Number Protocol

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 1)

Unknown directive type "highlight".

```
.. highlight:: c
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 9)

Unknown directive type "c:function".

```
.. c:function:: int PyNumber_Check(PyObject *o)

   Returns ``1`` if the object *o* provides numeric protocols, and false otherwise.
   This function always succeeds.

   .. versionchanged:: 3.8
      Returns ``1`` if *o* is an index integer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 18)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Add(PyObject *o1, PyObject *o2)

   Returns the result of adding *o1* and *o2*, or ``NULL`` on failure.  This is the
   equivalent of the Python expression ``o1 + o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 24)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Subtract(PyObject *o1, PyObject *o2)

   Returns the result of subtracting *o2* from *o1*, or ``NULL`` on failure.  This is
   the equivalent of the Python expression ``o1 - o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 30)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Multiply(PyObject *o1, PyObject *o2)

   Returns the result of multiplying *o1* and *o2*, or ``NULL`` on failure.  This is
   the equivalent of the Python expression ``o1 * o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 36)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_MatrixMultiply(PyObject *o1, PyObject *o2)

   Returns the result of matrix multiplication on *o1* and *o2*, or ``NULL`` on
   failure.  This is the equivalent of the Python expression ``o1 @ o2``.

   .. versionadded:: 3.5
```

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

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_FloorDivide(PyObject *o1, PyObject *o2)

   Return the floor of *o1* divided by *o2*, or ``NULL`` on failure.  This is
   the equivalent of the Python expression ``o1 // o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 50)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_TrueDivide(PyObject *o1, PyObject *o2)

   Return a reasonable approximation for the mathematical value of *o1* divided by
   *o2*, or ``NULL`` on failure.  The return value is "approximate" because binary
   floating point numbers are approximate; it is not possible to represent all real
   numbers in base two.  This function can return a floating point value when
   passed two integers.  This is the equivalent of the Python expression ``o1 / o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 59)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Remainder(PyObject *o1, PyObject *o2)

   Returns the remainder of dividing *o1* by *o2*, or ``NULL`` on failure.  This is
   the equivalent of the Python expression ``o1 % o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 65)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Divmod(PyObject *o1, PyObject *o2)

   .. index:: builtin: divmod

   See the built-in function :func:`divmod`. Returns ``NULL`` on failure.  This is
   the equivalent of the Python expression ``divmod(o1, o2)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 73)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Power(PyObject *o1, PyObject *o2, PyObject *o3)

   .. index:: builtin: pow

   See the built-in function :func:`pow`. Returns ``NULL`` on failure.  This is the
   equivalent of the Python expression ``pow(o1, o2, o3)``, where *o3* is optional.
   If *o3* is to be ignored, pass :c:data:`Py_None` in its place (passing ``NULL`` for
   *o3* would cause an illegal memory access).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 83)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Negative(PyObject *o)

   Returns the negation of *o* on success, or ``NULL`` on failure. This is the
   equivalent of the Python expression ``-o``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 89)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Positive(PyObject *o)

   Returns *o* on success, or ``NULL`` on failure.  This is the equivalent of the
   Python expression ``+o``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 95)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Absolute(PyObject *o)

   .. index:: builtin: abs

   Returns the absolute value of *o*, or ``NULL`` on failure.  This is the equivalent
   of the Python expression ``abs(o)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 103)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Invert(PyObject *o)

   Returns the bitwise negation of *o* on success, or ``NULL`` on failure.  This is
   the equivalent of the Python expression ``~o``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 109)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Lshift(PyObject *o1, PyObject *o2)

   Returns the result of left shifting *o1* by *o2* on success, or ``NULL`` on
   failure.  This is the equivalent of the Python expression ``o1 << o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 115)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Rshift(PyObject *o1, PyObject *o2)

   Returns the result of right shifting *o1* by *o2* on success, or ``NULL`` on
   failure.  This is the equivalent of the Python expression ``o1 >> o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 121)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_And(PyObject *o1, PyObject *o2)
```

```
   Returns the "bitwise and" of *o1* and *o2* on success and ``NULL`` on failure.
   This is the equivalent of the Python expression ``o1 & o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 127)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Xor(PyObject *o1, PyObject *o2)

   Returns the "bitwise exclusive or" of *o1* by *o2* on success, or ``NULL`` on
   failure.  This is the equivalent of the Python expression ``o1 ^ o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 133)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Or(PyObject *o1, PyObject *o2)

   Returns the "bitwise or" of *o1* and *o2* on success, or ``NULL`` on failure.
   This is the equivalent of the Python expression ``o1 | o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 139)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceAdd(PyObject *o1, PyObject *o2)

   Returns the result of adding *o1* and *o2*, or ``NULL`` on failure.  The operation
   is done *in-place* when *o1* supports it.  This is the equivalent of the Python
   statement ``o1 += o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 146)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceSubtract(PyObject *o1, PyObject *o2)

   Returns the result of subtracting *o2* from *o1*, or ``NULL`` on failure.  The
   operation is done *in-place* when *o1* supports it.  This is the equivalent of
   the Python statement ``o1 -= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 153)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceMultiply(PyObject *o1, PyObject *o2)

   Returns the result of multiplying *o1* and *o2*, or ``NULL`` on failure.  The
   operation is done *in-place* when *o1* supports it.  This is the equivalent of
   the Python statement ``o1 *= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 160)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceMatrixMultiply(PyObject *o1, PyObject *o2)
```

```
   Returns the result of matrix multiplication on *o1* and *o2*, or ``NULL`` on
   failure.  The operation is done *in-place* when *o1* supports it.  This is
   the equivalent of the Python statement ``o1 @= o2``.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 169)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceFloorDivide(PyObject *o1, PyObject *o2)

   Returns the mathematical floor of dividing *o1* by *o2*, or ``NULL`` on failure.
   The operation is done *in-place* when *o1* supports it.  This is the equivalent
   of the Python statement ``o1 //= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 176)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceTrueDivide(PyObject *o1, PyObject *o2)

   Return a reasonable approximation for the mathematical value of *o1* divided by
   *o2*, or ``NULL`` on failure.  The return value is "approximate" because binary
   floating point numbers are approximate; it is not possible to represent all real
   numbers in base two.  This function can return a floating point value when
   passed two integers.  The operation is done *in-place* when *o1* supports it.
   This is the equivalent of the Python statement ``o1 /= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 186)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceRemainder(PyObject *o1, PyObject *o2)

   Returns the remainder of dividing *o1* by *o2*, or ``NULL`` on failure.  The
   operation is done *in-place* when *o1* supports it.  This is the equivalent of
   the Python statement ``o1 %= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 193)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlacePower(PyObject *o1, PyObject *o2, PyObject *o3)

   .. index:: builtin: pow

   See the built-in function :func:`pow`. Returns ``NULL`` on failure.  The operation
   is done *in-place* when *o1* supports it.  This is the equivalent of the Python
   statement ``o1 **= o2`` when o3 is :c:data:`Py_None`, or an in-place variant of
   ``pow(o1, o2, o3)`` otherwise. If *o3* is to be ignored, pass :c:data:`Py_None`
   in its place (passing ``NULL`` for *o3* would cause an illegal memory access).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 204)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceLshift(PyObject *o1, PyObject *o2)

   Returns the result of left shifting *o1* by *o2* on success, or ``NULL`` on
```

```
   failure.  The operation is done *in-place* when *o1* supports it.  This is the
   equivalent of the Python statement ``o1 <<= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 211)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceRshift(PyObject *o1, PyObject *o2)

   Returns the result of right shifting *o1* by *o2* on success, or ``NULL`` on
   failure.  The operation is done *in-place* when *o1* supports it.  This is the
   equivalent of the Python statement ``o1 >>= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 218)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceAnd(PyObject *o1, PyObject *o2)

   Returns the "bitwise and" of *o1* and *o2* on success and ``NULL`` on failure. The
   operation is done *in-place* when *o1* supports it.  This is the equivalent of
   the Python statement ``o1 &= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 225)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceXor(PyObject *o1, PyObject *o2)

   Returns the "bitwise exclusive or" of *o1* by *o2* on success, or ``NULL`` on
   failure.  The operation is done *in-place* when *o1* supports it.  This is the
   equivalent of the Python statement ``o1 ^= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 232)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_InPlaceOr(PyObject *o1, PyObject *o2)

   Returns the "bitwise or" of *o1* and *o2* on success, or ``NULL`` on failure.  The
   operation is done *in-place* when *o1* supports it.  This is the equivalent of
   the Python statement ``o1 |= o2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 239)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Long(PyObject *o)

   .. index:: builtin: int

   Returns the *o* converted to an integer object on success, or ``NULL`` on
   failure.  This is the equivalent of the Python expression ``int(o)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 247)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Float(PyObject *o)

   .. index:: builtin: float

   Returns the *o* converted to a float object on success, or ``NULL`` on failure.
   This is the equivalent of the Python expression ``float(o)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 255)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_Index(PyObject *o)

   Returns the *o* converted to a Python int on success or ``NULL`` with a
   :exc:`TypeError` exception raised on failure.

   .. versionchanged:: 3.10
      The result always has exact type :class:`int`.  Previously, the result
      could have been an instance of a subclass of ``int``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 265)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyNumber_ToBase(PyObject *n, int base)

   Returns the integer *n* converted to base *base* as a string.  The *base*
   argument must be one of 2, 8, 10, or 16.  For base 2, 8, or 16, the
   returned string is prefixed with a base marker of ``'0b'``, ``'0o'``, or
   ``'0x'``, respectively.  If *n* is not a Python int, it is converted with
   :c:func:`PyNumber_Index` first.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 274)

Unknown directive type "c:function".

```
.. c:function:: Py_ssize_t PyNumber_AsSsize_t(PyObject *o, PyObject *exc)

   Returns *o* converted to a Py_ssize_t value if *o* can be interpreted as an
   integer.  If the call fails, an exception is raised and ``-1`` is returned.

   If *o* can be converted to a Python int but the attempt to
   convert to a Py_ssize_t value would raise an :exc:`OverflowError`, then the
   *exc* argument is the type of exception that will be raised (usually
   :exc:`IndexError` or :exc:`OverflowError`).  If *exc* is ``NULL``, then the
   exception is cleared and the value is clipped to ``PY_SSIZE_T_MIN`` for a negative
   integer or ``PY_SSIZE_T_MAX`` for a positive integer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]number.rst, line 287)

Unknown directive type "c:function".

```
.. c:function:: int PyIndex_Check(PyObject *o)

   Returns ``1`` if *o* is an index integer (has the ``nb_index`` slot of the
   ``tp_as_number`` structure filled in), and ``0`` otherwise.
   This function always succeeds.
```