Built-in Constants

A small number of constants live in the built-in namespace. They are:

```
.. data:: False

    The false value of the :class:`bool` type. Assignments to ``False``
    are illegal and raise a :exc:`SyntaxError`.
```

```
.. data:: True

    The true value of the :class:`bool` type. Assignments to ``True``
    are illegal and raise a :exc:`SyntaxError`.
```

```
.. data:: None

    An object frequently used to represent the absence of a value, as when
    default arguments are not passed to a function. Assignments to ``None``
    are illegal and raise a :exc:`SyntaxError`.
    ``None`` is the sole instance of the :data:`NoneType` type.
```

```
.. data:: NotImplemented

    A special value which should be returned by the binary special methods
    (e.g. :meth:`__eq__`, :meth:`__lt__`, :meth:`__add__`, :meth:`__rsub__`,
    etc.) to indicate that the operation is not implemented with respect to
    the other type; may be returned by the in-place binary special methods
    (e.g. :meth:`__imul__`, :meth:`__iand__`, etc.) for the same purpose.
    It should not be evaluated in a boolean context.
    ``NotImplemented`` is the sole instance of the :data:`types.NotImplementedType` type.

    .. note::

        When a binary (or in-place) method returns ``NotImplemented`` the
        interpreter will try the reflected operation on the other type (or some
        other fallback, depending on the operator).  If all attempts return
        ``NotImplemented``, the interpreter will raise an appropriate exception.
        Incorrectly returning ``NotImplemented`` will result in a misleading
        error message or the ``NotImplemented`` value being returned to Python code.

        See :ref:`implementing-the-arithmetic-operations` for examples.

    .. note::

        ``NotImplementedError`` and ``NotImplemented`` are not interchangeable,
        even though they have similar names and purposes.
        See :exc:`NotImplementedError` for details on when to use it.
```

```
.. versionchanged:: 3.9
   Evaluating ``NotImplemented`` in a boolean context is deprecated. While
   it currently evaluates as true, it will emit a :exc:`DeprecationWarning`.
   It will raise a :exc:`TypeError` in a future version of Python.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]constants.rst`, line 61)

Unknown directive type "index".

```
.. index:: single: ...; ellipsis literal
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]constants.rst`, line 62)

Unknown directive type "data".

```
.. data:: Ellipsis

   The same as the ellipsis literal "``...``". Special value used mostly in conjunction
   with extended slicing syntax for user-defined container data types.
   ``Ellipsis`` is the sole instance of the :data:`types.EllipsisType` type.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]constants.rst`, line 69)

Unknown directive type "data".

```
.. data:: __debug__

   This constant is true if Python was not started with an :option:`-O` option.
   See also the :keyword:`assert` statement.
```

Note

The names :data:`None`, :data:`False`, :data:`True` and :data:`__debug__` cannot be reassigned (assignments to them, even as an attribute name, raise :exc:`SyntaxError`), so they can be considered "true" constants.

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

Unknown interpreted text role "data".

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

Unknown interpreted text role "data".

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

Unknown interpreted text role "data".

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

Unknown interpreted text role "data".

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

Constants added by the :mod:`site` module

The :mod:`site` module (which is imported automatically during startup, except if the :option:`-S` command-line option is given) adds several constants to the built-in namespace. They are useful for the interactive interpreter shell and should not be used in programs.

```
.. data:: quit(code=None)
          exit(code=None)

   Objects that when printed, print a message like "Use quit() or Ctrl-D
   (i.e. EOF) to exit", and when called, raise :exc:`SystemExit` with the
   specified exit code.
```

```
.. data:: copyright
          credits

   Objects that when printed or called, print the text of copyright or
   credits, respectively.
```

```
.. data:: license

   Object that when printed, prints the message "Type license() to see the
   full license text", and when called, displays the full license text in a
   pager-like fashion (one screen at a time).
```