different error handling scheme.
   The default for *errors* is ``'strict'``, meaning that encoding errors raise
   a :exc:`UnicodeError`. Other possible
   values are ``'ignore'``, ``'replace'``, ``'xmlcharrefreplace'``,
   ``'backslashreplace'`` and any other name registered via
   :func:`codecs.register_error`, see section :ref:`error-handlers`. For a
   list of possible encodings, see section :ref:`standard-encodings`.

   By default, the *errors* argument is not checked for best performances, but
   only used at the first encoding error. Enable the :ref:`Python Development
   Mode <devmode>`, or use a :ref:`debug build <debug-build>` to check
   *errors*.

   .. versionchanged:: 3.1
      Support for keyword arguments added.

   .. versionchanged:: 3.9
      The *errors* is now checked in development mode and
      in :ref:`debug mode <debug-build>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1645)

Unknown directive type "method".

```
.. method:: str.endswith(suffix[, start[, end]])

   Return ``True`` if the string ends with the specified *suffix*, otherwise return
   ``False``.  *suffix* can also be a tuple of suffixes to look for.  With optional
   *start*, test beginning at that position.  With optional *end*, stop comparing
   at that position.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1653)

Unknown directive type "method".

```
.. method:: str.expandtabs(tabsize=8)

   Return a copy of the string where all tab characters are replaced by one or
   more spaces, depending on the current column and the given tab size.  Tab
   positions occur every *tabsize* characters (default is 8, giving tab
   positions at columns 0, 8, 16 and so on).  To expand the string, the current
   column is set to zero and the string is examined character by character.  If
   the character is a tab (``\t``), one or more space characters are inserted
   in the result until the current column is equal to the next tab position.
   (The tab character itself is not copied.)  If the character is a newline
   (``\n``) or return (``\r``), it is copied and the current column is reset to
   zero.  Any other character is copied unchanged and the current column is
   incremented by one regardless of how the character is represented when
   printed.

      >>> '01\t012\t0123\t01234'.expandtabs()
      '01      012     0123    01234'
      >>> '01\t012\t0123\t01234'.expandtabs(4)
      '01  012 0123    01234'
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1674)

Unknown directive type "method".

```
.. method:: str.find(sub[, start[, end]])

   Return the lowest index in the string where substring *sub* is found within
   the slice ``s[start:end]``.  Optional arguments *start* and *end* are
   interpreted as in slice notation.  Return ``-1`` if *sub* is not found.

   .. note::

      The :meth:`~str.find` method should be used only if you need to know the
      position of *sub*.  To check if *sub* is a substring or not, use the
      :keyword:`in` operator::

         >>> 'Py' in 'Python'
         True
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1690)

Unknown directive type "method".

```
.. method:: str.format(*args, **kwargs)

   Perform a string formatting operation.  The string on which this method is
   called can contain literal text or replacement fields delimited by braces
   ``{}``.  Each replacement field contains either the numeric index of a
   positional argument, or the name of a keyword argument.  Returns a copy of
   the string where each replacement field is replaced with the string value of
   the corresponding argument.

      >>> "The sum of 1 + 2 is {0}".format(1+2)
      'The sum of 1 + 2 is 3'

   See :ref:`formatstrings` for a description of the various formatting options
   that can be specified in format strings.

   .. note::
      When formatting a number (:class:`int`, :class:`float`, :class:`complex`,
      :class:`decimal.Decimal` and subclasses) with the ``n`` type
      (ex: ``'{:n}'.format(1234)``), the function temporarily sets the
      ``LC_CTYPE`` locale to the ``LC_NUMERIC`` locale to decode
      ``decimal_point`` and ``thousands_sep`` fields of :c:func:`localeconv` if
      they are non-ASCII or longer than 1 byte, and the ``LC_NUMERIC`` locale is
      different than the ``LC_CTYPE`` locale.  This temporary change affects
      other threads.

   .. versionchanged:: 3.7
      When formatting a number with the ``n`` type, the function sets
      temporarily the ``LC_CTYPE`` locale to the ``LC_NUMERIC`` locale in some
      cases.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1721)

Unknown directive type "method".

```
.. method:: str.format_map(mapping)

   Similar to ``str.format(**mapping)``, except that ``mapping`` is
   used directly and not copied to a :class:`dict`.  This is useful
   if for example ``mapping`` is a dict subclass:

   >>> class Default(dict):
   ...     def __missing__(self, key):
   ...         return key
   ...
   >>> '{name} was born in {country}'.format_map(Default(name='Guido'))
   'Guido was born in country'

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1737)

Unknown directive type "method".

```
.. method:: str.index(sub[, start[, end]])

   Like :meth:`~str.find`, but raise :exc:`ValueError` when the substring is
   not found.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1743)

Unknown directive type "method".

```
.. method:: str.isalnum()

   Return ``True`` if all characters in the string are alphanumeric and there is at
   least one character, ``False`` otherwise.  A character ``c`` is alphanumeric if one
   of the following returns ``True``: ``c.isalpha()``, ``c.isdecimal()``,
   ``c.isdigit()``, or ``c.isnumeric()``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1751)

Unknown directive type "method".

```
.. method:: str.isalpha()

   Return ``True`` if all characters in the string are alphabetic and there is at least
   one character, ``False`` otherwise.  Alphabetic characters are those characters defined
   in the Unicode character database as "Letter", i.e., those with general category
   property being one of "Lm", "Lt", "Lu", "Ll", or "Lo".  Note that this is different
   from the "Alphabetic" property defined in the Unicode Standard.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1760)

Unknown directive type "method".

```
.. method:: str.isascii()

   Return ``True`` if the string is empty or all characters in the string are ASCII,
   ``False`` otherwise.
   ASCII characters have code points in the range U+0000-U+007F.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1769)

Unknown directive type "method".

```
.. method:: str.isdecimal()

   Return ``True`` if all characters in the string are decimal
   characters and there is at least one character, ``False``
   otherwise. Decimal characters are those that can be used to form
   numbers in base 10, e.g. U+0660, ARABIC-INDIC DIGIT
   ZERO.  Formally a decimal character is a character in the Unicode
   General Category "Nd".
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1779)

Unknown directive type "method".

```
.. method:: str.isdigit()

   Return ``True`` if all characters in the string are digits and there is at least one
   character, ``False`` otherwise.  Digits include decimal characters and digits that need
   special handling, such as the compatibility superscript digits.
   This covers digits which cannot be used to form numbers in base 10,
   like the Kharosthi numbers.  Formally, a digit is a character that has the
```

```
property value Numeric_Type=Digit or Numeric_Type=Decimal.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1789)

Unknown directive type "method".

```
.. method:: str.isidentifier()

   Return ``True`` if the string is a valid identifier according to the language
   definition, section :ref:`identifiers`.

   Call :func:`keyword.iskeyword` to test whether string ``s`` is a reserved
   identifier, such as :keyword:`def` and :keyword:`class`.

   Example:
   ::

      >>> from keyword import iskeyword

      >>> 'hello'.isidentifier(), iskeyword('hello')
      (True, False)
      >>> 'def'.isidentifier(), iskeyword('def')
      (True, True)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1808)

Unknown directive type "method".

```
.. method:: str.islower()

   Return ``True`` if all cased characters [4]_ in the string are lowercase and
   there is at least one cased character, ``False`` otherwise.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1814)

Unknown directive type "method".

```
.. method:: str.isnumeric()

   Return ``True`` if all characters in the string are numeric
   characters, and there is at least one character, ``False``
   otherwise. Numeric characters include digit characters, and all characters
   that have the Unicode numeric value property, e.g. U+2155,
   VULGAR FRACTION ONE FIFTH.  Formally, numeric characters are those with the property
   value Numeric_Type=Digit, Numeric_Type=Decimal or Numeric_Type=Numeric.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1824)

Unknown directive type "method".

```
.. method:: str.isprintable()

   Return ``True`` if all characters in the string are printable or the string is
   empty, ``False`` otherwise.  Nonprintable characters are those characters defined
   in the Unicode character database as "Other" or "Separator", excepting the
   ASCII space (0x20) which is considered printable.  (Note that printable
   characters in this context are those which should not be escaped when
   :func:`repr` is invoked on a string.  It has no bearing on the handling of
   strings written to :data:`sys.stdout` or :data:`sys.stderr`.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1835)

Unknown directive type "method".

```
.. method:: str.isspace()

   Return ``True`` if there are only whitespace characters in the string and there is
   at least one character, ``False`` otherwise.

   A character is *whitespace* if in the Unicode character database
   (see :mod:`unicodedata`), either its general category is ``Zs``
   ("Separator, space"), or its bidirectional class is one of ``WS``,
   ``B``, or ``S``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1846)

Unknown directive type "method".

```
.. method:: str.istitle()

   Return ``True`` if the string is a titlecased string and there is at least one
   character, for example uppercase characters may only follow uncased characters
   and lowercase characters only cased ones.  Return ``False`` otherwise.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1853)

Unknown directive type "method".

```
.. method:: str.isupper()

   Return ``True`` if all cased characters [4]_ in the string are uppercase and
   there is at least one cased character, ``False`` otherwise.

      >>> 'BANANA'.isupper()
      True
      >>> 'banana'.isupper()
      False
      >>> 'baNana'.isupper()
      False
      >>> ' '.isupper()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1869)

Unknown directive type "method".

```
.. method:: str.join(iterable)

   Return a string which is the concatenation of the strings in *iterable*.
   A :exc:`TypeError` will be raised if there are any non-string values in
   *iterable*, including :class:`bytes` objects.  The separator between
   elements is the string providing this method.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1877)

Unknown directive type "method".

```
.. method:: str.ljust(width[, fillchar])

   Return the string left justified in a string of length *width*. Padding is
   done using the specified *fillchar* (default is an ASCII space). The
   original string is returned if *width* is less than or equal to ``len(s)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1884)

Unknown directive type "method".

```
.. method:: str.lower()

   Return a copy of the string with all the cased characters [4]_ converted to
   lowercase.

   The lowercasing algorithm used is described in section 3.13 of the Unicode
   Standard.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1893)

Unknown directive type "method".

```
.. method:: str.lstrip([chars])

   Return a copy of the string with leading characters removed.  The *chars*
   argument is a string specifying the set of characters to be removed.  If omitted
   or ``None``, the *chars* argument defaults to removing whitespace.  The *chars*
   argument is not a prefix; rather, all combinations of its values are stripped::

      >>> '   spacious   '.lstrip()
      'spacious   '
      >>> 'www.example.com'.lstrip('cmowz.')
      'example.com'

   See :meth:`str.removeprefix` for a method that will remove a single prefix
   string rather than all of a set of characters.  For example::

      >>> 'Arthur: three!'.lstrip('Arthur: ')
      'ee!'
      >>> 'Arthur: three!'.removeprefix('Arthur: ')
      'three!'
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1914)

Unknown directive type "staticmethod".

```
.. staticmethod:: str.maketrans(x[, y[, z]])

   This static method returns a translation table usable for :meth:`str.translate`.

   If there is only one argument, it must be a dictionary mapping Unicode
   ordinals (integers) or characters (strings of length 1) to Unicode ordinals,
   strings (of arbitrary lengths) or ``None``.  Character keys will then be
   converted to ordinals.

   If there are two arguments, they must be strings of equal length, and in the
   resulting dictionary, each character in x will be mapped to the character at
   the same position in y.  If there is a third argument, it must be a string,
   whose characters will be mapped to ``None`` in the result.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1929)

Unknown directive type "method".

```
.. method:: str.partition(sep)

   Split the string at the first occurrence of *sep*, and return a 3-tuple
   containing the part before the separator, the separator itself, and the part
   after the separator.  If the separator is not found, return a 3-tuple containing
   the string itself, followed by two empty strings.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1937)

Unknown directive type "method".

```
.. method:: str.removeprefix(prefix, /)

   If the string starts with the *prefix* string, return
```

```
``string[len(prefix):]``. Otherwise, return a copy of the original
string::

   >>> 'TestHook'.removeprefix('Test')
   'Hook'
   >>> 'BaseTestCase'.removeprefix('Test')
   'BaseTestCase'

.. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1951)

Unknown directive type "method".

```
.. method:: str.removesuffix(suffix, /)

   If the string ends with the *suffix* string and that *suffix* is not empty,
   return ``string[:-len(suffix)]``. Otherwise, return a copy of the
   original string::

      >>> 'MiscTests'.removesuffix('Tests')
      'Misc'
      >>> 'TmpDirMixin'.removesuffix('Tests')
      'TmpDirMixin'

   .. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1965)

Unknown directive type "method".

```
.. method:: str.replace(old, new[, count])

   Return a copy of the string with all occurrences of substring *old* replaced by
   *new*.  If the optional argument *count* is given, only the first *count*
   occurrences are replaced.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1972)

Unknown directive type "method".

```
.. method:: str.rfind(sub[, start[, end]])

   Return the highest index in the string where substring *sub* is found, such
   that *sub* is contained within ``s[start:end]``.  Optional arguments *start*
   and *end* are interpreted as in slice notation.  Return ``-1`` on failure.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1979)

Unknown directive type "method".

```
.. method:: str.rindex(sub[, start[, end]])

   Like :meth:`rfind` but raises :exc:`ValueError` when the substring *sub* is not
   found.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1985)

Unknown directive type "method".

```
.. method:: str.rjust(width[, fillchar])

   Return the string right justified in a string of length *width*. Padding is
   done using the specified *fillchar* (default is an ASCII space). The
```

```
   original string is returned if *width* is less than or equal to ``len(s)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 1992)

Unknown directive type "method".

```
.. method:: str.rpartition(sep)

   Split the string at the last occurrence of *sep*, and return a 3-tuple
   containing the part before the separator, the separator itself, and the part
   after the separator.  If the separator is not found, return a 3-tuple containing
   two empty strings, followed by the string itself.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2000)

Unknown directive type "method".

```
.. method:: str.rsplit(sep=None, maxsplit=-1)

   Return a list of the words in the string, using *sep* as the delimiter string.
   If *maxsplit* is given, at most *maxsplit* splits are done, the *rightmost*
   ones.  If *sep* is not specified or ``None``, any whitespace string is a
   separator.  Except for splitting from the right, :meth:`rsplit` behaves like
   :meth:`split` which is described in detail below.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2009)

Unknown directive type "method".

```
.. method:: str.rstrip([chars])

   Return a copy of the string with trailing characters removed.  The *chars*
   argument is a string specifying the set of characters to be removed.  If omitted
   or ``None``, the *chars* argument defaults to removing whitespace.  The *chars*
   argument is not a suffix; rather, all combinations of its values are stripped::

      >>> '   spacious   '.rstrip()
      '   spacious'
      >>> 'mississippi'.rstrip('ipz')
      'mississ'

   See :meth:`str.removesuffix` for a method that will remove a single suffix
   string rather than all of a set of characters.  For example::

      >>> 'Monty Python'.rstrip(' Python')
      'M'
      >>> 'Monty Python'.removesuffix(' Python')
      'Monty'
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2029)

Unknown directive type "method".

```
.. method:: str.split(sep=None, maxsplit=-1)

   Return a list of the words in the string, using *sep* as the delimiter
   string.  If *maxsplit* is given, at most *maxsplit* splits are done (thus,
   the list will have at most ``maxsplit+1`` elements).  If *maxsplit* is not
   specified or ``-1``, then there is no limit on the number of splits
   (all possible splits are made).

   If *sep* is given, consecutive delimiters are not grouped together and are
   deemed to delimit empty strings (for example, ``'1,,2'.split(',')`` returns
   ``['1', '', '2']``).  The *sep* argument may consist of multiple characters
   (for example, ``'1<>2<>3'.split('<>')`` returns ``['1', '2', '3']``).
   Splitting an empty string with a specified separator returns ``['']``.

   For example::
```

```
   >>> '1,2,3'.split(',')
   ['1', '2', '3']
   >>> '1,2,3'.split(',', maxsplit=1)
   ['1', '2,3']
   >>> '1,2,,3,'.split(',')
   ['1', '2', '', '3', '']

If *sep* is not specified or is ``None``, a different splitting algorithm is
applied: runs of consecutive whitespace are regarded as a single separator,
and the result will contain no empty strings at the start or end if the
string has leading or trailing whitespace.  Consequently, splitting an empty
string or a string consisting of just whitespace with a ``None`` separator
returns ``[]``.

For example::

   >>> '1 2 3'.split()
   ['1', '2', '3']
   >>> '1 2 3'.split(maxsplit=1)
   ['1', '2 3']
   >>> '   1   2   3   '.split()
   ['1', '2', '3']
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2069)

Unknown directive type "index".

```
.. index::
   single: universal newlines; str.splitlines method
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2072)

Unknown directive type "method".

```
.. method:: str.splitlines(keepends=False)

   Return a list of the lines in the string, breaking at line boundaries.  Line
   breaks are not included in the resulting list unless *keepends* is given and
   true.

   This method splits on the following line boundaries.  In particular, the
   boundaries are a superset of :term:`universal newlines`.

   +-----------------------+-----------------------------+
   | Representation        | Description                 |
   +=======================+=============================+
   | ``\n``                | Line Feed                   |
   +-----------------------+-----------------------------+
   | ``\r``                | Carriage Return             |
   +-----------------------+-----------------------------+
   | ``\r\n``              | Carriage Return + Line Feed |
   +-----------------------+-----------------------------+
   | ``\v`` or ``\x0b``    | Line Tabulation             |
   +-----------------------+-----------------------------+
   | ``\f`` or ``\x0c``    | Form Feed                   |
   +-----------------------+-----------------------------+
   | ``\x1c``              | File Separator              |
   +-----------------------+-----------------------------+
   | ``\x1d``              | Group Separator             |
   +-----------------------+-----------------------------+
   | ``\x1e``              | Record Separator            |
   +-----------------------+-----------------------------+
   | ``\x85``              | Next Line (C1 Control Code) |
   +-----------------------+-----------------------------+
   | ``\u2028``            | Line Separator              |
   +-----------------------+-----------------------------+
   | ``\u2029``            | Paragraph Separator         |
   +-----------------------+-----------------------------+

   .. versionchanged:: 3.2

      ``\v`` and ``\f`` added to list of line boundaries.

   For example::

      >>> 'ab c\n\nde fg\rkl\r\n'.splitlines()
      ['ab c', '', 'de fg', 'kl']
```

```
   >>> 'ab c\n\nde fg\rkl\r\n'.splitlines(keepends=True)
   ['ab c\n', '\n', 'de fg\r', 'kl\r\n']

Unlike :meth:`~str.split` when a delimiter string *sep* is given, this
method returns an empty list for the empty string, and a terminal line
break does not result in an extra line::

   >>> "".splitlines()
   []
   >>> "One line\n".splitlines()
   ['One line']

For comparison, ``split('\n')`` gives::

   >>> ''.split('\n')
   ['']
   >>> 'Two lines\n'.split('\n')
   ['Two lines', '']
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2135)

Unknown directive type "method".

```
.. method:: str.startswith(prefix[, start[, end]])

   Return ``True`` if string starts with the *prefix*, otherwise return ``False``.
   *prefix* can also be a tuple of prefixes to look for.  With optional *start*,
   test string beginning at that position.  With optional *end*, stop comparing
   string at that position.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2143)

Unknown directive type "method".

```
.. method:: str.strip([chars])

   Return a copy of the string with the leading and trailing characters removed.
   The *chars* argument is a string specifying the set of characters to be removed.
   If omitted or ``None``, the *chars* argument defaults to removing whitespace.
   The *chars* argument is not a prefix or suffix; rather, all combinations of its
   values are stripped::

      >>> '   spacious   '.strip()
      'spacious'
      >>> 'www.example.com'.strip('cmowz.')
      'example'

   The outermost leading and trailing *chars* argument values are stripped
   from the string. Characters are removed from the leading end until
   reaching a string character that is not contained in the set of
   characters in *chars*. A similar action takes place on the trailing end.
   For example::

      >>> comment_string = '#....... Section 3.2.1 Issue #32 .......'
      >>> comment_string.strip('.#! ')
      'Section 3.2.1 Issue #32'
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2167)

Unknown directive type "method".

```
.. method:: str.swapcase()

   Return a copy of the string with uppercase characters converted to lowercase and
   vice versa. Note that it is not necessarily true that
   ``s.swapcase().swapcase() == s``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2174)

Unknown directive type "method".

```
.. method:: str.title()

   Return a titlecased version of the string where words start with an uppercase
   character and the remaining characters are lowercase.

   For example::

      >>> 'Hello world'.title()
      'Hello World'

   The algorithm uses a simple language-independent definition of a word as
   groups of consecutive letters.  The definition works in many contexts but
   it means that apostrophes in contractions and possessives form word
   boundaries, which may not be the desired result::

        >>> "they're bill's friends from the UK".title()
        "They'Re Bill'S Friends From The Uk"

   The :func:`string.capwords` function does not have this problem, as it
   splits words on spaces only.

   Alternatively, a workaround for apostrophes can be constructed using regular
   expressions::

        >>> import re
        >>> def titlecase(s):
        ...     return re.sub(r"[A-Za-z]+('[A-Za-z]+)?",
        ...                   lambda mo: mo.group(0).capitalize(),
        ...                   s)
        ...
        >>> titlecase("they're bill's friends.")
        "They're Bill's Friends."
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2208)

Unknown directive type "method".

```
.. method:: str.translate(table)

   Return a copy of the string in which each character has been mapped through
   the given translation table.  The table must be an object that implements
   indexing via :meth:`__getitem__`, typically a :term:`mapping` or
   :term:`sequence`.  When indexed by a Unicode ordinal (an integer), the
   table object can do any of the following: return a Unicode ordinal or a
   string, to map the character to one or more other characters; return
   ``None``, to delete the character from the return string; or raise a
   :exc:`LookupError` exception, to map the character to itself.

   You can use :meth:`str.maketrans` to create a translation map from
   character-to-character mappings in different formats.

   See also the :mod:`codecs` module for a more flexible approach to custom
   character mappings.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2226)

Unknown directive type "method".

```
.. method:: str.upper()

   Return a copy of the string with all the cased characters [4]_ converted to
   uppercase.  Note that ``s.upper().isupper()`` might be ``False`` if ``s``
   contains uncased characters or if the Unicode category of the resulting
   character(s) is not "Lu" (Letter, uppercase), but e.g. "Lt" (Letter,
   titlecase).

   The uppercasing algorithm used is described in section 3.13 of the Unicode
   Standard.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2238)

Unknown directive type "method".

```
.. method:: str.zfill(width)

   Return a copy of the string left filled with ASCII ``'0'`` digits to
   make a string of length *width*. A leading sign prefix (``'+'``/``'-'``)
   is handled by inserting the padding *after* the sign character rather
   than before. The original string is returned if *width* is less than
   or equal to ``len(s)``.

   For example::

      >>> "42".zfill(5)
      '00042'
      >>> "-42".zfill(5)
      '-0042'
```

`printf`-style String Formatting

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2260)

Unknown directive type "index".

```
.. index::
   single: formatting, string (%)
   single: interpolation, string (%)
   single: string; formatting, printf
   single: string; interpolation, printf
   single: printf-style formatting
   single: sprintf-style formatting
   single: % (percent); printf-style formatting
```

Note

The formatting operations described here exhibit a variety of quirks that lead to a number of common errors (such as failing to display tuples and dictionaries correctly). Using the newer :ref:`formatted string literals <f-strings>`, the :meth:`str.format` interface, or :ref:`template strings <template-strings>` may help avoid these errors. Each of these alternatives provides their own trade-offs and benefits of simplicity, flexibility, and/or extensibility.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2271); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2271); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2271); *backlink*

Unknown interpreted text role "ref".

String objects have one unique built-in operation: the `%` operator (modulo). This is also known as the string *formatting* or *interpolation* operator. Given `format % values` (where *format* is a string), `%` conversion specifications in *format* are replaced with zero or more elements of *values*. The effect is similar to using the :c:func:`sprintf` in the C language.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2279); *backlink*

Unknown interpreted text role "c:func".

If *format* requires a single argument, *values* may be a single non-tuple object. [5] Otherwise, *values* must be a tuple with exactly the number of items specified by the format string, or a single mapping object (for example, a dictionary).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2290)

Unknown directive type "index".

```
.. index::
   single: () (parentheses); in printf-style formatting
   single: * (asterisk); in printf-style formatting
   single: . (dot); in printf-style formatting
```

A conversion specifier contains two or more characters and has the following components, which must occur in this order:

1. The `'%'` character, which marks the start of the specifier.
2. Mapping key (optional), consisting of a parenthesised sequence of characters (for example, `(somename)`).
3. Conversion flags (optional), which affect the result of some conversion types.
4. Minimum field width (optional). If specified as an `'*'` (asterisk), the actual width is read from the next element of the tuple in *values*, and the object to convert comes after the minimum field width and optional precision.
5. Precision (optional), given as a `'.'` (dot) followed by the precision. If specified as `'*'` (an asterisk), the actual precision is read from the next element of the tuple in *values*, and the value to convert comes after the precision.
6. Length modifier (optional).
7. Conversion type.

When the right argument is a dictionary (or other mapping type), then the formats in the string *must* include a parenthesised mapping key into that dictionary inserted immediately after the `'%'` character. The mapping key selects the value to be formatted from the mapping. For example:

```
>>> print('%(language)s has %(number)03d quote types.' %
...       {'language': "Python", "number": 2})
Python has 002 quote types.
```

In this case no `*` specifiers may occur in a format (since they require a sequential parameter list).

The conversion flag characters are:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2333)

Unknown directive type "index".

```
.. index::
   single: # (hash); in printf-style formatting
   single: - (minus); in printf-style formatting
   single: + (plus); in printf-style formatting
   single: space; in printf-style formatting
```

Flag	Meaning
`'#'`	The value conversion will use the "alternate form" (where defined below).
`'0'`	The conversion will be zero padded for numeric values.
`'-'`	The converted value is left adjusted (overrides the `'0'` conversion if both are given).
`' '`	(a space) A blank should be left before a positive number (or empty string) produced by a signed conversion.
`'+'`	A sign character (`'+'` or `'-'`) will precede the conversion (overrides a "space" flag).

A length modifier (`h`, `l`, or `L`) may be present, but is ignored as it is not necessary for Python -- so e.g. `%ld` is identical to `%d`.

The conversion types are:

Conversion	Meaning	Notes
`'d'`	Signed integer decimal.	
`'i'`	Signed integer decimal.	
`'o'`	Signed octal value.	(1)
`'u'`	Obsolete type -- it is identical to `'d'`.	(6)
`'x'`	Signed hexadecimal (lowercase).	(2)
`'X'`	Signed hexadecimal (uppercase).	(2)
`'e'`	Floating point exponential format (lowercase).	(3)
`'E'`	Floating point exponential format (uppercase).	(3)

Conversion	Meaning	Notes
`'f'`	Floating point decimal format.	(3)
`'F'`	Floating point decimal format.	(3)
`'g'`	Floating point format. Uses lowercase exponential format if exponent is less than -4 or not less than precision, decimal format otherwise.	(4)
`'G'`	Floating point format. Uses uppercase exponential format if exponent is less than -4 or not less than precision, decimal format otherwise.	(4)
`'c'`	Single character (accepts integer or single character string).	
`'r'`	String (converts any Python object using :func:`repr`). **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2397); *backlink*** Unknown interpreted text role "func".	(5)
`'s'`	String (converts any Python object using :func:`str`). **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2400); *backlink*** Unknown interpreted text role "func".	(5)
`'a'`	String (converts any Python object using :func:`ascii`). **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2403); *backlink*** Unknown interpreted text role "func".	(5)
`'%'`	No argument is converted, results in a `'%'` character in the result.	

Notes:

1. The alternate form causes a leading octal specifier (`'0o'`) to be inserted before the first digit.
2. The alternate form causes a leading `'0x'` or `'0X'` (depending on whether the `'x'` or `'X'` format was used) to be inserted before the first digit.
3. The alternate form causes the result to always contain a decimal point, even if no digits follow it.

   The precision determines the number of digits after the decimal point and defaults to 6.
4. The alternate form causes the result to always contain a decimal point, and trailing zeroes are not removed as they would otherwise be.

   The precision determines the number of significant digits before and after the decimal point and defaults to 6.
5. If precision is `N`, the output is truncated to `N` characters.
6. See PEP 237.

Since Python strings have an explicit length, `%s` conversions do not assume that `'\0'` is the end of the string.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2444)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.1
   ``%f`` conversions for numbers whose absolute value is over 1e50 are no
   longer replaced by ``%g`` conversions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2449)

Unknown directive type "index".

```
.. index::
   single: buffer protocol; binary sequence types
```

Binary Sequence Types --- :class:`bytes`, :class:`bytearray`, :class:`memoryview`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2454); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2454); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2454); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2457)

Unknown directive type "index".

```
.. index::
   object: bytes
   object: bytearray
   object: memoryview
   module: array
```

The core built-in types for manipulating binary data are :class:`bytes` and :class:`bytearray`. They are supported by :class:`memoryview` which uses the :ref:`buffer protocol <bufferobjects>` to access the memory of other binary objects without needing to make a copy.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2463); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2463); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2463); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2463); *backlink*

Unknown interpreted text role "ref".

The :mod:`array` module supports efficient storage of basic data types like 32-bit integers and IEEE754 double-precision floating values.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2468); *backlink*

Unknown interpreted text role "mod".

Bytes Objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2476)

Unknown directive type "index".

```
.. index:: object: bytes
```

Bytes objects are immutable sequences of single bytes. Since many major binary protocols are based on the ASCII text encoding, bytes objects offer several methods that are only valid when working with ASCII compatible data and are closely related to string objects in a variety of other ways.

Firstly, the syntax for bytes literals is largely the same as that for string literals, except that a `b` prefix is added:

- Single quotes: `b'still allows embedded "double" quotes'`
- Double quotes: `b"still allows embedded 'single' quotes"`
- Triple quoted: `b'''3 single quotes'''`, `b"""3 double quotes"""`

Only ASCII characters are permitted in bytes literals (regardless of the declared source code encoding). Any binary values over 127 must be entered into bytes literals using the appropriate escape sequence.

As with string literals, bytes literals may also use a `r` prefix to disable processing of escape sequences. See :ref:`strings` for more about the various forms of bytes literal, including supported escape sequences.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2496); *backlink*

Unknown interpreted text role "ref".

While bytes literals and representations are based on ASCII text, bytes objects actually behave like immutable sequences of integers, with each value in the sequence restricted such that `0 <= x < 256` (attempts to violate this restriction will trigger :exc:`ValueError`). This is done deliberately to emphasise that while many binary formats include ASCII based elements and can be usefully manipulated with some text-oriented algorithms, this is not generally the case for arbitrary binary data (blindly applying text processing algorithms to binary data formats that are not ASCII compatible will usually lead to data corruption).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2500); *backlink*

Unknown interpreted text role "exc".

In addition to the literal forms, bytes objects can be created in a number of other ways:

- A zero-filled bytes object of a specified length: `bytes(10)`
- From an iterable of integers: `bytes(range(20))`
- Copying existing binary data via the buffer protocol: `bytes(obj)`

Also see the :ref:`bytes <func-bytes>` built-in.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2517); *backlink*

Unknown interpreted text role "ref".

Since 2 hexadecimal digits correspond precisely to a single byte, hexadecimal numbers are a commonly used format for describing binary data. Accordingly, the bytes type has an additional class method to read data in that format:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2523)

Unknown directive type "classmethod".

```
.. classmethod:: fromhex(string)

   This :class:`bytes` class method returns a bytes object, decoding the
   given string object.  The string must contain two hexadecimal digits per
   byte, with ASCII whitespace being ignored.

   >>> bytes.fromhex('2Ef0 F1f2  ')
   b'.\xf0\xf1\xf2'

   .. versionchanged:: 3.7
      :meth:`bytes.fromhex` now skips all ASCII whitespace in the string,
      not just spaces.
```

A reverse conversion function exists to transform a bytes object into its hexadecimal representation.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2539)

Unknown directive type "method".

```
.. method:: hex([sep[, bytes_per_sep]])

   Return a string object containing two hexadecimal digits for each
   byte in the instance.

   >>> b'\xf0\xf1\xf2'.hex()
   'f0f1f2'

   If you want to make the hex string easier to read, you can specify a
   single character separator *sep* parameter to include in the output.
   By default between each byte.  A second optional *bytes_per_sep*
   parameter controls the spacing.  Positive values calculate the
   separator position from the right, negative values from the left.

   >>> value = b'\xf0\xf1\xf2'
   >>> value.hex('-')
   'f0-f1-f2'
   >>> value.hex('_', 2)
   'f0_f1f2'
   >>> b'UUDDLRLRAB'.hex(' ', -4)
   '55554444 4c524c52 4142'

   .. versionadded:: 3.5

   .. versionchanged:: 3.8
      :meth:`bytes.hex` now supports optional *sep* and *bytes_per_sep*
      parameters to insert separators between bytes in the hex output.
```

Since bytes objects are sequences of integers (akin to a tuple), for a bytes object *b*, `b[0]` will be an integer, while `b[0:1]` will be a bytes object of length 1. (This contrasts with text strings, where both indexing and slicing will produce a string of length 1)

The representation of bytes objects uses the literal format (`b'...'`) since it is often more useful than e.g. `bytes([46, 46, 46])`. You can always convert a bytes object into a list of integers using `list(b)`.

Note

For Python 2.x users: In the Python 2.x series, a variety of implicit conversions between 8-bit strings (the closest thing 2.x offers to a built-in binary data type) and Unicode strings were permitted. This was a backwards compatibility workaround to account for the fact that Python originally only supported 8-bit text, and Unicode text was a later addition. In Python 3.x, those implicit conversions are gone - conversions between 8-bit binary data and Unicode text must be explicit, and bytes and string objects will always compare unequal.

Bytearray Objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2592)

Unknown directive type "index".

```
.. index:: object: bytearray
```

:class:`bytearray` objects are a mutable counterpart to :class:`bytes` objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2594); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2594); *backlink*

Unknown interpreted text role "class".

There is no dedicated literal syntax for bytearray objects, instead they are always created by calling the constructor:

- Creating an empty instance: `bytearray()`
- Creating a zero-filled instance with a given length: `bytearray(10)`
- From an iterable of integers: `bytearray(range(20))`

- Copying existing binary data via the buffer protocol: `bytearray(b'Hi!')`

As bytearray objects are mutable, they support the :ref:`mutable <typesseq-mutable>` sequence operations in addition to the common bytes and bytearray operations described in :ref:`bytes-methods`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2607); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2607); *backlink***
>
> Unknown interpreted text role "ref".

Also see the :ref:`bytearray <func-bytearray>` built-in.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2611); *backlink***
>
> Unknown interpreted text role "ref".

Since 2 hexadecimal digits correspond precisely to a single byte, hexadecimal numbers are a commonly used format for describing binary data. Accordingly, the bytearray type has an additional class method to read data in that format:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2617)**
>
> Unknown directive type "classmethod".
>
> ```
> .. classmethod:: fromhex(string)
>
>    This :class:`bytearray` class method returns bytearray object, decoding
>    the given string object.  The string must contain two hexadecimal digits
>    per byte, with ASCII whitespace being ignored.
>
>    >>> bytearray.fromhex('2Ef0 F1f2  ')
>    bytearray(b'.\xf0\xf1\xf2')
>
>    .. versionchanged:: 3.7
>       :meth:`bytearray.fromhex` now skips all ASCII whitespace in the string,
>       not just spaces.
> ```

A reverse conversion function exists to transform a bytearray object into its hexadecimal representation.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2633)**
>
> Unknown directive type "method".
>
> ```
> .. method:: hex([sep[, bytes_per_sep]])
>
>    Return a string object containing two hexadecimal digits for each
>    byte in the instance.
>
>    >>> bytearray(b'\xf0\xf1\xf2').hex()
>    'f0f1f2'
>
>    .. versionadded:: 3.5
>
>    .. versionchanged:: 3.8
>       Similar to :meth:`bytes.hex`, :meth:`bytearray.hex` now supports
>       optional *sep* and *bytes_per_sep* parameters to insert separators
>       between bytes in the hex output.
> ```

Since bytearray objects are sequences of integers (akin to a list), for a bytearray object *b*, `b[0]` will be an integer, while `b[0:1]` will be a bytearray object of length 1. (This contrasts with text strings, where both indexing and slicing will produce a string of length 1)

The representation of bytearray objects uses the bytes literal format (`bytearray(b'...')`) since it is often more useful than e.g. `bytearray([46, 46, 46])`. You can always convert a bytearray object into a list of integers using `list(b)`.

Bytes and Bytearray Operations

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2664)

Unknown directive type "index".

```
.. index:: pair: bytes; methods
           pair: bytearray; methods
```

Both bytes and bytearray objects support the :ref:`common <typesseq-common>` sequence operations. They interoperate not just with operands of the same type, but with any :term:`bytes-like object`. Due to this flexibility, they can be freely mixed in operations without causing errors. However, the return type of the result may depend on the order of operands.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2667); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2667); *backlink*

Unknown interpreted text role "term".

Note

The methods on bytes and bytearray objects don't accept strings as their arguments, just as the methods on strings don't accept bytes as their arguments. For example, you have to write:

```
a = "abc"
b = a.replace("a", "f")
```

and:

```
a = b"abc"
b = a.replace(b"a", b"f")
```

Some bytes and bytearray operations assume the use of ASCII compatible binary formats, and hence should be avoided when working with arbitrary binary data. These restrictions are covered below.

Note

Using these ASCII based operations to manipulate binary data that is not stored in an ASCII based format may lead to data corruption.

The following methods on bytes and bytearray objects can be used with arbitrary binary data.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2698)

Unknown directive type "method".

```
.. method:: bytes.count(sub[, start[, end]])
            bytearray.count(sub[, start[, end]])

   Return the number of non-overlapping occurrences of subsequence *sub* in
   the range [*start*, *end*].  Optional arguments *start* and *end* are
   interpreted as in slice notation.

   The subsequence to search for may be any :term:`bytes-like object` or an
   integer in the range 0 to 255.

   .. versionchanged:: 3.3
      Also accept an integer in the range 0 to 255 as the subsequence.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2712)

Unknown directive type "method".

```
.. method:: bytes.removeprefix(prefix, /)
            bytearray.removeprefix(prefix, /)
```

```
   If the binary data starts with the *prefix* string, return
   ``bytes[len(prefix):]``. Otherwise, return a copy of the original
   binary data::

      >>> b'TestHook'.removeprefix(b'Test')
      b'Hook'
      >>> b'BaseTestCase'.removeprefix(b'Test')
      b'BaseTestCase'

   The *prefix* may be any :term:`bytes-like object`.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2734)

Unknown directive type "method".

```
.. method:: bytes.removesuffix(suffix, /)
            bytearray.removesuffix(suffix, /)

   If the binary data ends with the *suffix* string and that *suffix* is
   not empty, return ``bytes[:-len(suffix)]``.  Otherwise, return a copy of
   the original binary data::

      >>> b'MiscTests'.removesuffix(b'Tests')
      b'Misc'
      >>> b'TmpDirMixin'.removesuffix(b'Tests')
      b'TmpDirMixin'

   The *suffix* may be any :term:`bytes-like object`.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2756)

Unknown directive type "method".

```
.. method:: bytes.decode(encoding="utf-8", errors="strict")
            bytearray.decode(encoding="utf-8", errors="strict")

   Return a string decoded from the given bytes.  Default encoding is
   ``'utf-8'``. *errors* may be given to set a different
   error handling scheme.  The default for *errors* is ``'strict'``, meaning
   that encoding errors raise a :exc:`UnicodeError`.  Other possible values are
   ``'ignore'``, ``'replace'`` and any other name registered via
   :func:`codecs.register_error`, see section :ref:`error-handlers`. For a
   list of possible encodings, see section :ref:`standard-encodings`.

   By default, the *errors* argument is not checked for best performances, but
   only used at the first decoding error. Enable the :ref:`Python Development
   Mode <devmode>`, or use a :ref:`debug build <debug-build>` to check *errors*.

   .. note::

      Passing the *encoding* argument to :class:`str` allows decoding any
      :term:`bytes-like object` directly, without needing to make a temporary
      bytes or bytearray object.

   .. versionchanged:: 3.1
      Added support for keyword arguments.

   .. versionchanged:: 3.9
      The *errors* is now checked in development mode and
      in :ref:`debug mode <debug-build>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2785)

Unknown directive type "method".

```
.. method:: bytes.endswith(suffix[, start[, end]])
            bytearray.endswith(suffix[, start[, end]])

   Return ``True`` if the binary data ends with the specified *suffix*,
   otherwise return ``False``.  *suffix* can also be a tuple of suffixes to
   look for.  With optional *start*, test beginning at that position.  With
   optional *end*, stop comparing at that position.

   The suffix(es) to search for may be any :term:`bytes-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2796)

Unknown directive type "method".

```
.. method:: bytes.find(sub[, start[, end]])
            bytearray.find(sub[, start[, end]])

   Return the lowest index in the data where the subsequence *sub* is found,
   such that *sub* is contained in the slice ``s[start:end]``.  Optional
   arguments *start* and *end* are interpreted as in slice notation.  Return
   ``-1`` if *sub* is not found.

   The subsequence to search for may be any :term:`bytes-like object` or an
   integer in the range 0 to 255.

   .. note::

      The :meth:`~bytes.find` method should be used only if you need to know the
      position of *sub*.  To check if *sub* is a substring or not, use the
      :keyword:`in` operator::

         >>> b'Py' in b'Python'
         True

   .. versionchanged:: 3.3
      Also accept an integer in the range 0 to 255 as the subsequence.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2820)

Unknown directive type "method".

```
.. method:: bytes.index(sub[, start[, end]])
            bytearray.index(sub[, start[, end]])

   Like :meth:`~bytes.find`, but raise :exc:`ValueError` when the
   subsequence is not found.

   The subsequence to search for may be any :term:`bytes-like object` or an
   integer in the range 0 to 255.

   .. versionchanged:: 3.3
      Also accept an integer in the range 0 to 255 as the subsequence.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2833)

Unknown directive type "method".

```
.. method:: bytes.join(iterable)
            bytearray.join(iterable)

   Return a bytes or bytearray object which is the concatenation of the
   binary data sequences in *iterable*.  A :exc:`TypeError` will be raised
   if there are any values in *iterable* that are not :term:`bytes-like
   objects <bytes-like object>`, including :class:`str` objects.  The
   separator between elements is the contents of the bytes or
```

```
   bytearray object providing this method.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2844)

Unknown directive type "staticmethod".

```
.. staticmethod:: bytes.maketrans(from, to)
                  bytearray.maketrans(from, to)

   This static method returns a translation table usable for
   :meth:`bytes.translate` that will map each character in *from* into the
   character at the same position in *to*; *from* and *to* must both be
   :term:`bytes-like objects <bytes-like object>` and have the same length.

   .. versionadded:: 3.1
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2855)

Unknown directive type "method".

```
.. method:: bytes.partition(sep)
            bytearray.partition(sep)

   Split the sequence at the first occurrence of *sep*, and return a 3-tuple
   containing the part before the separator, the separator itself or its
   bytearray copy, and the part after the separator.
   If the separator is not found, return a 3-tuple
   containing a copy of the original sequence, followed by two empty bytes or
   bytearray objects.

   The separator to search for may be any :term:`bytes-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2868)

Unknown directive type "method".

```
.. method:: bytes.replace(old, new[, count])
            bytearray.replace(old, new[, count])

   Return a copy of the sequence with all occurrences of subsequence *old*
   replaced by *new*.  If the optional argument *count* is given, only the
   first *count* occurrences are replaced.

   The subsequence to search for and its replacement may be any
   :term:`bytes-like object`.

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2884)

Unknown directive type "method".

```
.. method:: bytes.rfind(sub[, start[, end]])
            bytearray.rfind(sub[, start[, end]])

   Return the highest index in the sequence where the subsequence *sub* is
   found, such that *sub* is contained within ``s[start:end]``.  Optional
   arguments *start* and *end* are interpreted as in slice notation. Return
   ``-1`` on failure.

   The subsequence to search for may be any :term:`bytes-like object` or an
   integer in the range 0 to 255.

   .. versionchanged:: 3.3
      Also accept an integer in the range 0 to 255 as the subsequence.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2899)

Unknown directive type "method".

```
.. method:: bytes.rindex(sub[, start[, end]])
            bytearray.rindex(sub[, start[, end]])

   Like :meth:`~bytes.rfind` but raises :exc:`ValueError` when the
   subsequence *sub* is not found.

   The subsequence to search for may be any :term:`bytes-like object` or an
   integer in the range 0 to 255.

   .. versionchanged:: 3.3
      Also accept an integer in the range 0 to 255 as the subsequence.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2912)

Unknown directive type "method".

```
.. method:: bytes.rpartition(sep)
            bytearray.rpartition(sep)

   Split the sequence at the last occurrence of *sep*, and return a 3-tuple
   containing the part before the separator, the separator itself or its
   bytearray copy, and the part after the separator.
   If the separator is not found, return a 3-tuple
   containing two empty bytes or bytearray objects, followed by a copy of the
   original sequence.

   The separator to search for may be any :term:`bytes-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2925)

Unknown directive type "method".

```
.. method:: bytes.startswith(prefix[, start[, end]])
            bytearray.startswith(prefix[, start[, end]])

   Return ``True`` if the binary data starts with the specified *prefix*,
   otherwise return ``False``.  *prefix* can also be a tuple of prefixes to
   look for.  With optional *start*, test beginning at that position.  With
   optional *end*, stop comparing at that position.

   The prefix(es) to search for may be any :term:`bytes-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2936)

Unknown directive type "method".

```
.. method:: bytes.translate(table, /, delete=b'')
            bytearray.translate(table, /, delete=b'')

   Return a copy of the bytes or bytearray object where all bytes occurring in
   the optional argument *delete* are removed, and the remaining bytes have
   been mapped through the given translation table, which must be a bytes
   object of length 256.

   You can use the :func:`bytes.maketrans` method to create a translation
   table.

   Set the *table* argument to ``None`` for translations that only delete
   characters::

      >>> b'read this short text'.translate(None, b'aeiou')
      b'rd ths shrt txt'
```

```
   .. versionchanged:: 3.6
      *delete* is now supported as a keyword argument.
```

The following methods on bytes and bytearray objects have default behaviours that assume the use of ASCII compatible binary formats, but can still be used with arbitrary binary data by passing appropriate arguments. Note that all of the bytearray methods in this section do *not* operate in place, and instead produce new objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2963)

Unknown directive type "method".

```
.. method:: bytes.center(width[, fillbyte])
            bytearray.center(width[, fillbyte])

   Return a copy of the object centered in a sequence of length *width*.
   Padding is done using the specified *fillbyte* (default is an ASCII
   space). For :class:`bytes` objects, the original sequence is returned if
   *width* is less than or equal to ``len(s)``.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2977)

Unknown directive type "method".

```
.. method:: bytes.ljust(width[, fillbyte])
            bytearray.ljust(width[, fillbyte])

   Return a copy of the object left justified in a sequence of length *width*.
   Padding is done using the specified *fillbyte* (default is an ASCII
   space). For :class:`bytes` objects, the original sequence is returned if
   *width* is less than or equal to ``len(s)``.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 2991)

Unknown directive type "method".

```
.. method:: bytes.lstrip([chars])
            bytearray.lstrip([chars])

   Return a copy of the sequence with specified leading bytes removed.  The
   *chars* argument is a binary sequence specifying the set of byte values to
   be removed - the name refers to the fact this method is usually used with
   ASCII characters.  If omitted or ``None``, the *chars* argument defaults
   to removing ASCII whitespace.  The *chars* argument is not a prefix;
   rather, all combinations of its values are stripped::

      >>> b'   spacious   '.lstrip()
      b'spacious   '
      >>> b'www.example.com'.lstrip(b'cmowz.')
      b'example.com'

   The binary sequence of byte values to remove may be any
   :term:`bytes-like object`. See :meth:`~bytes.removeprefix` for a method
   that will remove a single prefix string rather than all of a set of
   characters.  For example::

      >>> b'Arthur: three!'.lstrip(b'Arthur: ')
      b'ee!'
      >>> b'Arthur: three!'.removeprefix(b'Arthur: ')
      b'three!'

   .. note::
```

```
      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3022)

Unknown directive type "method".

```
.. method:: bytes.rjust(width[, fillbyte])
            bytearray.rjust(width[, fillbyte])

   Return a copy of the object right justified in a sequence of length *width*.
   Padding is done using the specified *fillbyte* (default is an ASCII
   space). For :class:`bytes` objects, the original sequence is returned if
   *width* is less than or equal to ``len(s)``.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3036)

Unknown directive type "method".

```
.. method:: bytes.rsplit(sep=None, maxsplit=-1)
            bytearray.rsplit(sep=None, maxsplit=-1)

   Split the binary sequence into subsequences of the same type, using *sep*
   as the delimiter string. If *maxsplit* is given, at most *maxsplit* splits
   are done, the *rightmost* ones.  If *sep* is not specified or ``None``,
   any subsequence consisting solely of ASCII whitespace is a separator.
   Except for splitting from the right, :meth:`rsplit` behaves like
   :meth:`split` which is described in detail below.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3047)

Unknown directive type "method".

```
.. method:: bytes.rstrip([chars])
            bytearray.rstrip([chars])

   Return a copy of the sequence with specified trailing bytes removed.  The
   *chars* argument is a binary sequence specifying the set of byte values to
   be removed - the name refers to the fact this method is usually used with
   ASCII characters.  If omitted or ``None``, the *chars* argument defaults to
   removing ASCII whitespace.  The *chars* argument is not a suffix; rather,
   all combinations of its values are stripped::

      >>> b'   spacious   '.rstrip()
      b'   spacious'
      >>> b'mississippi'.rstrip(b'ipz')
      b'mississ'

   The binary sequence of byte values to remove may be any
   :term:`bytes-like object`. See :meth:`~bytes.removesuffix` for a method
   that will remove a single suffix string rather than all of a set of
   characters.  For example::

      >>> b'Monty Python'.rstrip(b' Python')
      b'M'
      >>> b'Monty Python'.removesuffix(b' Python')
      b'Monty'

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3078)

Unknown directive type "method".

```
.. method:: bytes.split(sep=None, maxsplit=-1)
            bytearray.split(sep=None, maxsplit=-1)

   Split the binary sequence into subsequences of the same type, using *sep*
   as the delimiter string. If *maxsplit* is given and non-negative, at most
   *maxsplit* splits are done (thus, the list will have at most ``maxsplit+1``
   elements).  If *maxsplit* is not specified or is ``-1``, then there is no
   limit on the number of splits (all possible splits are made).

   If *sep* is given, consecutive delimiters are not grouped together and are
   deemed to delimit empty subsequences (for example, ``b'1,,2'.split(b',')``
   returns ``[b'1', b'', b'2']``).  The *sep* argument may consist of a
   multibyte sequence (for example, ``b'1<>2<>3'.split(b'<>')`` returns
   ``[b'1', b'2', b'3']``). Splitting an empty sequence with a specified
   separator returns ``[b'']`` or ``[bytearray(b'')]`` depending on the type
   of object being split.  The *sep* argument may be any
   :term:`bytes-like object`.

   For example::

      >>> b'1,2,3'.split(b',')
      [b'1', b'2', b'3']
      >>> b'1,2,3'.split(b',', maxsplit=1)
      [b'1', b'2,3']
      >>> b'1,2,,3,'.split(b',')
      [b'1', b'2', b'', b'3', b'']

   If *sep* is not specified or is ``None``, a different splitting algorithm
   is applied: runs of consecutive ASCII whitespace are regarded as a single
   separator, and the result will contain no empty strings at the start or
   end if the sequence has leading or trailing whitespace.  Consequently,
   splitting an empty sequence or a sequence consisting solely of ASCII
   whitespace without a specified separator returns ``[]``.

   For example::


      >>> b'1 2 3'.split()
      [b'1', b'2', b'3']
      >>> b'1 2 3'.split(maxsplit=1)
      [b'1', b'2 3']
      >>> b'   1   2   3   '.split()
      [b'1', b'2', b'3']
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3123)

Unknown directive type "method".

```
.. method:: bytes.strip([chars])
            bytearray.strip([chars])

   Return a copy of the sequence with specified leading and trailing bytes
   removed. The *chars* argument is a binary sequence specifying the set of
   byte values to be removed - the name refers to the fact this method is
   usually used with ASCII characters.  If omitted or ``None``, the *chars*
   argument defaults to removing ASCII whitespace. The *chars* argument is
   not a prefix or suffix; rather, all combinations of its values are
   stripped::

      >>> b'   spacious   '.strip()
      b'spacious'
      >>> b'www.example.com'.strip(b'cmowz.')
      b'example'

   The binary sequence of byte values to remove may be any
   :term:`bytes-like object`.

   .. note::

      The bytearray version of this method does *not* operate in place -
      it always produces a new object, even if no changes were made.
```

The following methods on bytes and bytearray objects assume the use of ASCII compatible binary formats and should not be applied to arbitrary binary data. Note that all of the bytearray methods in this section do *not* operate in place, and instead produce new objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3153)

Unknown directive type "method".

```
.. method:: bytes.capitalize()
            bytearray.capitalize()

   Return a copy of the sequence with each byte interpreted as an ASCII
   character, and the first byte capitalized and the rest lowercased.
   Non-ASCII byte values are passed through unchanged.

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3166)

Unknown directive type "method".

```
.. method:: bytes.expandtabs(tabsize=8)
            bytearray.expandtabs(tabsize=8)

   Return a copy of the sequence where all ASCII tab characters are replaced
   by one or more ASCII spaces, depending on the current column and the given
   tab size.  Tab positions occur every *tabsize* bytes (default is 8,
   giving tab positions at columns 0, 8, 16 and so on).  To expand the
   sequence, the current column is set to zero and the sequence is examined
   byte by byte.  If the byte is an ASCII tab character (``b'\t'``), one or
   more space characters are inserted in the result until the current column
   is equal to the next tab position. (The tab character itself is not
   copied.)  If the current byte is an ASCII newline (``b'\n'``) or
   carriage return (``b'\r'``), it is copied and the current column is reset
   to zero.  Any other byte value is copied unchanged and the current column
   is incremented by one regardless of how the byte value is represented when
   printed::

      >>> b'01\t012\t0123\t01234'.expandtabs()
      b'01      012     0123    01234'
      >>> b'01\t012\t0123\t01234'.expandtabs(4)
      b'01  012 0123    01234'

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3194)

Unknown directive type "method".

```
.. method:: bytes.isalnum()
            bytearray.isalnum()

   Return ``True`` if all bytes in the sequence are alphabetical ASCII characters
   or ASCII decimal digits and the sequence is not empty, ``False`` otherwise.
   Alphabetic ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'``. ASCII decimal
   digits are those byte values in the sequence ``b'0123456789'``.

   For example::

      >>> b'ABCabc1'.isalnum()
      True
      >>> b'ABC abc1'.isalnum()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3211)

Unknown directive type "method".

```
.. method:: bytes.isalpha()
            bytearray.isalpha()

   Return ``True`` if all bytes in the sequence are alphabetic ASCII characters
   and the sequence is not empty, ``False`` otherwise.  Alphabetic ASCII
   characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'``.

   For example::

      >>> b'ABCabc'.isalpha()
      True
      >>> b'ABCabc1'.isalpha()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3227)

Unknown directive type "method".

```
.. method:: bytes.isascii()
            bytearray.isascii()

   Return ``True`` if the sequence is empty or all bytes in the sequence are ASCII,
   ``False`` otherwise.
   ASCII bytes are in the range 0-0x7F.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3237)

Unknown directive type "method".

```
.. method:: bytes.isdigit()
            bytearray.isdigit()

   Return ``True`` if all bytes in the sequence are ASCII decimal digits
   and the sequence is not empty, ``False`` otherwise. ASCII decimal digits are
   those byte values in the sequence ``b'0123456789'``.

   For example::

      >>> b'1234'.isdigit()
      True
      >>> b'1.23'.isdigit()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3252)

Unknown directive type "method".

```
.. method:: bytes.islower()
            bytearray.islower()

   Return ``True`` if there is at least one lowercase ASCII character
   in the sequence and no uppercase ASCII characters, ``False`` otherwise.

   For example::

      >>> b'hello world'.islower()
      True
      >>> b'Hello world'.islower()
      False

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3270)

Unknown directive type "method".

```
.. method:: bytes.isspace()
            bytearray.isspace()

   Return ``True`` if all bytes in the sequence are ASCII whitespace and the
   sequence is not empty, ``False`` otherwise.  ASCII whitespace characters are
   those byte values in the sequence ``b' \t\n\r\x0b\f'`` (space, tab, newline,
   carriage return, vertical tab, form feed).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3279)

Unknown directive type "method".

```
.. method:: bytes.istitle()
            bytearray.istitle()

   Return ``True`` if the sequence is ASCII titlecase and the sequence is not
   empty, ``False`` otherwise. See :meth:`bytes.title` for more details on the
   definition of "titlecase".

   For example::

      >>> b'Hello World'.istitle()
      True
      >>> b'Hello world'.istitle()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3294)

Unknown directive type "method".

```
.. method:: bytes.isupper()
            bytearray.isupper()

   Return ``True`` if there is at least one uppercase alphabetic ASCII character
   in the sequence and no lowercase ASCII characters, ``False`` otherwise.

   For example::

      >>> b'HELLO WORLD'.isupper()
      True
      >>> b'Hello world'.isupper()
      False

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3312)

Unknown directive type "method".

```
.. method:: bytes.lower()
            bytearray.lower()

   Return a copy of the sequence with all the uppercase ASCII characters
   converted to their corresponding lowercase counterpart.

   For example::

      >>> b'Hello World'.lower()
      b'hello world'

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
```

```
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3333)

Unknown directive type "index".

```
.. index::
   single: universal newlines; bytes.splitlines method
   single: universal newlines; bytearray.splitlines method
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3337)

Unknown directive type "method".

```
.. method:: bytes.splitlines(keepends=False)
            bytearray.splitlines(keepends=False)

   Return a list of the lines in the binary sequence, breaking at ASCII
   line boundaries. This method uses the :term:`universal newlines` approach
   to splitting lines. Line breaks are not included in the resulting list
   unless *keepends* is given and true.

   For example::

      >>> b'ab c\n\nde fg\rkl\r\n'.splitlines()
      [b'ab c', b'', b'de fg', b'kl']
      >>> b'ab c\n\nde fg\rkl\r\n'.splitlines(keepends=True)
      [b'ab c\n', b'\n', b'de fg\r', b'kl\r\n']

   Unlike :meth:`~bytes.split` when a delimiter string *sep* is given, this
   method returns an empty list for the empty string, and a terminal line
   break does not result in an extra line::

      >>> b"".split(b'\n'), b"Two lines\n".split(b'\n')
      ([b''], [b'Two lines', b''])
      >>> b"".splitlines(), b"One line\n".splitlines()
      ([], [b'One line'])
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3362)

Unknown directive type "method".

```
.. method:: bytes.swapcase()
            bytearray.swapcase()

   Return a copy of the sequence with all the lowercase ASCII characters
   converted to their corresponding uppercase counterpart and vice-versa.

   For example::

      >>> b'Hello World'.swapcase()
      b'hELLO wORLD'

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.

   Unlike :func:`str.swapcase()`, it is always the case that
   ``bin.swapcase().swapcase() == bin`` for the binary versions. Case
   conversions are symmetrical in ASCII, even though that is not generally
   true for arbitrary Unicode code points.

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3388)

Unknown directive type "method".

```
.. method:: bytes.title()
            bytearray.title()

   Return a titlecased version of the binary sequence where words start with
   an uppercase ASCII character and the remaining characters are lowercase.
   Uncased byte values are left unmodified.

   For example::

      >>> b'Hello world'.title()
      b'Hello World'

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.
   All other byte values are uncased.

   The algorithm uses a simple language-independent definition of a word as
   groups of consecutive letters.  The definition works in many contexts but
   it means that apostrophes in contractions and possessives form word
   boundaries, which may not be the desired result::

        >>> b"they're bill's friends from the UK".title()
        b"They're Bill'S Friends From The Uk"

   A workaround for apostrophes can be constructed using regular expressions::

        >>> import re
        >>> def titlecase(s):
        ...     return re.sub(rb"[A-Za-z]+('[A-Za-z]+)?",
        ...                   lambda mo: mo.group(0)[0:1].upper() +
        ...                              mo.group(0)[1:].lower(),
        ...                   s)
        ...
        >>> titlecase(b"they're bill's friends.")
        b"They're Bill's Friends."

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3431)

Unknown directive type "method".

```
.. method:: bytes.upper()
            bytearray.upper()

   Return a copy of the sequence with all the lowercase ASCII characters
   converted to their corresponding uppercase counterpart.

   For example::

      >>> b'Hello World'.upper()
      b'HELLO WORLD'

   Lowercase ASCII characters are those byte values in the sequence
   ``b'abcdefghijklmnopqrstuvwxyz'``. Uppercase ASCII characters
   are those byte values in the sequence ``b'ABCDEFGHIJKLMNOPQRSTUVWXYZ'``.

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3452)

Unknown directive type "method".

```
.. method:: bytes.zfill(width)
            bytearray.zfill(width)

   Return a copy of the sequence left filled with ASCII ``b'0'`` digits to
   make a sequence of length *width*. A leading sign prefix (``b'+'``/
   ``b'-'``) is handled by inserting the padding *after* the sign character
   rather than before. For :class:`bytes` objects, the original sequence is
   returned if *width* is less than or equal to ``len(seq)``.

   For example::

      >>> b"42".zfill(5)
      b'00042'
      >>> b"-42".zfill(5)
      b'-0042'

   .. note::

      The bytearray version of this method does *not* operate in place - it
      always produces a new object, even if no changes were made.
```

`printf`-style Bytes Formatting

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3479)

Unknown directive type "index".

```
.. index::
   single: formatting; bytes (%)
   single: formatting; bytearray (%)
   single: interpolation; bytes (%)
   single: interpolation; bytearray (%)
   single: bytes; formatting
   single: bytearray; formatting
   single: bytes; interpolation
   single: bytearray; interpolation
   single: printf-style formatting
   single: sprintf-style formatting
   single: % (percent); printf-style formatting
```

Note

The formatting operations described here exhibit a variety of quirks that lead to a number of common errors (such as failing to display tuples and dictionaries correctly). If the value being printed may be a tuple or dictionary, wrap it in a tuple.

Bytes objects (`bytes`/`bytearray`) have one unique built-in operation: the `%` operator (modulo). This is also known as the bytes *formatting* or *interpolation* operator. Given `format % values` (where *format* is a bytes object), `%` conversion specifications in *format* are replaced with zero or more elements of *values*. The effect is similar to using the :c:func:`sprintf` in the C language.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3499); *backlink*

Unknown interpreted text role "c:func".

If *format* requires a single argument, *values* may be a single non-tuple object. [5] Otherwise, *values* must be a tuple with exactly the number of items specified by the format bytes object, or a single mapping object (for example, a dictionary).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3511)

Unknown directive type "index".

```
.. index::
   single: () (parentheses); in printf-style formatting
   single: * (asterisk); in printf-style formatting
   single: . (dot); in printf-style formatting
```

A conversion specifier contains two or more characters and has the following components, which must occur in this order:

1. The `'%'` character, which marks the start of the specifier.

2. Mapping key (optional), consisting of a parenthesised sequence of characters (for example, `(somename)`).
3. Conversion flags (optional), which affect the result of some conversion types.
4. Minimum field width (optional). If specified as an `'*'` (asterisk), the actual width is read from the next element of the tuple in *values*, and the object to convert comes after the minimum field width and optional precision.
5. Precision (optional), given as a `'.'` (dot) followed by the precision. If specified as `'*'` (an asterisk), the actual precision is read from the next element of the tuple in *values*, and the value to convert comes after the precision.
6. Length modifier (optional).
7. Conversion type.

When the right argument is a dictionary (or other mapping type), then the formats in the bytes object *must* include a parenthesised mapping key into that dictionary inserted immediately after the `'%'` character. The mapping key selects the value to be formatted from the mapping. For example:

```
>>> print(b'%(language)s has %(number)03d quote types.' %
...       {b'language': b"Python", b"number": 2})
b'Python has 002 quote types.'
```

In this case no `*` specifiers may occur in a format (since they require a sequential parameter list).

The conversion flag characters are:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst`, line 3554)

Unknown directive type "index".

```
.. index::
   single: # (hash); in printf-style formatting
   single: - (minus); in printf-style formatting
   single: + (plus); in printf-style formatting
   single: space; in printf-style formatting
```

Flag	Meaning
`'#'`	The value conversion will use the "alternate form" (where defined below).
`'0'`	The conversion will be zero padded for numeric values.
`'-'`	The converted value is left adjusted (overrides the `'0'` conversion if both are given).
`' '`	(a space) A blank should be left before a positive number (or empty string) produced by a signed conversion.
`'+'`	A sign character (`'+'` or `'-'`) will precede the conversion (overrides a "space" flag).

A length modifier (`h`, `l`, or `L`) may be present, but is ignored as it is not necessary for Python -- so e.g. `%ld` is identical to `%d`.

The conversion types are:

Conversion	Meaning	Notes
`'d'`	Signed integer decimal.	
`'i'`	Signed integer decimal.	
`'o'`	Signed octal value.	(1)
`'u'`	Obsolete type -- it is identical to `'d'`.	(8)
`'x'`	Signed hexadecimal (lowercase).	(2)
`'X'`	Signed hexadecimal (uppercase).	(2)
`'e'`	Floating point exponential format (lowercase).	(3)
`'E'`	Floating point exponential format (uppercase).	(3)
`'f'`	Floating point decimal format.	(3)
`'F'`	Floating point decimal format.	(3)
`'g'`	Floating point format. Uses lowercase exponential format if exponent is less than -4 or not less than precision, decimal format otherwise.	(4)
`'G'`	Floating point format. Uses uppercase exponential format if exponent is less than -4 or not less than precision, decimal format otherwise.	(4)
`'c'`	Single byte (accepts integer or single byte objects).	

Conversion	Meaning	Notes
`'b'`	Bytes (any object that follows the :ref:`buffer protocol <bufferobjects>` or has :meth:`__bytes__`). **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3618); *backlink*** Unknown interpreted text role "ref". **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3618); *backlink*** Unknown interpreted text role "meth".	(5)
`'s'`	`'s'` is an alias for `'b'` and should only be used for Python2/3 code bases.	(6)
`'a'`	Bytes (converts any Python object using `repr(obj).encode('ascii', 'backslashreplace')`).	(5)
`'r'`	`'r'` is an alias for `'a'` and should only be used for Python2/3 code bases.	(7)
`'%'`	No argument is converted, results in a `'%'` character in the result.	

Notes:

1. The alternate form causes a leading octal specifier (`'0o'`) to be inserted before the first digit.
2. The alternate form causes a leading `'0x'` or `'0X'` (depending on whether the `'x'` or `'X'` format was used) to be inserted before the first digit.
3. The alternate form causes the result to always contain a decimal point, even if no digits follow it.

   The precision determines the number of digits after the decimal point and defaults to 6.
4. The alternate form causes the result to always contain a decimal point, and trailing zeroes are not removed as they would otherwise be.

   The precision determines the number of significant digits before and after the decimal point and defaults to 6.
5. If precision is `N`, the output is truncated to `N` characters.
6. `b'%s'` is deprecated, but will not be removed during the 3.x series.
7. `b'%r'` is deprecated, but will not be removed during the 3.x series.
8. See PEP 237.

Note

The bytearray version of this method does *not* operate in place - it always produces a new object, even if no changes were made.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3675)

Unknown directive type "seealso".

```
.. seealso::

   :pep:`461` - Adding % formatting to bytes and bytearray
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3679)

Unknown directive type "versionadded".

```
.. versionadded:: 3.5
```

Memory Views

:class:`memoryview` objects allow Python code to access the internal data of an object that supports the :ref:`buffer protocol <bufferobjects>` without copying.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3686); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3686); *backlink*

Unknown interpreted text role "ref".

Create a :class:`memoryview` that references *object*. *object* must support the buffer protocol. Built-in objects that support the buffer protocol include :class:`bytes` and :class:`bytearray`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3692); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3692); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3692); *backlink*

Unknown interpreted text role "class".

A :class:`memoryview` has the notion of an *element*, which is the atomic memory unit handled by the originating *object*. For many simple types such as :class:`bytes` and :class:`bytearray`, an element is a single byte, but other types such as :class:`array.array` may have bigger elements.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3696); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3696); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3696); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3696); *backlink*

Unknown interpreted text role "class".

`len(view)` is equal to the length of :class:`~memoryview.tolist`. If `view.ndim = 0`, the length is 1. If `view.ndim = 1`, the length is equal to the number of elements in the view. For higher dimensions, the length is equal to the length of the nested list representation of the view. The :class:`~memoryview.itemsize` attribute will give you the number of bytes in a single element.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3701); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3701); *backlink*

Unknown interpreted text role "class".

A :class:`memoryview` supports slicing and indexing to expose its data. One-dimensional slicing will result in a subview:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3708); *backlink*

Unknown interpreted text role "class".

```
>>> v = memoryview(b'abcefg')
>>> v[1]
98
>>> v[-1]
103
>>> v[1:4]
<memory at 0x7f3ddc9f4350>
>>> bytes(v[1:4])
b'bce'
```

If :class:`~memoryview.format` is one of the native format specifiers from the :mod:`struct` module, indexing with an integer or a tuple of integers is also supported and returns a single *element* with the correct type. One-dimensional memoryviews can be indexed with an integer or a one-integer tuple. Multi-dimensional memoryviews can be indexed with tuples of exactly *ndim* integers where *ndim* is the number of dimensions. Zero-dimensional memoryviews can be indexed with the empty tuple.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3721); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3721); *backlink*

Unknown interpreted text role "mod".

Here is an example with a non-byte format:

```
>>> import array
>>> a = array.array('l', [-11111111, 22222222, -33333333, 44444444])
>>> m = memoryview(a)
>>> m[0]
-11111111
>>> m[-1]
44444444
>>> m[::2].tolist()
[-11111111, -33333333]
```

If the underlying object is writable, the memoryview supports one-dimensional slice assignment. Resizing is not allowed:

```
>>> data = bytearray(b'abcefg')
>>> v = memoryview(data)
>>> v.readonly
False
>>> v[0] = ord(b'z')
>>> data
bytearray(b'zbcefg')
>>> v[1:4] = b'123'
>>> data
bytearray(b'z123fg')
>>> v[2:3] = b'spam'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: memoryview assignment: lvalue and rvalue have different structures
>>> v[2:6] = b'spam'
>>> data
bytearray(b'z1spam')
```

One-dimensional memoryviews of hashable (read-only) types with formats 'B', 'b' or 'c' are also hashable. The hash is defined as `hash(m) == hash(m.tobytes())`:

```
>>> v = memoryview(b'abcefg')
>>> hash(v) == hash(b'abcefg')
True
>>> hash(v[2:4]) == hash(b'ce')
True
>>> hash(v[::-2]) == hash(b'abcefg'[::-2])
True
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3775)

Unknown directive type "versionchanged".

```
   .. versionchanged:: 3.3
      One-dimensional memoryviews can now be sliced.
      One-dimensional memoryviews with formats 'B', 'b' or 'c' are now hashable.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3779)

Unknown directive type "versionchanged".

```
   .. versionchanged:: 3.4
      memoryview is now registered automatically with
      :class:`collections.abc.Sequence`
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3783)

Unknown directive type "versionchanged".

```
   .. versionchanged:: 3.5
      memoryviews can now be indexed with tuple of integers.
```

:class:`memoryview` has several methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3786); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3788)

Unknown directive type "method".

```
   .. method:: __eq__(exporter)

      A memoryview and a :pep:`3118` exporter are equal if their shapes are
      equivalent and if all corresponding values are equal when the operands'
      respective format codes are interpreted using :mod:`struct` syntax.

      For the subset of :mod:`struct` format strings currently supported by
      :meth:`tolist`, ``v`` and ``w`` are equal if ``v.tolist() == w.tolist()``::

         >>> import array
         >>> a = array.array('I', [1, 2, 3, 4, 5])
         >>> b = array.array('d', [1.0, 2.0, 3.0, 4.0, 5.0])
         >>> c = array.array('b', [5, 3, 1])
         >>> x = memoryview(a)
         >>> y = memoryview(b)
         >>> x == a == y == b
         True
         >>> x.tolist() == a.tolist() == y.tolist() == b.tolist()
         True
         >>> z = y[::-2]
         >>> z == c
         True
         >>> z.tolist() == c.tolist()
         True

      If either format string is not supported by the :mod:`struct` module,
      then the objects will always compare as unequal (even if the format
      strings and buffer contents are identical)::

         >>> from ctypes import BigEndianStructure, c_long
         >>> class BEPoint(BigEndianStructure):
         ...     _fields_ = [("x", c_long), ("y", c_long)]
         ...
         >>> point = BEPoint(100, 200)
         >>> a = memoryview(point)
         >>> b = memoryview(point)
         >>> a == point
         False
         >>> a == b
         False

      Note that, as with floating point numbers, ``v is w`` does *not* imply
      ``v == w`` for memoryview objects.
```

```
.. versionchanged:: 3.3
   Previous versions compared the raw memory disregarding the item format
   and the logical array structure.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3836)

Unknown directive type "method".

```
.. method:: tobytes(order='C')

   Return the data in the buffer as a bytestring.  This is equivalent to
   calling the :class:`bytes` constructor on the memoryview. ::

      >>> m = memoryview(b"abc")
      >>> m.tobytes()
      b'abc'
      >>> bytes(m)
      b'abc'

   For non-contiguous arrays the result is equal to the flattened list
   representation with all elements converted to bytes. :meth:`tobytes`
   supports all format strings, including those that are not in
   :mod:`struct` module syntax.

   .. versionadded:: 3.8
      *order* can be {'C', 'F', 'A'}.  When *order* is 'C' or 'F', the data
      of the original array is converted to C or Fortran order. For contiguous
      views, 'A' returns an exact copy of the physical memory. In particular,
      in-memory Fortran order is preserved. For non-contiguous views, the
      data is converted to C first. *order=None* is the same as *order='C'*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3859)

Unknown directive type "method".

```
.. method:: hex([sep[, bytes_per_sep]])

   Return a string object containing two hexadecimal digits for each
   byte in the buffer. ::

      >>> m = memoryview(b"abc")
      >>> m.hex()
      '616263'

   .. versionadded:: 3.5

   .. versionchanged:: 3.8
      Similar to :meth:`bytes.hex`, :meth:`memoryview.hex` now supports
      optional *sep* and *bytes_per_sep* parameters to insert separators
      between bytes in the hex output.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3875)

Unknown directive type "method".

```
.. method:: tolist()

   Return the data in the buffer as a list of elements. ::

      >>> memoryview(b'abc').tolist()
      [97, 98, 99]
      >>> import array
      >>> a = array.array('d', [1.1, 2.2, 3.3])
      >>> m = memoryview(a)
      >>> m.tolist()
      [1.1, 2.2, 3.3]

   .. versionchanged:: 3.3
      :meth:`tolist` now supports all single character native formats in
      :mod:`struct` module syntax as well as multi-dimensional
      representations.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3892)

Unknown directive type "method".

```
.. method:: toreadonly()

   Return a readonly version of the memoryview object.  The original
   memoryview object is unchanged. ::

      >>> m = memoryview(bytearray(b'abc'))
      >>> mm = m.toreadonly()
      >>> mm.tolist()
      [89, 98, 99]
      >>> mm[0] = 42
      Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
      TypeError: cannot modify read-only memory
      >>> m[0] = 43
      >>> mm.tolist()
      [43, 98, 99]

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3911)

Unknown directive type "method".

```
.. method:: release()

   Release the underlying buffer exposed by the memoryview object.  Many
   objects take special actions when a view is held on them (for example,
   a :class:`bytearray` would temporarily forbid resizing); therefore,
   calling release() is handy to remove these restrictions (and free any
   dangling resources) as soon as possible.

   After this method has been called, any further operation on the view
   raises a :class:`ValueError` (except :meth:`release()` itself which can
   be called multiple times)::

      >>> m = memoryview(b'abc')
      >>> m.release()
      >>> m[0]
      Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
      ValueError: operation forbidden on released memoryview object

   The context management protocol can be used for a similar effect,
   using the ``with`` statement::

      >>> with memoryview(b'abc') as m:
      ...     m[0]
      ...
      97
      >>> m[0]
      Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
      ValueError: operation forbidden on released memoryview object

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 3944)

Unknown directive type "method".

```
.. method:: cast(format[, shape])

   Cast a memoryview to a new format or shape. *shape* defaults to
   ``[byte_length//new_itemsize]``, which means that the result view
   will be one-dimensional. The return value is a new memoryview, but
   the buffer itself is not copied. Supported casts are 1D -> C-:term:`contiguous`
   and C-contiguous -> 1D.

   The destination format is restricted to a single element native format in
   :mod:`struct` syntax. One of the formats must be a byte format
   ('B', 'b' or 'c'). The byte length of the result must be the same
   as the original length.
```

```
Cast 1D/long to 1D/unsigned bytes::

   >>> import array
   >>> a = array.array('l', [1,2,3])
   >>> x = memoryview(a)
   >>> x.format
   'l'
   >>> x.itemsize
   8
   >>> len(x)
   3
   >>> x.nbytes
   24
   >>> y = x.cast('B')
   >>> y.format
   'B'
   >>> y.itemsize
   1
   >>> len(y)
   24
   >>> y.nbytes
   24

Cast 1D/unsigned bytes to 1D/char::

   >>> b = bytearray(b'zyz')
   >>> x = memoryview(b)
   >>> x[0] = b'a'
   Traceback (most recent call last):
     File "<stdin>", line 1, in <module>
   ValueError: memoryview: invalid value for format "B"
   >>> y = x.cast('c')
   >>> y[0] = b'a'
   >>> b
   bytearray(b'ayz')

Cast 1D/bytes to 3D/ints to 1D/signed char::

   >>> import struct
   >>> buf = struct.pack("i"*12, *list(range(12)))
   >>> x = memoryview(buf)
   >>> y = x.cast('i', shape=[2,2,3])
   >>> y.tolist()
   [[[0, 1, 2], [3, 4, 5]], [[6, 7, 8], [9, 10, 11]]]
   >>> y.format
   'i'
   >>> y.itemsize
   4
   >>> len(y)
   2
   >>> y.nbytes
   48
   >>> z = y.cast('b')
   >>> z.format
   'b'
   >>> z.itemsize
   1
   >>> len(z)
   48
   >>> z.nbytes
   48

Cast 1D/unsigned long to 2D/unsigned long::

   >>> buf = struct.pack("L"*6, *list(range(6)))
   >>> x = memoryview(buf)
   >>> y = x.cast('L', shape=[2,3])
   >>> len(y)
   2
   >>> y.nbytes
   48
   >>> y.tolist()
   [[0, 1, 2], [3, 4, 5]]

.. versionadded:: 3.3

.. versionchanged:: 3.5
   The source format is no longer restricted when casting to a byte view.
```

There are also several readonly attributes available:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4038)

Unknown directive type "attribute".

```
.. attribute:: obj

   The underlying object of the memoryview::

      >>> b  = bytearray(b'xyz')
      >>> m = memoryview(b)
      >>> m.obj is b
      True

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4049)

Unknown directive type "attribute".

```
.. attribute:: nbytes

   ``nbytes == product(shape) * itemsize == len(m.tobytes())``. This is
   the amount of space in bytes that the array would use in a contiguous
   representation. It is not necessarily equal to ``len(m)``::

      >>> import array
      >>> a = array.array('i', [1,2,3,4,5])
      >>> m = memoryview(a)
      >>> len(m)
      5
      >>> m.nbytes
      20
      >>> y = m[::2]
      >>> len(y)
      3
      >>> y.nbytes
      12
      >>> len(y.tobytes())
      12

   Multi-dimensional arrays::

      >>> import struct
      >>> buf = struct.pack("d"*12, *[1.5*x for x in range(12)])
      >>> x = memoryview(buf)
      >>> y = x.cast('d', shape=[3,4])
      >>> y.tolist()
      [[0.0, 1.5, 3.0, 4.5], [6.0, 7.5, 9.0, 10.5], [12.0, 13.5, 15.0, 16.5]]
      >>> len(y)
      3
      >>> y.nbytes
      96

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4085)

Unknown directive type "attribute".

```
.. attribute:: readonly

   A bool indicating whether the memory is read only.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4089)

Unknown directive type "attribute".

```
.. attribute:: format

   A string containing the format (in :mod:`struct` module style) for each
   element in the view. A memoryview can be created from exporters with
   arbitrary format strings, but some methods (e.g. :meth:`tolist`) are
   restricted to native single element formats.
```

```
   .. versionchanged:: 3.3
      format ``'B'`` is now handled according to the struct module syntax.
      This means that ``memoryview(b'abc')[0] == b'abc'[0] == 97``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4100)

Unknown directive type "attribute".

```
   .. attribute:: itemsize

      The size in bytes of each element of the memoryview::

         >>> import array, struct
         >>> m = memoryview(array.array('H', [32000, 32001, 32002]))
         >>> m.itemsize
         2
         >>> m[0]
         32000
         >>> struct.calcsize('H') == m.itemsize
         True
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4113)

Unknown directive type "attribute".

```
   .. attribute:: ndim

      An integer indicating how many dimensions of a multi-dimensional array the
      memory represents.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4118)

Unknown directive type "attribute".

```
   .. attribute:: shape

      A tuple of integers the length of :attr:`ndim` giving the shape of the
      memory as an N-dimensional array.

      .. versionchanged:: 3.3
         An empty tuple instead of ``None`` when ndim = 0.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4126)

Unknown directive type "attribute".

```
   .. attribute:: strides

      A tuple of integers the length of :attr:`ndim` giving the size in bytes to
      access each element for each dimension of the array.

      .. versionchanged:: 3.3
         An empty tuple instead of ``None`` when ndim = 0.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4134)

Unknown directive type "attribute".

```
   .. attribute:: suboffsets

      Used internally for PIL-style arrays. The value is informational only.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4138)

Unknown directive type "attribute".

```
.. attribute:: c_contiguous

   A bool indicating whether the memory is C-:term:`contiguous`.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4144)

Unknown directive type "attribute".

```
.. attribute:: f_contiguous

   A bool indicating whether the memory is Fortran :term:`contiguous`.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4150)

Unknown directive type "attribute".

```
.. attribute:: contiguous

   A bool indicating whether the memory is :term:`contiguous`.

   .. versionadded:: 3.3
```

Set Types --- :class:`set`, :class:`frozenset`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4159); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4159); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4162)

Unknown directive type "index".

```
.. index:: object: set
```

A :dfn:`set` object is an unordered collection of distinct :term:`hashable` objects. Common uses include membership testing, removing duplicates from a sequence, and computing mathematical operations such as intersection, union, difference, and symmetric difference. (For other containers see the built-in :class:`dict`, :class:`list`, and :class:`tuple` classes, and the :mod:`collections` module.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4164); *backlink*

Unknown interpreted text role "mod".

Like other collections, sets support `x in set`, `len(set)`, and `for x in set`. Being an unordered collection, sets do not record element position or order of insertion. Accordingly, sets do not support indexing, slicing, or other sequence-like behavior.

There are currently two built-in set types, :class:`set` and :class:`frozenset`. The :class:`set` type is mutable --- the contents can be changed using methods like :meth:`~set.add` and :meth:`~set.remove`. Since it is mutable, it has no hash value and cannot be used as either a dictionary key or as an element of another set. The :class:`frozenset` type is immutable and :term:`hashable` --- its contents cannot be altered after it is created; it can therefore be used as a dictionary key or as an element of another set.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4176); *backlink*

Unknown interpreted text role "term".

Non-empty sets (not frozensets) can be created by placing a comma-separated list of elements within braces, for example: `{'jack', 'sjoerd'}`, in addition to the :class:`set` constructor.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4184); *backlink*

Unknown interpreted text role "class".

The constructors for both classes work the same:

Return a new set or frozenset object whose elements are taken from *iterable*. The elements of a set must be :term:`hashable`. To

represent sets of sets, the inner sets must be :class:`frozenset` objects. If *iterable* is not specified, a new empty set is returned.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4193); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4193); *backlink*

Unknown interpreted text role "class".

Sets can be created by several means:

- Use a comma-separated list of elements within braces: `{'jack', 'sjoerd'}`
- Use a set comprehension: `{c for c in 'abracadabra' if c not in 'abc'}`
- Use the type constructor: `set()`, `set('foobar')`, `set(['a', 'b', 'foo'])`

Instances of :class:`set` and :class:`frozenset` provide the following operations:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4205); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4205); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4208)

Unknown directive type "describe".

```
.. describe:: len(s)

   Return the number of elements in set *s* (cardinality of *s*).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4212)

Unknown directive type "describe".

```
.. describe:: x in s

   Test *x* for membership in *s*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4216)

Unknown directive type "describe".

```
.. describe:: x not in s

   Test *x* for non-membership in *s*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4220)

Unknown directive type "method".

```
.. method:: isdisjoint(other)

   Return ``True`` if the set has no elements in common with *other*.  Sets are
   disjoint if and only if their intersection is the empty set.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4225)

Unknown directive type "method".

```
.. method:: issubset(other)
            set <= other

   Test whether every element in the set is in *other*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4230)

Unknown directive type "method".

```
.. method:: set < other

   Test whether the set is a proper subset of *other*, that is,
   ``set <= other and set != other``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4235)

Unknown directive type "method".

```
.. method:: issuperset(other)
            set >= other

   Test whether every element in *other* is in the set.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4240)

Unknown directive type "method".

```
.. method:: set > other

   Test whether the set is a proper superset of *other*, that is, ``set >=
   other and set != other``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4245)

Unknown directive type "method".

```
.. method:: union(*others)
            set | other | ...

   Return a new set with elements from the set and all others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4250)

Unknown directive type "method".

```
.. method:: intersection(*others)
            set & other & ...

   Return a new set with elements common to the set and all others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4255)

Unknown directive type "method".

```
.. method:: difference(*others)
            set - other - ...

   Return a new set with elements in the set that are not in the others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4260)

Unknown directive type "method".

```
.. method:: symmetric_difference(other)
            set ^ other

   Return a new set with elements in either the set or *other* but not both.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4265)

Unknown directive type "method".

```
.. method:: copy()

   Return a shallow copy of the set.
```

Note, the non-operator versions of :meth:`union`, :meth:`intersection`, :meth:`difference`, :meth:`symmetric_difference`, :meth:`issubset`, and :meth:`issuperset` methods will accept any iterable as an argument. In contrast, their operator based counterparts require their arguments to be sets. This precludes error-prone constructions like `set('abc') & 'cbs'` in favor of the more readable `set('abc').intersection('cbs')`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4270); *backlink*

Unknown interpreted text role "meth".

Both :class:`set` and :class:`frozenset` support set to set comparisons. Two sets are equal if and only if every element of each set is contained in the other (each is a subset of the other). A set is less than another set if and only if the first set is a proper subset of the second set (is a subset, but is not equal). A set is greater than another set if and only if the first set is a proper superset of the second set (is a superset, but is not equal).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4277); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4277); *backlink*

Unknown interpreted text role "class".

Instances of :class:`set` are compared to instances of :class:`frozenset` based on their members. For example, `set('abc') == frozenset('abc')` returns `True` and so does `set('abc') in set([frozenset('abc')])`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4284); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4284); *backlink*

Unknown interpreted text role "class".

The subset and equality comparisons do not generalize to a total ordering function. For example, any two nonempty disjoint sets are not equal and are not subsets of each other, so *all* of the following return `False`: `a<b`, `a==b`, or `a>b`.

Since sets only define partial ordering (subset relationships), the output of the :meth:`list.sort` method is undefined for lists of sets.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4293); *backlink*

Unknown interpreted text role "meth".

Set elements, like dictionary keys, must be :term:`hashable`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4296); *backlink*

Unknown interpreted text role "term".

Binary operations that mix :class:`set` instances with :class:`frozenset` return the type of the first operand. For example: `frozenset('ab') | set('bc')` returns an instance of :class:`frozenset`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4298); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4298); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4298); *backlink*

Unknown interpreted text role "class".

The following table lists operations available for :class:`set` that do not apply to immutable instances of :class:`frozenset`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4302); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4302); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4305)

Unknown directive type "method".

```
.. method:: update(*others)
            set |= other | ...

   Update the set, adding elements from all others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4310)

Unknown directive type "method".

```
.. method:: intersection_update(*others)
            set &= other & ...

   Update the set, keeping only elements found in it and all others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4315)

Unknown directive type "method".

```
.. method:: difference_update(*others)
            set -= other | ...

   Update the set, removing elements found in others.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4320)

Unknown directive type "method".

```
.. method:: symmetric_difference_update(other)
            set ^= other

   Update the set, keeping only elements found in either set, but not in both.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4325)

Unknown directive type "method".

```
.. method:: add(elem)

   Add element *elem* to the set.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4329)

Unknown directive type "method".

```
.. method:: remove(elem)

   Remove element *elem* from the set.  Raises :exc:`KeyError` if *elem* is
   not contained in the set.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4334)

Unknown directive type "method".

```
.. method:: discard(elem)

   Remove element *elem* from the set if it is present.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4338)

Unknown directive type "method".

```
.. method:: pop()

   Remove and return an arbitrary element from the set.  Raises
   :exc:`KeyError` if the set is empty.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4343)

Unknown directive type "method".

```
.. method:: clear()

   Remove all elements from the set.
```

Note, the non-operator versions of the :meth:`update`, :meth:`intersection_update`, :meth:`difference_update`, and :meth:`symmetric_difference_update` methods will accept any iterable as an argument.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4348); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4348); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4348); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4348); *backlink*

Unknown interpreted text role "meth".

Note, the *elem* argument to the :meth:`__contains__`, :meth:`remove`, and :meth:`discard` methods may be a set. To support searching for an equivalent frozenset, a temporary one is created from *elem*.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4353); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4353); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4353); *backlink*

Unknown interpreted text role "meth".

Mapping Types --- :class:`dict`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4360); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]stdtypes.rst, line 4363)

Unknown directive type "index".

```
.. index::
   object: mapping
   object: dictionary
   triple: operations on; mapping; types
   triple: operations on; dictionary; type
```

```
statement: del
builtin: len
```

A :term:`mapping` object maps :term:`hashable` values to arbitrary objects. Mappings are mutable objects. There is currently only one standard mapping type, the :dfn:`dictionary`. (For other containers see the built-in :class:`list`, :class:`set`, and :class:`tuple` classes, and the :mod:`collections` module.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4371); *backlink*

Unknown interpreted text role "mod".

A dictionary's keys are *almost* arbitrary values. Values that are not :term:`hashable`, that is, values containing lists, dictionaries or other mutable types (that are compared by value rather than by object identity) may not be used as keys. Numeric types used for keys obey the normal rules for numeric comparison: if two numbers compare equal (such as `1` and `1.0`) then they can be used interchangeably to index the same dictionary entry. (Note however, that since computers store floating-point numbers as approximations it is usually unwise to use them as dictionary keys.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4377); *backlink*

Unknown interpreted text role "term".

Dictionaries can be created by placing a comma-separated list of `key: value` pairs within braces, for example: `{'jack': 4098, 'sjoerd': 4127}` or `{4098: 'jack', 4127: 'sjoerd'}`, or by the :class:`dict` constructor.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4386); *backlink*

Unknown interpreted text role "class".

Return a new dictionary initialized from an optional positional argument and a possibly empty set of keyword arguments.

Dictionaries can be created by several means:

- Use a comma-separated list of `key: value` pairs within braces: `{'jack': 4098, 'sjoerd': 4127}` or `{4098: 'jack', 4127: 'sjoerd'}`

- Use a dict comprehension: `{}`, `{x: x ** 2 for x in range(10)}`
- Use the type constructor: `dict()`, `dict([('foo', 100), ('bar', 200)])`, `dict(foo=100, bar=200)`

If no positional argument is given, an empty dictionary is created. If a positional argument is given and it is a mapping object, a dictionary is created with the same key-value pairs as the mapping object. Otherwise, the positional argument must be an :term:`iterable` object. Each item in the iterable must itself be an iterable with exactly two objects. The first object of each item becomes a key in the new dictionary, and the second object the corresponding value. If a key occurs more than once, the last value for that key becomes the corresponding value in the new dictionary.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4405); *backlink*

Unknown interpreted text role "term".

If keyword arguments are given, the keyword arguments and their values are added to the dictionary created from the positional argument. If a key being added is already present, the value from the keyword argument replaces the value from the positional argument.

To illustrate, the following examples all return a dictionary equal to `{"one": 1, "two": 2, "three": 3}`:

```
>>> a = dict(one=1, two=2, three=3)
>>> b = {'one': 1, 'two': 2, 'three': 3}
>>> c = dict(zip(['one', 'two', 'three'], [1, 2, 3]))
>>> d = dict([('two', 2), ('one', 1), ('three', 3)])
>>> e = dict({'three': 3, 'one': 1, 'two': 2})
>>> f = dict({'one': 1, 'three': 3}, two=2)
>>> a == b == c == d == e == f
True
```

Providing keyword arguments as in the first example only works for keys that are valid Python identifiers. Otherwise, any valid keys can be used.

These are the operations that dictionaries support (and therefore, custom mapping types should support too):

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4439)

Unknown directive type "describe".

```
.. describe:: list(d)

   Return a list of all the keys used in the dictionary *d*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4443)

Unknown directive type "describe".

```
.. describe:: len(d)

   Return the number of items in the dictionary *d*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4447)

Unknown directive type "describe".

```
.. describe:: d[key]

   Return the item of *d* with key *key*.  Raises a :exc:`KeyError` if *key* is
   not in the map.

   .. index:: __missing__()

   If a subclass of dict defines a method :meth:`__missing__` and *key*
   is not present, the ``d[key]`` operation calls that method with the key *key*
   as argument.  The ``d[key]`` operation then returns or raises whatever is
   returned or raised by the ``__missing__(key)`` call.
   No other operations or methods invoke :meth:`__missing__`. If
   :meth:`__missing__` is not defined, :exc:`KeyError` is raised.
   :meth:`__missing__` must be a method; it cannot be an instance variable::

       >>> class Counter(dict):
       ...     def __missing__(self, key):
       ...         return 0
       >>> c = Counter()
```

```
        >>> c['red']
        0
        >>> c['red'] += 1
        >>> c['red']
        1

    The example above shows part of the implementation of
    :class:`collections.Counter`.  A different ``__missing__`` method is used
    by :class:`collections.defaultdict`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4476)

Unknown directive type "describe".

```
.. describe:: d[key] = value

   Set ``d[key]`` to *value*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4480)

Unknown directive type "describe".

```
.. describe:: del d[key]

   Remove ``d[key]`` from *d*.  Raises a :exc:`KeyError` if *key* is not in the
   map.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4485)

Unknown directive type "describe".

```
.. describe:: key in d

   Return ``True`` if *d* has a key *key*, else ``False``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4489)

Unknown directive type "describe".

```
.. describe:: key not in d

   Equivalent to ``not key in d``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4493)

Unknown directive type "describe".

```
.. describe:: iter(d)

   Return an iterator over the keys of the dictionary.  This is a shortcut
   for ``iter(d.keys())``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4498)

Unknown directive type "method".

```
.. method:: clear()

   Remove all items from the dictionary.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4502)

Unknown directive type "method".

```
   .. method:: copy()

      Return a shallow copy of the dictionary.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4506)

Unknown directive type "classmethod".

```
   .. classmethod:: fromkeys(iterable[, value])

      Create a new dictionary with keys from *iterable* and values set to *value*.

      :meth:`fromkeys` is a class method that returns a new dictionary. *value*
      defaults to ``None``.  All of the values refer to just a single instance,
      so it generally doesn't make sense for *value* to be a mutable object
      such as an empty list.  To get distinct values, use a :ref:`dict
      comprehension <dict>` instead.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4516)

Unknown directive type "method".

```
   .. method:: get(key[, default])

      Return the value for *key* if *key* is in the dictionary, else *default*.
      If *default* is not given, it defaults to ``None``, so that this method
      never raises a :exc:`KeyError`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4522)

Unknown directive type "method".

```
   .. method:: items()

      Return a new view of the dictionary's items (``(key, value)`` pairs).
      See the :ref:`documentation of view objects <dict-views>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4527)

Unknown directive type "method".

```
   .. method:: keys()

      Return a new view of the dictionary's keys.  See the :ref:`documentation
      of view objects <dict-views>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4532)

Unknown directive type "method".

```
   .. method:: pop(key[, default])

      If *key* is in the dictionary, remove it and return its value, else return
      *default*.  If *default* is not given and *key* is not in the dictionary,
      a :exc:`KeyError` is raised.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4538)

Unknown directive type "method".

```
   .. method:: popitem()

      Remove and return a ``(key, value)`` pair from the dictionary.
      Pairs are returned in :abbr:`LIFO (last-in, first-out)` order.

      :meth:`popitem` is useful to destructively iterate over a dictionary, as
```

```
often used in set algorithms.  If the dictionary is empty, calling
:meth:`popitem` raises a :exc:`KeyError`.

.. versionchanged:: 3.7
   LIFO order is now guaranteed. In prior versions, :meth:`popitem` would
   return an arbitrary key/value pair.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4551)

Unknown directive type "describe".

```
.. describe:: reversed(d)

   Return a reverse iterator over the keys of the dictionary. This is a
   shortcut for ``reversed(d.keys())``.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4558)

Unknown directive type "method".

```
.. method:: setdefault(key[, default])

   If *key* is in the dictionary, return its value.  If not, insert *key*
   with a value of *default* and return *default*.  *default* defaults to
   ``None``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4564)

Unknown directive type "method".

```
.. method:: update([other])

   Update the dictionary with the key/value pairs from *other*, overwriting
   existing keys.  Return ``None``.

   :meth:`update` accepts either another dictionary object or an iterable of
   key/value pairs (as tuples or other iterables of length two).  If keyword
   arguments are specified, the dictionary is then updated with those
   key/value pairs: ``d.update(red=1, blue=2)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4574)

Unknown directive type "method".

```
.. method:: values()

   Return a new view of the dictionary's values.  See the
   :ref:`documentation of view objects <dict-views>`.

   An equality comparison between one ``dict.values()`` view and another
   will always return ``False``. This also applies when comparing
   ``dict.values()`` to itself::

      >>> d = {'a': 1}
      >>> d.values() == d.values()
      False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4587)

Unknown directive type "describe".

```
.. describe:: d | other

   Create a new dictionary with the merged keys and values of *d* and
   *other*, which must both be dictionaries. The values of *other* take
   priority when *d* and *other* share keys.
```

```
.. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4595)

Unknown directive type "describe".

```
.. describe:: d |= other

   Update the dictionary *d* with keys and values from *other*, which may be
   either a :term:`mapping` or an :term:`iterable` of key/value pairs. The
   values of *other* take priority when *d* and *other* share keys.

   .. versionadded:: 3.9
```

Dictionaries compare equal if and only if they have the same `(key, value)` pairs (regardless of ordering). Order comparisons ('<', '<=', '>=', '>') raise :exc:`TypeError`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4603); *backlink*

Unknown interpreted text role "exc".

Dictionaries preserve insertion order. Note that updating a key does not affect the order. Keys added after deletion are inserted at the end.

```
>>> d = {"one": 1, "two": 2, "three": 3, "four": 4}
>>> d
{'one': 1, 'two': 2, 'three': 3, 'four': 4}
>>> list(d)
['one', 'two', 'three', 'four']
>>> list(d.values())
[1, 2, 3, 4]
>>> d["one"] = 42
>>> d
{'one': 42, 'two': 2, 'three': 3, 'four': 4}
>>> del d["two"]
>>> d["two"] = None
>>> d
{'one': 42, 'three': 3, 'four': 4, 'two': None}
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4625)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.7
   Dictionary order is guaranteed to be insertion order.  This behavior was
   an implementation detail of CPython from 3.6.
```

Dictionaries and dictionary views are reversible.

```
>>> d = {"one": 1, "two": 2, "three": 3, "four": 4}
>>> d
{'one': 1, 'two': 2, 'three': 3, 'four': 4}
>>> list(reversed(d))
['four', 'three', 'two', 'one']
>>> list(reversed(d.values()))
[4, 3, 2, 1]
>>> list(reversed(d.items()))
[('four', 4), ('three', 3), ('two', 2), ('one', 1)]
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4641)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8
   Dictionaries are now reversible.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4645)

Unknown directive type "seealso".

```
.. seealso::
   :class:`types.MappingProxyType` can be used to create a read-only view
   of a :class:`dict`.
```

Dictionary view objects

The objects returned by :meth:`dict.keys`, :meth:`dict.values` and :meth:`dict.items` are *view objects*. They provide a dynamic view on the dictionary's entries, which means that when the dictionary changes, the view reflects these changes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4655); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4655); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4655); *backlink*

Unknown interpreted text role "meth".

Dictionary views can be iterated over to yield their respective data, and support membership tests:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4663)

Unknown directive type "describe".

```
.. describe:: len(dictview)

   Return the number of entries in the dictionary.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4667)

Unknown directive type "describe".

```
.. describe:: iter(dictview)

   Return an iterator over the keys, values or items (represented as tuples of
   ``(key, value)``) in the dictionary.

   Keys and values are iterated over in insertion order.
   This allows the creation of ``(value, key)`` pairs
   using :func:`zip`: ``pairs = zip(d.values(), d.keys())``.  Another way to
   create the same list is ``pairs = [(v, k) for (k, v) in d.items()]``.

   Iterating views while adding or deleting entries in the dictionary may raise
   a :exc:`RuntimeError` or fail to iterate over all entries.

   .. versionchanged:: 3.7
      Dictionary order is guaranteed to be insertion order.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4683)

Unknown directive type "describe".

```
.. describe:: x in dictview

   Return ``True`` if *x* is in the underlying dictionary's keys, values or
   items (in the latter case, *x* should be a ``(key, value)`` tuple).
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4688)

Unknown directive type "describe".

```
.. describe:: reversed(dictview)

   Return a reverse iterator over the keys, values or items of the dictionary.
   The view will be iterated in reverse order of the insertion.

   .. versionchanged:: 3.8
      Dictionary views are now reversible.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4696)

Unknown directive type "describe".

```
.. describe:: dictview.mapping

   Return a :class:`types.MappingProxyType` that wraps the original
   dictionary to which the view refers.

   .. versionadded:: 3.10
```

Keys views are set-like since their entries are unique and hashable. If all values are hashable, so that `(key, value)` pairs are unique and hashable, then the items view is also set-like. (Values views are not treated as set-like since the entries are generally not unique.) For set-like views, all of the operations defined for the abstract base class :class:`collections.abc.Set` are available (for example, ==, <, or ^).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4703); *backlink*

Unknown interpreted text role "class".

An example of dictionary view usage:

```
>>> dishes = {'eggs': 2, 'sausage': 1, 'bacon': 1, 'spam': 500}
>>> keys = dishes.keys()
>>> values = dishes.values()

>>> # iteration
>>> n = 0
>>> for val in values:
...     n += val
>>> print(n)
504

>>> # keys and values are iterated over in the same order (insertion order)
>>> list(keys)
['eggs', 'sausage', 'bacon', 'spam']
>>> list(values)
[2, 1, 1, 500]

>>> # view objects are dynamic and reflect dict changes
>>> del dishes['eggs']
>>> del dishes['sausage']
>>> list(keys)
['bacon', 'spam']

>>> # set operations
>>> keys & {'eggs', 'bacon', 'salad'}
{'bacon'}
>>> keys ^ {'sausage', 'juice'}
{'juice', 'sausage', 'bacon', 'spam'}

>>> # get back a read-only proxy for the original dictionary
>>> values.mapping
mappingproxy({'eggs': 2, 'sausage': 1, 'bacon': 1, 'spam': 500})
>>> values.mapping['spam']
500
```

Context Manager Types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4753)

Unknown directive type "index".

```
.. index::
   single: context manager
   single: context management protocol
   single: protocol; context management
```

Python's :keyword:`with` statement supports the concept of a runtime context defined by a context manager. This is implemented using a pair of methods that allow user-defined classes to define a runtime context that is entered before the statement body is executed and exited when the statement ends:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4758); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4764)

Unknown directive type "method".

```
.. method:: contextmanager.__enter__()

   Enter the runtime context and return either this object or another object
   related to the runtime context. The value returned by this method is bound to
   the identifier in the :keyword:`!as` clause of :keyword:`with` statements using
   this context manager.

   An example of a context manager that returns itself is a :term:`file object`.
   File objects return themselves from __enter__() to allow :func:`open` to be
   used as the context expression in a :keyword:`with` statement.

   An example of a context manager that returns a related object is the one
   returned by :func:`decimal.localcontext`. These managers set the active
   decimal context to a copy of the original decimal context and then return the
   copy. This allows changes to be made to the current decimal context in the body
   of the :keyword:`with` statement without affecting code outside the
   :keyword:`!with` statement.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4783)

Unknown directive type "method".

```
.. method:: contextmanager.__exit__(exc_type, exc_val, exc_tb)

   Exit the runtime context and return a Boolean flag indicating if any exception
   that occurred should be suppressed. If an exception occurred while executing the
   body of the :keyword:`with` statement, the arguments contain the exception type,
   value and traceback information. Otherwise, all three arguments are ``None``.

   Returning a true value from this method will cause the :keyword:`with` statement
   to suppress the exception and continue execution with the statement immediately
   following the :keyword:`!with` statement. Otherwise the exception continues
   propagating after this method has finished executing. Exceptions that occur
   during execution of this method will replace any exception that occurred in the
   body of the :keyword:`!with` statement.

   The exception passed in should never be reraised explicitly - instead, this
   method should return a false value to indicate that the method completed
   successfully and does not want to suppress the raised exception. This allows
   context management code to easily detect whether or not an :meth:`__exit__`
   method has actually failed.
```

Python defines several context managers to support easy thread synchronisation, prompt closure of files or other objects, and simpler manipulation of the active decimal arithmetic context. The specific types are not treated specially beyond their implementation of the context management protocol. See the :mod:`contextlib` module for some examples.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4803); *backlink*

Unknown interpreted text role "mod".

Python's :term:`generator`s and the :class:`contextlib.contextmanager` decorator provide a convenient way to implement these protocols. If a generator function is decorated with the :class:`contextlib.contextmanager` decorator, it will return a context manager implementing the necessary :meth:`~contextmanager.__enter__` and :meth:`~contextmanager.__exit__` methods, rather than the iterator produced by an undecorated generator function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4809); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4809); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4809); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4809); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4809); *backlink*

Unknown interpreted text role "meth".

Note that there is no specific slot for any of these methods in the type structure for Python objects in the Python/C API. Extension types wanting to define these methods must provide them as a normal Python accessible method. Compared to the overhead of setting up the runtime context, the overhead of a single class dictionary lookup is negligible.

Type Annotation Types --- :ref:`Generic Alias <types-genericalias>`, :ref:`Union <types-union>`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4823); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4823); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4826)

Unknown directive type "index".

```
.. index::
   single: annotation; type annotation; type hint
```

The core built-in types for :term:`type annotations <annotation>` are :ref:`Generic Alias <types-genericalias>` and :ref:`Union <types-union>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4829); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4829); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4829); *backlink*

Unknown interpreted text role "ref".

Generic Alias Type

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4838)

Unknown directive type "index".

```
.. index::
   object: GenericAlias
   pair: Generic; Alias
```

`GenericAlias` objects are generally created by :ref:`subscripting <subscriptions>` a class. They are most often used with :ref:`container classes <sequence-types>`, such as :class:`list` or :class:`dict`. For example, `list[int]` is a `GenericAlias` object created by subscripting the `list` class with the argument :class:`int`. `GenericAlias` objects are intended primarily for use with :term:`type annotations <annotation>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4842); *backlink*

Unknown interpreted text role "term".

Note

It is generally only possible to subscript a class if the class implements the special method :meth:`~object.__class_getitem__`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4852); *backlink*

Unknown interpreted text role "meth".

A `GenericAlias` object acts as a proxy for a :term:`generic type`, implementing *parameterized generics*.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4855); *backlink*

Unknown interpreted text role "term".

For a container class, the argument(s) supplied to a :ref:`subscription <subscriptions>` of the class may indicate the type(s) of the elements an object contains. For example, `set[bytes]` can be used in type annotations to signify a :class:`set` in which all the elements are of type :class:`bytes`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4858); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4858); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4858); *backlink*

Unknown interpreted text role "class".

For a class which defines :meth:`~object.__class_getitem__` but is not a container, the argument(s) supplied to a subscription of the class will often indicate the return type(s) of one or more methods defined on an object. For example, :mod:`regular expressions <re>` can be used on both the :class:`str` data type and the :class:`bytes` data type:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4864); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4864); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4864); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4864); *backlink*

Unknown interpreted text role "class".

- If `x = re.search('foo', 'foo')`, `x` will be a :ref:`re.Match <match-objects>` object where the return values of `x.group(0)` and `x[0]` will both be of type :class:`str`. We can represent this kind of object in type annotations with the `GenericAlias` `re.Match[str]`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4870); *backlink*

  Unknown interpreted text role "ref".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4870); *backlink*

  Unknown interpreted text role "class".

- If `y = re.search(b'bar', b'bar')`, (note the `b` for :class:`bytes`), `y` will also be an instance of `re.Match`, but the return values of `y.group(0)` and `y[0]` will both be of type :class:`bytes`. In type annotations, we would represent this variety of :ref:`re.Match <match-objects>` objects with `re.Match[bytes]`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4876); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4876); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4876); *backlink*

Unknown interpreted text role "ref".

`GenericAlias` objects are instances of the class :class:`types.GenericAlias`, which can also be used to create `GenericAlias` objects directly.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4882); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4886)

Unknown directive type "describe".

```
.. describe:: T[X, Y, ...]

   Creates a ``GenericAlias`` representing a type ``T`` parameterized by types
   *X*, *Y*, and more depending on the ``T`` used.
   For example, a function expecting a :class:`list` containing
   :class:`float` elements::

      def average(values: list[float]) -> float:
          return sum(values) / len(values)

   Another example for :term:`mapping` objects, using a :class:`dict`, which
   is a generic type expecting two type parameters representing the key type
   and the value type.  In this example, the function expects a ``dict`` with
   keys of type :class:`str` and values of type :class:`int`::

      def send_post_request(url: str, body: dict[str, int]) -> None:
          ...
```

The builtin functions :func:`isinstance` and :func:`issubclass` do not accept `GenericAlias` types for their second argument:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4904); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4904); *backlink*

Unknown interpreted text role "func".

```
>>> isinstance([1, 2], list[str])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: isinstance() argument 2 cannot be a parameterized generic
```

The Python runtime does not enforce :term:`type annotations <annotation>`. This extends to generic types and their type parameters. When creating a container object from a `GenericAlias`, the elements in the container are not checked against their type. For example, the following code is discouraged, but will run without errors:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4912); *backlink*

Unknown interpreted text role "term".

```
>>> t = list[str]
>>> t([1, 2, 3])
[1, 2, 3]
```

Furthermore, parameterized generics erase type parameters during object creation:

```
>>> t = list[str]
>>> type(t)
<class 'types.GenericAlias'>

>>> l = t()
>>> type(l)
<class 'list'>
```

Calling :func:`repr` or :func:`str` on a generic shows the parameterized type:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4933); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4933); *backlink*

Unknown interpreted text role "func".

```
>>> repr(list[int])
'list[int]'

>>> str(list[int])
'list[int]'
```

The :meth:`~object.__getitem__` method of generic containers will raise an exception to disallow mistakes like `dict[str][str]`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4941); *backlink*

Unknown interpreted text role "meth".

```
>>> dict[str][str]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: There are no type variables left in dict[str]
```

However, such expressions are valid when :ref:`type variables <generics>` are used. The index must have as many elements as there are type variable items in the `GenericAlias` object's :attr:`~genericalias.__args__`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4949); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4949); *backlink*

Unknown interpreted text role "attr".

```
>>> from typing import TypeVar
>>> Y = TypeVar('Y')
>>> dict[str, Y][int]
dict[str, int]
```

Standard Generic Classes

The following standard library classes support parameterized generics. This list is non-exhaustive.

- :class:`tuple`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4965); *backlink*

Unknown interpreted text role "class".

- :class:`list`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4966); *backlink*

Unknown interpreted text role "class".

- :class:`dict`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4967); *backlink*

Unknown interpreted text role "class".

- :class:`set`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4968); *backlink*

Unknown interpreted text role "class".

- :class:`frozenset`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4969); *backlink*

Unknown interpreted text role "class".

- :class:`type`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4970); *backlink*

Unknown interpreted text role "class".

- :class:`collections.deque`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4971); *backlink*

Unknown interpreted text role "class".

- :class:`collections.defaultdict`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4972); *backlink*

Unknown interpreted text role "class".

- :class:`collections.OrderedDict`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4973); *backlink*

Unknown interpreted text role "class".

- :class:`collections.Counter`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4974); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.ChainMap`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4975); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.Awaitable`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4976); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.Coroutine`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4977); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.AsyncIterable`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4978); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.AsyncIterator`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4979); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.AsyncGenerator`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4980); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.Iterable`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4981); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.Iterator`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4982); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Generator`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4983); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Reversible`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4984); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Container`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4985); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Collection`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4986); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Callable`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4987); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Set`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4988); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.MutableSet`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4989); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.Mapping`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4990); *backlink*

Unknown interpreted text role "class".

- :class:`collections.abc.MutableMapping`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4991); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.Sequence`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4992); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.MutableSequence`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4993); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.ByteString`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4994); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.MappingView`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4995); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.KeysView`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4996); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.ItemsView`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4997); *backlink*

  Unknown interpreted text role "class".

- :class:`collections.abc.ValuesView`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4998); *backlink*

  Unknown interpreted text role "class".

- :class:`contextlib.AbstractContextManager`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 4999); *backlink*

Unknown interpreted text role "class".

- :class:`contextlib.AbstractAsyncContextManager`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5000); *backlink*

Unknown interpreted text role "class".

- :class:`dataclasses.Field`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5001); *backlink*

Unknown interpreted text role "class".

- :class:`functools.cached_property`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5002); *backlink*

Unknown interpreted text role "class".

- :class:`functools.partialmethod`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5003); *backlink*

Unknown interpreted text role "class".

- :class:`os.PathLike`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5004); *backlink*

Unknown interpreted text role "class".

- :class:`queue.LifoQueue`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5005); *backlink*

Unknown interpreted text role "class".

- :class:`queue.Queue`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5006); *backlink*

Unknown interpreted text role "class".

- :class:`queue.PriorityQueue`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5007); *backlink*

Unknown interpreted text role "class".

- :class:`queue.SimpleQueue`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5008); *backlink*

  Unknown interpreted text role "class".

- :ref:`re.Pattern <re-objects>`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5009); *backlink*

  Unknown interpreted text role "ref".

- :ref:`re.Match <match-objects>`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5010); *backlink*

  Unknown interpreted text role "ref".

- :class:`shelve.BsdDbShelf`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5011); *backlink*

  Unknown interpreted text role "class".

- :class:`shelve.DbfilenameShelf`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5012); *backlink*

  Unknown interpreted text role "class".

- :class:`shelve.Shelf`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5013); *backlink*

  Unknown interpreted text role "class".

- :class:`types.MappingProxyType`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5014); *backlink*

  Unknown interpreted text role "class".

- :class:`weakref.WeakKeyDictionary`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5015); *backlink*

  Unknown interpreted text role "class".

- :class:`weakref.WeakMethod`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5016); *backlink*

Unknown interpreted text role "class".

- :class:`weakref.WeakSet`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5017); *backlink*

Unknown interpreted text role "class".

- :class:`weakref.WeakValueDictionary`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5018); *backlink*

Unknown interpreted text role "class".

Special Attributes of `GenericAlias` objects

All parameterized generics implement special read-only attributes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5027)

Unknown directive type "attribute".

```
.. attribute:: genericalias.__origin__

   This attribute points at the non-parameterized generic class::

      >>> list[int].__origin__
      <class 'list'>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5035)

Unknown directive type "attribute".

```
.. attribute:: genericalias.__args__

   This attribute is a :class:`tuple` (possibly of length 1) of generic
   types passed to the original :meth:`~object.__class_getitem__` of the
   generic class::

      >>> dict[str, list[int]].__args__
      (<class 'str'>, list[int])
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5045)

Unknown directive type "attribute".

```
.. attribute:: genericalias.__parameters__

   This attribute is a lazily computed tuple (possibly empty) of unique type
   variables found in ``__args__``::

      >>> from typing import TypeVar

      >>> T = TypeVar('T')
      >>> list[T].__parameters__
      (~T,)


   .. note::
      A ``GenericAlias`` object with :class:`typing.ParamSpec` parameters may not
      have correct ``__parameters__`` after substitution because
      :class:`typing.ParamSpec` is intended primarily for static type checking.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5062)

Unknown directive type "seealso".

```
.. seealso::

   :pep:`484` - Type Hints
      Introducing Python's framework for type annotations.

   :pep:`585` - Type Hinting Generics In Standard Collections
      Introducing the ability to natively parameterize standard-library
      classes, provided they implement the special class method
      :meth:`~object.__class_getitem__`.

   :ref:`Generics`, :ref:`user-defined generics <user-defined-generics>` and :class:`typing.Generic
      Documentation on how to implement generic classes that can be
      parameterized at runtime and understood by static type-checkers.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5076)

Unknown directive type "versionadded".

```
.. versionadded:: 3.9
```

Union Type

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5084)

Unknown directive type "index".

```
.. index::
   object: Union
   pair: union; type
```

A union object holds the value of the `|` (bitwise or) operation on multiple :ref:`type objects <bltin-type-objects>`. These types are intended primarily for :term:`type annotations <annotation>`. The union type expression enables cleaner type hinting syntax compared to :data:`typing.Union`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5088); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5088); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5088); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5093)

Unknown directive type "describe".

```
.. describe:: X | Y | ...

   Defines a union object which holds types *X*, *Y*, and so forth. ``X | Y``
   means either X or Y.  It is equivalent to ``typing.Union[X, Y]``.
   For example, the following function expects an argument of type
   :class:`int` or :class:`float`::

      def square(number: int | float) -> int | float:
```

```
            return number ** 2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5103)

Unknown directive type "describe".

```
.. describe:: union_object == other

   Union objects can be tested for equality with other union objects.  Details:

   * Unions of unions are flattened::

       (int | str) | float == int | str | float

   * Redundant types are removed::

       int | str | int == int | str

   * When comparing unions, the order is ignored::

      int | str == str | int

   * It is compatible with :data:`typing.Union`::

      int | str == typing.Union[int, str]

   * Optional types can be spelled as a union with ``None``::

      str | None == typing.Optional[str]
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5127)

Unknown directive type "describe".

```
.. describe:: isinstance(obj, union_object)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5128)

Unknown directive type "describe".

```
.. describe:: issubclass(obj, union_object)

   Calls to :func:`isinstance` and :func:`issubclass` are also supported with a
   union object::

      >>> isinstance("", int | str)
      True

   However, union objects containing :ref:`parameterized generics
   <types-genericalias>` cannot be used::

      >>> isinstance(1, int | list[int])
      Traceback (most recent call last):
        File "<stdin>", line 1, in <module>
      TypeError: isinstance() argument 2 cannot contain a parameterized generic
```

The user-exposed type for the union object can be accessed from :data:`types.UnionType` and used for :func:`isinstance` checks. An object cannot be instantiated from the type:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5144); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5144); *backlink*

Unknown interpreted text role "func".

```
>>> import types
```

```
>>> isinstance(int | str, types.UnionType)
True
>>> types.UnionType()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: cannot create 'types.UnionType' instances
```

Note

The :meth:`__or__` method for type objects was added to support the syntax `X | Y`. If a metaclass implements :meth:`__or__`, the Union may override it:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5157); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5157); *backlink*

Unknown interpreted text role "meth".

```
>>> class M(type):
...     def __or__(self, other):
...         return "Hello"
...
>>> class C(metaclass=M):
...     pass
...
>>> C | int
'Hello'
>>> int | C
int | __main__.C
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5173)

Unknown directive type "seealso".

```
.. seealso::

   :pep:`604` -- PEP proposing the ``X | Y`` syntax and the Union type.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5177)

Unknown directive type "versionadded".

```
.. versionadded:: 3.10
```

Other Built-in Types

The interpreter supports several other kinds of objects. Most of these support only one or two operations.

Modules

The only special operation on a module is attribute access: `m.name`, where *m* is a module and *name* accesses a name defined in *m*'s symbol table. Module attributes can be assigned to. (Note that the :keyword:`import` statement is not, strictly speaking, an operation on a module object; `import foo` does not require a module object named *foo* to exist, rather it requires an (external) *definition* for a module named *foo* somewhere.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5194); *backlink*

Unknown interpreted text role "keyword".

A special attribute of every module is :attr:`~object.__dict__`. This is the dictionary containing the module's symbol table. Modifying

this dictionary will actually change the module's symbol table, but direct assignment to the :attr:`~object.__dict__` attribute is not possible (you can write `m.__dict__['a'] = 1`, which defines `m.a` to be `1`, but you can't write `m.__dict__ = {}`). Modifying :attr:`~object.__dict__` directly is not recommended.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5201); *backlink***
>
> Unknown interpreted text role "attr".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5201); *backlink***
>
> Unknown interpreted text role "attr".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5201); *backlink***
>
> Unknown interpreted text role "attr".

Modules built into the interpreter are written like this: `<module 'sys' (built-in)>`. If loaded from a file, they are written as `<module 'os' from '/usr/local/lib/pythonX.Y/os.pyc'>`.

Classes and Class Instances

See :ref:`objects` and :ref:`class` for these.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5219); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5219); *backlink***
>
> Unknown interpreted text role "ref".

Functions

Function objects are created by function definitions. The only operation on a function object is to call it: `func(argument-list)`.

There are really two flavors of function objects: built-in functions and user-defined functions. Both support the same operation (to call the function), but the implementation is different, hence the different object types.

See :ref:`function` for more information.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5234); *backlink***
>
> Unknown interpreted text role "ref".

Methods

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5242)**
>
> Unknown directive type "index".
>
> ```
> .. index:: object: method
> ```

Methods are functions that are called using the attribute notation. There are two flavors: built-in methods (such as :meth:`append` on lists) and class instance methods. Built-in methods are described with the types that support them.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5244); *backlink***
>
> Unknown interpreted text role "meth".

If you access a method (a function defined in a class namespace) through an instance, you get a special object: a :dfn:`bound method` (also called :dfn:`instance method`) object. When called, it will add the `self` argument to the argument list. Bound methods have two

special read-only attributes: `m.__self__` is the object on which the method operates, and `m.__func__` is the function implementing the method. Calling `m(arg-1, arg-2, ..., arg-n)` is completely equivalent to calling `m.__func__(m.__self__, arg-1, arg-2, ..., arg-n)`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5249); *backlink*

Unknown interpreted text role "dfn".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5249); *backlink*

Unknown interpreted text role "dfn".

Like function objects, bound method objects support getting arbitrary attributes. However, since method attributes are actually stored on the underlying function object (`meth.__func__`), setting method attributes on bound methods is disallowed. Attempting to set an attribute on a method results in an :exc:`AttributeError` being raised. In order to set a method attribute, you need to explicitly set it on the underlying function object:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5258); *backlink*

Unknown interpreted text role "exc".

```
>>> class C:
...     def method(self):
...         pass
...
>>> c = C()
>>> c.method.whoami = 'my name is method'  # can't set on the method
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'method' object has no attribute 'whoami'
>>> c.method.__func__.whoami = 'my name is method'
>>> c.method.whoami
'my name is method'
```

See :ref:`types` for more information.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5278); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5281)

Unknown directive type "index".

```
.. index:: object; code, code object
```

Code Objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5288)

Unknown directive type "index".

```
.. index::
   builtin: compile
   single: __code__ (function object attribute)
```

Code objects are used by the implementation to represent "pseudo-compiled" executable Python code such as a function body. They differ from function objects because they don't contain a reference to their global execution environment. Code objects are returned by the built-in :func:`compile` function and can be extracted from function objects through their :attr:`__code__` attribute. See also the :mod:`code` module.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5292); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5292); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5292); *backlink*

Unknown interpreted text role "mod".

Accessing `__code__` raises an :ref:`auditing event <auditing>` `object.__getattr__` with arguments `obj` and `"__code__"`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5299); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5302)

Unknown directive type "index".

```
.. index::
   builtin: exec
   builtin: eval
```

A code object can be executed or evaluated by passing it (instead of a source string) to the :func:`exec` or :func:`eval` built-in functions.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5306); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5306); *backlink*

Unknown interpreted text role "func".

See :ref:`types` for more information.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5309); *backlink*

Unknown interpreted text role "ref".

Type Objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5317)

Unknown directive type "index".

```
.. index::
   builtin: type
   module: types
```

Type objects represent the various object types. An object's type is accessed by the built-in function :func:`type`. There are no special operations on types. The standard module :mod:`types` defines names for all standard built-in types.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5321); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5321); *backlink*

Unknown interpreted text role "mod".

Types are written like this: `<class 'int'>`.

The Null Object

This object is returned by functions that don't explicitly return a value. It supports no special operations. There is exactly one null object, named `None` (a built-in name). `type(None)()` produces the same singleton.

It is written as `None`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5341)

Unknown directive type "index".

```
.. index:: single: ...; ellipsis literal
```

The Ellipsis Object

This object is commonly used by slicing (see :ref:`slicings`). It supports no special operations. There is exactly one ellipsis object, named :const:`Ellipsis` (a built-in name). `type(Ellipsis)()` produces the :const:`Ellipsis` singleton.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5347); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5347); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5347); *backlink*

Unknown interpreted text role "const".

It is written as `Ellipsis` or `...`.

The NotImplemented Object

This object is returned from comparisons and binary operations when they are asked to operate on types they don't support. See :ref:`comparisons` for more information. There is exactly one `NotImplemented` object. `type(NotImplemented)()` produces the singleton instance.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5360); *backlink*

Unknown interpreted text role "ref".

It is written as `NotImplemented`.

Boolean Values

Boolean values are the two constant objects `False` and `True`. They are used to represent truth values (although other values can also be considered false or true). In numeric contexts (for example when used as the argument to an arithmetic operator), they behave like the integers 0 and 1, respectively. The built-in function :func:`bool` can be used to convert any value to a Boolean, if the value can be interpreted as a truth value (see section :ref:`truth` above).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5373); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5373); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5381)

Unknown directive type "index".

```
.. index::
   single: False
   single: True
   pair: Boolean; values
```

They are written as `False` and `True`, respectively.

Internal Objects

See :ref:`types` for this information. It describes stack frame objects, traceback objects, and slice objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5394); *backlink*

Unknown interpreted text role "ref".

Special Attributes

The implementation adds a few special read-only attributes to several object types, where they are relevant. Some of these are not reported by the :func:`dir` built-in function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5403); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5408)

Unknown directive type "attribute".

```
.. attribute:: object.__dict__

   A dictionary or other mapping object used to store an object's (writable)
   attributes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5414)

Unknown directive type "attribute".

```
.. attribute:: instance.__class__

   The class to which a class instance belongs.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5419)

Unknown directive type "attribute".

```
.. attribute:: class.__bases__

   The tuple of base classes of a class object.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5424)

Unknown directive type "attribute".

```
.. attribute:: definition.__name__

   The name of the class, function, method, descriptor, or
   generator instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5430)

Unknown directive type "attribute".

```
.. attribute:: definition.__qualname__

   The :term:`qualified name` of the class, function, method, descriptor,
   or generator instance.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5438)

Unknown directive type "attribute".

```
.. attribute:: class.__mro__

   This attribute is a tuple of classes that are considered when looking for
   base classes during method resolution.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5444)

Unknown directive type "method".

```
.. method:: class.mro()

   This method can be overridden by a metaclass to customize the method
   resolution order for its instances.  It is called at class instantiation, and
   its result is stored in :attr:`~class.__mro__`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5451)

Unknown directive type "method".

```
.. method:: class.__subclasses__

   Each class keeps a list of weak references to its immediate subclasses.  This
   method returns a list of all those references still alive.  The list is in
   definition order.  Example::

      >>> int.__subclasses__()
      [<class 'bool'>]
```

Footnotes

[1] Additional information on these special methods may be found in the Python Reference Manual (:ref:`customization`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]stdtypes.rst, line 5463); *backlink*

Unknown interpreted text role "ref".

[2] As a consequence, the list `[1, 2]` is considered equal to `[1.0, 2.0]`, and similarly for tuples.

[3] They must have since the parser can't tell the type of the operands.

[4] Cased characters are those with general category property being one of "Lu" (Letter, uppercase), "Ll" (Letter, lowercase),

or "Lt" (Letter, titlecase).

[5] (*1*,*2*) To format only a tuple you should therefore provide a singleton tuple whose only element is the tuple to be formatted.