

## :mod:`\_\_future\_\_` --- Future statement definitions

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 4)

Unknown directive type "module".

```
.. module:: __future__
   :synopsis: Future statement definitions
```

**Source code:** :source:`Lib/\_\_future\_\_.py`

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 7); [backlink](#)

Unknown interpreted text role "source".

:mod:`\_\_future\_\_` is a real module, and serves three purposes:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 11); [backlink](#)

Unknown interpreted text role "mod".

- To avoid confusing existing tools that analyze import statements and expect to find the modules they're importing.
- To ensure that :ref:`future statements <future>` run under releases prior to 2.1 at least yield runtime exceptions (the import of :mod:`\_\_future\_\_` will fail, because there was no module of that name prior to 2.1).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 16); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 16); [backlink](#)

Unknown interpreted text role "mod".

- To document when incompatible changes were introduced, and when they will be --- or were --- made mandatory. This is a form of executable documentation, and can be inspected programmatically via importing :mod:`\_\_future\_\_` and examining its contents.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 20); [backlink](#)

Unknown interpreted text role "mod".

Each statement in :file:`\_\_future\_\_.py` is of the form

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 25); [backlink](#)

Unknown interpreted text role "file".

```
FeatureName = _Feature(OptionalRelease, MandatoryRelease,  
                        CompilerFlag)
```

where, normally, *OptionalRelease* is less than *MandatoryRelease*, and both are 5-tuples of the same form as `:data:'sys.version_info'`:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 31); [backlink](#)

Unknown interpreted text role "data".

```
(PY_MAJOR_VERSION, # the 2 in 2.1.0a3; an int  
PY_MINOR_VERSION, # the 1; an int  
PY_MICRO_VERSION, # the 0; an int  
PY_RELEASE_LEVEL, # "alpha", "beta", "candidate" or "final"; string  
PY_RELEASE_SERIAL # the 3; an int  
)
```

*OptionalRelease* records the first release in which the feature was accepted.

In the case of a *MandatoryRelease* that has not yet occurred, *MandatoryRelease* predicts the release in which the feature will become part of the language.

Else *MandatoryRelease* records when the feature became part of the language; in releases at or after that, modules no longer need a future statement to use the feature in question, but may continue to use such imports.

*MandatoryRelease* may also be `None`, meaning that a planned feature got dropped.

Instances of class `:class:'_Feature'` have two corresponding methods, `:meth:'getOptionalRelease'` and `:meth:'getMandatoryRelease'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 54); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 54); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 54); [backlink](#)

Unknown interpreted text role "meth".

*CompilerFlag* is the (bitfield) flag that should be passed in the fourth argument to the built-in function `:func:'compile'` to enable the feature in dynamically compiled code. This flag is stored in the `:attr:'compiler_flag'` attribute on `:class:'_Feature'` instances.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 57); [backlink](#)

Unknown interpreted text role "func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 57); [backlink](#)

Unknown interpreted text role "attr".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 57); [backlink](#)

Unknown interpreted text role "class".

No feature description will ever be deleted from `:mod:'__future__'`. Since its introduction in Python 2.1 the following features have found their way into the language using this mechanism:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]\_\_future\_\_.rst, line 62); [backlink](#)

Unknown interpreted text role "mod".

feature	optional in	mandatory in	effect
nested_scopes	2.1.0b1	2.2	<a href="#">PEP 227</a> : <i>Statically Nested Scopes</i>
generators	2.2.0a1	2.3	<a href="#">PEP 255</a> : <i>Simple Generators</i>
division	2.2.0a2	3.0	<a href="#">PEP 238</a> : <i>Changing the Division Operator</i>
absolute_import	2.5.0a1	3.0	<a href="#">PEP 328</a> : <i>Imports: Multi-Line and Absolute/Relative</i>
with_statement	2.5.0a1	2.6	<a href="#">PEP 343</a> : <i>The "with" Statement</i>
print_function	2.6.0a2	3.0	<a href="#">PEP 3105</a> : <i>Make print a function</i>
unicode_literals	2.6.0a2	3.0	<a href="#">PEP 3112</a> : <i>Bytes literals in Python 3000</i>
generator_stop	3.5.0b1	3.7	<a href="#">PEP 479</a> : <i>StopIteration handling inside generators</i>
annotations	3.7.0b1	3.11	<a href="#">PEP 563</a> : <i>Postponed evaluation of annotations</i>

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]\_\_future\_\_.rst, line 100)**

Unknown directive type "seealso".

```
.. seealso::

   :ref:`future`
      How the compiler treats future imports.
```