Complex Number Objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.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)complex.rst, line 8)

Unknown directive type "index".

```
.. index:: object: complex number
```

Python's complex number objects are implemented as two distinct types when viewed from the C API: one is the Python object exposed to Python programs, and the other is a C structure which represents the actual complex number value. The API provides functions for working with both.

Complex Numbers as C Structures

Note that the functions which accept these structures as parameters and return them as results do so *by value* rather than dereferencing them through pointers. This is consistent throughout the API.

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

Unknown directive type "c:type".

```
.. c:type:: Py_complex

   The C structure which corresponds to the value portion of a Python complex
   number object.  Most of the functions for dealing with complex number objects
   use structures of this type as input or output values, as appropriate.  It is
   defined as::

      typedef struct {
         double real;
         double imag;
      } Py_complex;
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 37)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_sum(Py_complex left, Py_complex right)

   Return the sum of two complex numbers, using the C :c:type:`Py_complex`
   representation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 43)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_diff(Py_complex left, Py_complex right)

   Return the difference between two complex numbers, using the C
   :c:type:`Py_complex` representation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 49)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_neg(Py_complex num)

   Return the negation of the complex number *num*, using the C
   :c:type:`Py_complex` representation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 55)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_prod(Py_complex left, Py_complex right)

   Return the product of two complex numbers, using the C :c:type:`Py_complex`
   representation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 61)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_quot(Py_complex dividend, Py_complex divisor)

   Return the quotient of two complex numbers, using the C :c:type:`Py_complex`
   representation.

   If *divisor* is null, this method returns zero and sets
   :c:data:`errno` to :c:data:`EDOM`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 70)

Unknown directive type "c:function".

```
.. c:function:: Py_complex _Py_c_pow(Py_complex num, Py_complex exp)

   Return the exponentiation of *num* by *exp*, using the C :c:type:`Py_complex`
   representation.

   If *num* is null and *exp* is not a positive real number,
   this method returns zero and sets :c:data:`errno` to :c:data:`EDOM`.
```

Complex Numbers as Python Objects

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

Unknown directive type "c:type".

```
.. c:type:: PyComplexObject

   This subtype of :c:type:`PyObject` represents a Python complex number object.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 88)

Unknown directive type "c:var".

```
.. c:var:: PyTypeObject PyComplex_Type

   This instance of :c:type:`PyTypeObject` represents the Python complex number
```

```
   type. It is the same object as :class:`complex` in the Python layer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 94)

Unknown directive type "c:function".

```
.. c:function:: int PyComplex_Check(PyObject *p)

   Return true if its argument is a :c:type:`PyComplexObject` or a subtype of
   :c:type:`PyComplexObject`.  This function always succeeds.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 100)

Unknown directive type "c:function".

```
.. c:function:: int PyComplex_CheckExact(PyObject *p)

   Return true if its argument is a :c:type:`PyComplexObject`, but not a subtype of
   :c:type:`PyComplexObject`.  This function always succeeds.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 106)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyComplex_FromCComplex(Py_complex v)

   Create a new Python complex number object from a C :c:type:`Py_complex` value.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 111)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PyComplex_FromDoubles(double real, double imag)

   Return a new :c:type:`PyComplexObject` object from *real* and *imag*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 116)

Unknown directive type "c:function".

```
.. c:function:: double PyComplex_RealAsDouble(PyObject *op)

   Return the real part of *op* as a C :c:type:`double`.
```

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

Unknown directive type "c:function".

```
.. c:function:: double PyComplex_ImagAsDouble(PyObject *op)

   Return the imaginary part of *op* as a C :c:type:`double`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)complex.rst, line 126)

Unknown directive type "c:function".

```
.. c:function:: Py_complex PyComplex_AsCComplex(PyObject *op)

   Return the :c:type:`Py_complex` value of the complex number *op*.

   If *op* is not a Python complex number object but has a :meth:`__complex__`
   method, this method will first be called to convert *op* to a Python complex
   number object.  If ``__complex__()`` is not defined then it falls back to
   :meth:`__float__`.  If ``__float__()`` is not defined then it falls back
   to :meth:`__index__`.  Upon failure, this method returns ``-1.0`` as a real
   value.

   .. versionchanged:: 3.8
      Use :meth:`__index__` if available.
```