Type Objects

Perhaps one of the most important structures of the Python object system is the structure that defines a new type: the :c:type:`PyTypeObject` structure. Type objects can be handled using any of the :c:func:`PyObject_\*` or :c:func:`PyType_\*` functions, but do not offer much that's interesting to most Python applications. These objects are fundamental to how objects behave, so they are very important to the interpreter itself and to any extension module that implements new types.

Type objects are fairly large compared to most of the standard types. The reason for the size is that each type object stores a large number of values, mostly C function pointers, each of which implements a small part of the type's functionality. The fields of the type object are examined in detail in this section. The fields will be described in the order in which they occur in the structure.

In addition to the following quick reference, the :ref:`typedef-examples` section provides at-a-glance insight into the meaning and use of :c:type:`PyTypeObject`.

Quick Reference

"tp slots"

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>`	special methods/attrs	Info [2]			
			O	T	D	I
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink* — **Unknown interpreted text role "ref".**					
\<R\> :c:member:`~PyTypeObject.tp_name` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 45`); *backlink* — Unknown interpreted text role "c:member".	const char *	__name__	X	X		

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41);` *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	T	D	I
:c:member:`~PyTypeObject.tp_basicsize` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 47);` *backlink* Unknown interpreted text role "c:member".	Py_ssize_t		X	X		X
:c:member:`~PyTypeObject.tp_itemsize` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 49);` *backlink* Unknown interpreted text role "c:member".	Py_ssize_t			X		X
:c:member:`~PyTypeObject.tp_dealloc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 51);` *backlink* Unknown interpreted text role "c:member".	:c:type:`destructor` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 51);` *backlink* Unknown interpreted text role "c:type".		X	X		X
:c:member:`~PyTypeObject.tp_vectorcall_offset` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 53);` *backlink* Unknown interpreted text role "c:member".	Py_ssize_t			X		X

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink*** **Unknown interpreted text role "ref".**	special methods/attrs	O	T	D	I
(:c:member:`~PyTypeObject.tp_getattr`) **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 55`); *backlink*** Unknown interpreted text role "c:member".	:c:type:`getattrfunc` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 55`); *backlink*** Unknown interpreted text role "c:type".	__getattribute__, __getattr__				G
(:c:member:`~PyTypeObject.tp_setattr`) **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 58`); *backlink*** Unknown interpreted text role "c:member".	:c:type:`setattrfunc` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 58`); *backlink*** Unknown interpreted text role "c:type".	__setattr__, __delattr__				G
:c:member:`~PyTypeObject.tp_as_async` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 61`); *backlink*** Unknown interpreted text role "c:member".	:c:type:`PyAsyncMethods` * **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 61`); *backlink*** Unknown interpreted text role "c:type".	:ref:`sub-slots` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 61`); *backlink*** Unknown interpreted text role "ref".				%
:c:member:`~PyTypeObject.tp_repr` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 63`); *backlink*** Unknown interpreted text role "c:member".	:c:type:`reprfunc` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 63`); *backlink*** Unknown interpreted text role "c:type".	__repr__	X	X		X

Info [2]

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 41); backlink** **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	T	D	I
:c:member:`~PyTypeObject.tp_as_number` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 65); backlink** Unknown interpreted text role "c:member".	:c:type:`PyNumberMethods` * **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 65); backlink** Unknown interpreted text role "c:type".	:ref:`sub-slots` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 65); backlink** Unknown interpreted text role "ref".				%
:c:member:`~PyTypeObject.tp_as_sequence` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 67); backlink** Unknown interpreted text role "c:member".	:c:type:`PySequenceMethods` * **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 67); backlink** Unknown interpreted text role "c:type".	:ref:`sub-slots` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 67); backlink** Unknown interpreted text role "ref".				%
:c:member:`~PyTypeObject.tp_as_mapping` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 69); backlink** Unknown interpreted text role "c:member".	:c:type:`PyMappingMethods` * **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 69); backlink** Unknown interpreted text role "c:type".	:ref:`sub-slots` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 69); backlink** Unknown interpreted text role "ref".				%
:c:member:`~PyTypeObject.tp_hash` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 71); backlink** Unknown interpreted text role "c:member".	:c:type:`hashfunc` **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 71); backlink** Unknown interpreted text role "c:type".	__hash__	X		D	G

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41); *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	**T**	**D**	**I**
:c:member:`~PyTypeObject.tp_call` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 73); *backlink* Unknown interpreted text role "c:member".	:c:type:`ternaryfunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 73); *backlink* Unknown interpreted text role "c:type".	__call__		X		X
:c:member:`~PyTypeObject.tp_str` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 75); *backlink* Unknown interpreted text role "c:member".	:c:type:`reprfunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 75); *backlink* Unknown interpreted text role "c:type".	__str__	X			X
:c:member:`~PyTypeObject.tp_getattro` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 77); *backlink* Unknown interpreted text role "c:member".	:c:type:`getattrofunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 77); *backlink* Unknown interpreted text role "c:type".	__getattribute__, __getattr__	X	X		G
:c:member:`~PyTypeObject.tp_setattro` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 80); *backlink* Unknown interpreted text role "c:member".	:c:type:`setattrofunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 80); *backlink* Unknown interpreted text role "c:type".	__setattr__, __delattr__	X	X		G

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41); *backlink*` **Unknown interpreted text role "ref".**	**special methods/attrs**	**Info [2]**			
			O	**T**	**D**	**I**
:c:member:`~PyTypeObject.tp_as_buffer` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 83); *backlink*` Unknown interpreted text role "c:member".	:c:type:`PyBufferProcs` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 83); *backlink*` Unknown interpreted text role "c:type".					%
:c:member:`~PyTypeObject.tp_flags` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 85); *backlink*` Unknown interpreted text role "c:member".	unsigned long		X	X		?
:c:member:`~PyTypeObject.tp_doc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 87); *backlink*` Unknown interpreted text role "c:member".	const char *	__doc__	X	X		
:c:member:`~PyTypeObject.tp_traverse` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 89); *backlink*` Unknown interpreted text role "c:member".	:c:type:`traverseproc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 89); *backlink*` Unknown interpreted text role "c:type".			X		G

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	**Info [2]**			
			O	**T**	**D**	**I**
:c:member:`~PyTypeObject.tp_clear` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 91`); *backlink* Unknown interpreted text role "c:member".	:c:type:`inquiry` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 91`); *backlink* Unknown interpreted text role "c:type".			X		G
:c:member:`~PyTypeObject.tp_richcompare` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 93`); *backlink* Unknown interpreted text role "c:member".	:c:type:`richcmpfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 93`); *backlink* Unknown interpreted text role "c:type".	__lt__, __le__, __eq__, __ne__, __gt__, __ge__	X			G
:c:member:`~PyTypeObject.tp_weaklistoffset` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 100`); *backlink* Unknown interpreted text role "c:member".	Py_ssize_t			X		?
:c:member:`~PyTypeObject.tp_iter` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 102`); *backlink* Unknown interpreted text role "c:member".	:c:type:`getiterfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 102`); *backlink* Unknown interpreted text role "c:type".	__iter__				X

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>`	special methods/attrs	O	T	D	I
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink* **Unknown interpreted text role "ref".**					
:c:member:`~PyTypeObject.tp_iternext` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 104`); *backlink* Unknown interpreted text role "c:member".	:c:type:`iternextfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 104`); *backlink* Unknown interpreted text role "c:type".	__next__				X
:c:member:`~PyTypeObject.tp_methods` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 106`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyMethodDef` [] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 106`); *backlink* Unknown interpreted text role "c:type".		X	X		
:c:member:`~PyTypeObject.tp_members` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 108`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyMemberDef` [] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 108`); *backlink* Unknown interpreted text role "c:type".			X		
:c:member:`~PyTypeObject.tp_getset` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 110`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyGetSetDef` [] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 110`); *backlink* Unknown interpreted text role "c:type".		X	X		

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>`	special methods/attrs	Info [2]			
			O	T	D	I
PyTypeObject Slot [1]	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 41); *backlink*** **Unknown interpreted text role "ref".**	special methods/attrs				
:c:member:`~PyTypeObject.tp_base` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 112); *backlink*** Unknown interpreted text role "c:member".	:c:type:`PyTypeObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 112); *backlink*** Unknown interpreted text role "c:type".	__base__			X	
:c:member:`~PyTypeObject.tp_dict` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 114); *backlink*** Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 114); *backlink*** Unknown interpreted text role "c:type".	__dict__			?	
:c:member:`~PyTypeObject.tp_descr_get` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 116); *backlink*** Unknown interpreted text role "c:member".	:c:type:`descrgetfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 116); *backlink*** Unknown interpreted text role "c:type".	__get__			X	
:c:member:`~PyTypeObject.tp_descr_set` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 118); *backlink*** Unknown interpreted text role "c:member".	:c:type:`descrsetfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api)typeobj.rst, line 118); *backlink*** Unknown interpreted text role "c:type".	__set__, __delete__			X	

PyTypeObject Slot [1]	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41); *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	**T**	**D**	**I**
:c:member:`~PyTypeObject.tp_dictoffset` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 121); *backlink* Unknown interpreted text role "c:member".	Py_ssize_t				X	?
:c:member:`~PyTypeObject.tp_init` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 123); *backlink* Unknown interpreted text role "c:member".	:c:type:`initproc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 123); *backlink* Unknown interpreted text role "c:type".	__init__	X	X		X
:c:member:`~PyTypeObject.tp_alloc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 125); *backlink* Unknown interpreted text role "c:member".	:c:type:`allocfunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 125); *backlink* Unknown interpreted text role "c:type".		X		?	?
:c:member:`~PyTypeObject.tp_new` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 127); *backlink* Unknown interpreted text role "c:member".	:c:type:`newfunc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 127); *backlink* Unknown interpreted text role "c:type".	__new__	X	X	?	?

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	T	D	I
:c:member:`~PyTypeObject.tp_free` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 129`); *backlink* Unknown interpreted text role "c:member".	:c:type:`freefunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 129`); *backlink* Unknown interpreted text role "c:type".		X	X	?	?
:c:member:`~PyTypeObject.tp_is_gc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 131`); *backlink* Unknown interpreted text role "c:member".	:c:type:`inquiry` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 131`); *backlink* Unknown interpreted text role "c:type".		X		X	
<:c:member:`~PyTypeObject.tp_bases`> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 133`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 133`); *backlink* Unknown interpreted text role "c:type".	__bases__			~	
<:c:member:`~PyTypeObject.tp_mro`> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 135`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 135`); *backlink* Unknown interpreted text role "c:type".	__mro__			~	

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 41`); *backlink* **Unknown interpreted text role "ref".**	special methods/attrs	Info [2]			
			O	T	D	I
[:c:member:`~PyTypeObject.tp_cache`] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 137`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 137`); *backlink* Unknown interpreted text role "c:type".					
[:c:member:`~PyTypeObject.tp_subclasses`] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 139`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 139`); *backlink* Unknown interpreted text role "c:type".	__subclasses__				
[:c:member:`~PyTypeObject.tp_weaklist`] **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 141`); *backlink* Unknown interpreted text role "c:member".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 141`); *backlink* Unknown interpreted text role "c:type".					
(:c:member:`~PyTypeObject.tp_del`) **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 143`); *backlink* Unknown interpreted text role "c:member".	:c:type:`destructor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 143`); *backlink* Unknown interpreted text role "c:type".					

PyTypeObject Slot [1]	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 41); backlink** Unknown interpreted text role "ref".	special methods/attrs	O	T	D	I
Info [2]						
[:c:member:`~PyTypeObject.tp_version_tag`] **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 145); backlink** Unknown interpreted text role "c:member".		unsigned int				
:c:member:`~PyTypeObject.tp_finalize` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 147); backlink** Unknown interpreted text role "c:member".	:c:type:`destructor` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 147); backlink** Unknown interpreted text role "c:type".	__del__				X
:c:member:`~PyTypeObject.tp_vectorcall` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 149); backlink** Unknown interpreted text role "c:member".	:c:type:`vectorcallfunc` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 149); backlink** Unknown interpreted text role "c:type".					

[1] A slot name in parentheses indicates it is (effectively) deprecated. Names in angle brackets should be treated as read-only. Names in square brackets are for internal use only. "<R>" (as a prefix) means the field is required (must be non-`NULL`).

[2] Columns:

"O": set on :c:type:`PyBaseObject_Type`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 158); backlink Unknown interpreted text role "c:type".

"T": set on :c:type:`PyType_Type`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 160); backlink Unknown interpreted text role "c:type".

"D": default (if slot is set to `NULL`)

```none
.. code-block:: none

    X - PyType_Ready sets this value if it is NULL
    ~ - PyType_Ready always sets this value (it should be NULL)
    ? - PyType_Ready may set this value depending on other slots

    Also see the inheritance column ("I").
```

"I": inheritance

```none
.. code-block:: none

    X - type slot is inherited via *PyType_Ready* if defined with a *NULL* value
    % - the slots of the sub-struct are inherited individually
    G - inherited, but only in combination with other slots; see the slot's description
    ? - it's complicated; see the slot's description
```

Note that some slots are effectively inherited through the normal attribute lookup chain.

sub-slots

Slot	:ref:`Type <slot-typedefs-table>`	special methods
:c:member:`~PyAsyncMethods.am_await`	:c:type:`unaryfunc`	__await__
:c:member:`~PyAsyncMethods.am_aiter`	:c:type:`unaryfunc`	__aiter__

Slot		:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 194);** *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyAsyncMethods.am_anext` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 201);** *backlink* Unknown interpreted text role "c:member".		:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 201);** *backlink* Unknown interpreted text role "c:type".	__anext__
:c:member:`~PyAsyncMethods.am_send` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 203);** *backlink* Unknown interpreted text role "c:member".		:c:type:`sendfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 203);** *backlink* Unknown interpreted text role "c:type".	
:c:member:`~PyNumberMethods.nb_add` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 207);** *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 207);** *backlink* Unknown interpreted text role "c:type".	__add__ __radd__
:c:member:`~PyNumberMethods.nb_inplace_add` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 210);** *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 210);** *backlink* Unknown interpreted text role "c:type".	__iadd__

Slot		:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 194);` *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_subtract` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 212);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 212);` *backlink* Unknown interpreted text role "c:type".	__sub__ __rsub__
:c:member:`~PyNumberMethods.nb_inplace_subtract` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 215);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 215);` *backlink* Unknown interpreted text role "c:type".	__isub__
:c:member:`~PyNumberMethods.nb_multiply` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 217);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 217);` *backlink* Unknown interpreted text role "c:type".	__mul__ __rmul__
:c:member:`~PyNumberMethods.nb_inplace_multiply` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 220);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 220);` *backlink* Unknown interpreted text role "c:type".	__imul__

Slot	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 194`); *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_remainder` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 222`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 222`); *backlink* Unknown interpreted text role "c:type".	__mod__ __rmod__
:c:member:`~PyNumberMethods.nb_inplace_remainder` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 225`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 225`); *backlink* Unknown interpreted text role "c:type".	__imod__
:c:member:`~PyNumberMethods.nb_divmod` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 227`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 227`); *backlink* Unknown interpreted text role "c:type".	__divmod__ __rdivmod__
:c:member:`~PyNumberMethods.nb_power` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 230`); *backlink* Unknown interpreted text role "c:member".	:c:type:`ternaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 230`); *backlink* Unknown interpreted text role "c:type".	__pow__ __rpow__

Slot	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 194`); *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_inplace_power` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 233`); *backlink* Unknown interpreted text role "c:member".	:c:type:`ternaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 233`); *backlink* Unknown interpreted text role "c:type".	__ipow__
:c:member:`~PyNumberMethods.nb_negative` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 235`); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 235`); *backlink* Unknown interpreted text role "c:type".	__neg__
:c:member:`~PyNumberMethods.nb_positive` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 237`); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 237`); *backlink* Unknown interpreted text role "c:type".	__pos__
:c:member:`~PyNumberMethods.nb_absolute` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 239`); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 239`); *backlink* Unknown interpreted text role "c:type".	__abs__

Slot	:ref:`Type <slot-typedefs-table>`	special methods

	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 194`); *backlink* **Unknown interpreted text role "ref".**	
Slot		**special methods**
:c:member:`~PyNumberMethods.nb_bool` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 241`); *backlink* Unknown interpreted text role "c:member".	:c:type:`inquiry` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 241`); *backlink* Unknown interpreted text role "c:type".	__bool__
:c:member:`~PyNumberMethods.nb_invert` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 243`); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 243`); *backlink* Unknown interpreted text role "c:type".	__invert__
:c:member:`~PyNumberMethods.nb_lshift` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 245`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 245`); *backlink* Unknown interpreted text role "c:type".	__lshift__ __rlshift__
:c:member:`~PyNumberMethods.nb_inplace_lshift` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 248`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 248`); *backlink* Unknown interpreted text role "c:type".	__ilshift__

Slot		:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 194);` *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_rshift` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 250);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 250);` *backlink* Unknown interpreted text role "c:type".	__rshift__ __rrshift__
:c:member:`~PyNumberMethods.nb_inplace_rshift` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 253);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 253);` *backlink* Unknown interpreted text role "c:type".	__irshift__
:c:member:`~PyNumberMethods.nb_and` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 255);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 255);` *backlink* Unknown interpreted text role "c:type".	__and__ __rand__
:c:member:`~PyNumberMethods.nb_inplace_and` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 258);` *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 258);` *backlink* Unknown interpreted text role "c:type".	__iand__
Slot			**special methods**

Slot	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 194`); *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_xor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 260`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 260`); *backlink* Unknown interpreted text role "c:type".	__xor__ __rxor__
:c:member:`~PyNumberMethods.nb_inplace_xor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 263`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 263`); *backlink* Unknown interpreted text role "c:type".	__ixor__
:c:member:`~PyNumberMethods.nb_or` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 265`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 265`); *backlink* Unknown interpreted text role "c:type".	__or__ __ror__
:c:member:`~PyNumberMethods.nb_inplace_or` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 268`); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 268`); *backlink* Unknown interpreted text role "c:type".	__ior__

Slot	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 194**); *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_int` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 270**); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 270**); *backlink* Unknown interpreted text role "c:type".	__int__
:c:member:`~PyNumberMethods.nb_reserved` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 272**); *backlink* Unknown interpreted text role "c:member".	void *	
:c:member:`~PyNumberMethods.nb_float` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 274**); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 274**); *backlink* Unknown interpreted text role "c:type".	__float__
:c:member:`~PyNumberMethods.nb_floor_divide` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 276**); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst`, **line 276**); *backlink* Unknown interpreted text role "c:type".	__floordiv__

Slot	**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 194**); *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyNumberMethods.nb_inplace_floor_divide` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 278**); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 278**); *backlink* Unknown interpreted text role "c:type".	__ifloordiv__
:c:member:`~PyNumberMethods.nb_true_divide` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 280**); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 280**); *backlink* Unknown interpreted text role "c:type".	__truediv__
:c:member:`~PyNumberMethods.nb_inplace_true_divide` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 282**); *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 282**); *backlink* Unknown interpreted text role "c:type".	__itruediv__
:c:member:`~PyNumberMethods.nb_index` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 284**); *backlink* Unknown interpreted text role "c:member".	:c:type:`unaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 284**); *backlink* Unknown interpreted text role "c:type".	__index__
Slot		

Slot	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 194);** *backlink* **Unknown interpreted text role "ref".**		**special methods**
:c:member:`~PyNumberMethods.nb_matrix_multiply` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 286);** *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 286);** *backlink* Unknown interpreted text role "c:type".		__matmul__ __rmatmul__
:c:member:`~PyNumberMethods.nb_inplace_matrix_multiply` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 289);** *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 289);** *backlink* Unknown interpreted text role "c:type".		__imatmul__
:c:member:`~PyMappingMethods.mp_length` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 293);** *backlink* Unknown interpreted text role "c:member".	:c:type:`lenfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 293);** *backlink* Unknown interpreted text role "c:type".		__len__
:c:member:`~PyMappingMethods.mp_subscript` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 295);** *backlink* Unknown interpreted text role "c:member".	:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 295);** *backlink* Unknown interpreted text role "c:type".		__getitem__

Slot		**:ref:`Type <slot-typedefs-table>`** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 194);** *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PyMappingMethods.mp_ass_subscript` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 297);** *backlink* Unknown interpreted text role "c:member".		:c:type:`objobjargproc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 297);** *backlink* Unknown interpreted text role "c:type".	__setitem__, __delitem__
:c:member:`~PySequenceMethods.sq_length` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 302);** *backlink* Unknown interpreted text role "c:member".		:c:type:`lenfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 302);** *backlink* Unknown interpreted text role "c:type".	__len__
:c:member:`~PySequenceMethods.sq_concat` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 304);** *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 304);** *backlink* Unknown interpreted text role "c:type".	__add__
:c:member:`~PySequenceMethods.sq_repeat` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 306);** *backlink* Unknown interpreted text role "c:member".		:c:type:`ssizeargfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 306);** *backlink* Unknown interpreted text role "c:type".	__mul__

Slot		:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 194);** *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PySequenceMethods.sq_item` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 308);** *backlink* Unknown interpreted text role "c:member".		:c:type:`ssizeargfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 308);** *backlink* Unknown interpreted text role "c:type".	__getitem__
:c:member:`~PySequenceMethods.sq_ass_item` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 310);** *backlink* Unknown interpreted text role "c:member".		:c:type:`ssizeobjargproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 310);** *backlink* Unknown interpreted text role "c:type".	__setitem__ __delitem__
:c:member:`~PySequenceMethods.sq_contains` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 313);** *backlink* Unknown interpreted text role "c:member".		:c:type:`objobjproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 313);** *backlink* Unknown interpreted text role "c:type".	__contains__
:c:member:`~PySequenceMethods.sq_inplace_concat` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 315);** *backlink* Unknown interpreted text role "c:member".		:c:type:`binaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst,` **line 315);** *backlink* Unknown interpreted text role "c:type".	__iadd__
Slot			**special methods**

Slot	:ref:`Type <slot-typedefs-table>` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 194);` *backlink* **Unknown interpreted text role "ref".**	**special methods**
:c:member:`~PySequenceMethods.sq_inplace_repeat` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 317);` *backlink* Unknown interpreted text role "c:member".	:c:type:`ssizeargfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 317);` *backlink* Unknown interpreted text role "c:type".	__imul__
:c:member:`~PyBufferProcs.bf_getbuffer` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 321);` *backlink* Unknown interpreted text role "c:member".	:c:func:`getbufferproc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 321);` *backlink* Unknown interpreted text role "c:func".	
:c:member:`~PyBufferProcs.bf_releasebuffer` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 323);` *backlink* Unknown interpreted text role "c:member".	:c:func:`releasebufferproc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 323);` *backlink* Unknown interpreted text role "c:func".	

slot typedefs

typedef	Parameter Types	Return Type

typedef	Parameter Types	Return Type
:c:type:`allocfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 334);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyTypeObject` * Py_ssize_t **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 669);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 334);` *backlink* Unknown interpreted text role "c:type".
:c:type:`destructor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 339);` *backlink* Unknown interpreted text role "c:type".	void *	void
:c:type:`freefunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 341);` *backlink* Unknown interpreted text role "c:type".	void *	void
:c:type:`traverseproc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 343);` *backlink* Unknown interpreted text role "c:type".	void * :c:type:`visitproc` void * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 688);` *backlink* Unknown interpreted text role "c:type".	int

typedef	Parameter Types	Return Type
	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` *	
:c:type:`newfunc`	**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 699); *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` *
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 349); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 700); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 349); *backlink* Unknown interpreted text role "c:type".
	System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 701); *backlink* Unknown interpreted text role "c:type".	
	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` *	

typedef	Parameter Types	Return Type
:c:type:`initproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 355);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 711);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 712);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 713);` *backlink* Unknown interpreted text role "c:type".	int
:c:type:`reprfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 361);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 361);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 361);` *backlink* Unknown interpreted text role "c:type".
:c:type:`getattrfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 363);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * const char * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 727);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 363);` *backlink* Unknown interpreted text role "c:type".

typedef	Parameter Types	Return Type
:c:type:`setattrfunc` **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 368); *backlink*** Unknown interpreted text role "c:type".	:c:type:`PyObject` * const char * :c:type:`PyObject` * **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 737); *backlink*** Unknown interpreted text role "c:type". **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 739); *backlink*** Unknown interpreted text role "c:type".	int
:c:type:`getattrofunc` **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 374); *backlink*** Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`PyObject` * **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 749); *backlink*** Unknown interpreted text role "c:type". **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 750); *backlink*** Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** **(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 374); *backlink*** Unknown interpreted text role "c:type".

typedef	Parameter Types	Return Type
	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` *	
:c:type:`setattrofunc`		int

System Message: ERROR/3 (typedef column)
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 379); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (Parameter Types column)
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 759); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 760); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst, line 761); *backlink*

Unknown interpreted text role "c:type".

typedef	Parameter Types	Return Type
	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` *	
:c:type:`descrgetfunc`		:c:type:`PyObject` *

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 385); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 771); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 772); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 773); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3
(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 385); *backlink*

Unknown interpreted text role "c:type".

typedef	Parameter Types	Return Type
:c:type:`descrsetfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 391);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 783);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 784);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 785);` *backlink* Unknown interpreted text role "c:type".	int
:c:type:`hashfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 397);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main)(Doc) (c-api)typeobj.rst, line 397);` *backlink* Unknown interpreted text role "c:type".	Py_hash_t

typedef	Parameter Types	Return Type
	:c:type:`PyObject` * :c:type:`PyObject` * int	
:c:type:`richcmpfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 399);` *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 799);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 800);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 399);` *backlink* Unknown interpreted text role "c:type".
:c:type:`getiterfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 405);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 405);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 405);` *backlink* Unknown interpreted text role "c:type".
:c:type:`iternextfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 407);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 407);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 407);` *backlink* Unknown interpreted text role "c:type".
:c:type:`lenfunc` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 409);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 409);` *backlink* Unknown interpreted text role "c:type".	Py_ssize_t

typedef	Parameter Types	Return Type
:c:type:`getbufferproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 411);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`Py_buffer` * int **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 823);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 824);` *backlink* Unknown interpreted text role "c:type".	int
:c:type:`releasebufferproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 417);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`Py_buffer` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 835);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 836);` *backlink* Unknown interpreted text role "c:type".	void
:c:type:`inquiry` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 422);` *backlink* Unknown interpreted text role "c:type".	void *	int

typedef	Parameter Types	Return Type
:c:type:`unaryfunc`	:c:type:`PyObject` *	:c:type:`PyObject` *
		System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 424`); *backlink* Unknown interpreted text role "c:type".
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 849`); *backlink* Unknown interpreted text role "c:type".	
System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 424`); *backlink* Unknown interpreted text role "c:type".		
:c:type:`binaryfunc`	:c:type:`PyObject` * :c:type:`PyObject` *	:c:type:`PyObject` *
System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 428`); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 857`); *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 858`); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 428`); *backlink* Unknown interpreted text role "c:type".

typedef	Parameter Types	Return Type
:c:type:`ternaryfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 433);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 867);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 868);` *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 869);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 433);` *backlink* Unknown interpreted text role "c:type".
:c:type:`ssizeargfunc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 439);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * Py_ssize_t **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 879);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 439);` *backlink* Unknown interpreted text role "c:type".
:c:type:`ssizeobjargproc` **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 444);` *backlink* Unknown interpreted text role "c:type".	:c:type:`PyObject` * Py_ssize_t **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) typeobj.rst, line 889);` *backlink* Unknown interpreted text role "c:type".	int

typedef	Parameter Types	Return Type
	:c:type:`PyObject` * :c:type:`PyObject` *	
:c:type:`objobjproc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 449); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 899); *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 900); *backlink* Unknown interpreted text role "c:type".	int
	:c:type:`PyObject` * :c:type:`PyObject` * :c:type:`PyObject` *	
:c:type:`objobjargproc` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 454); *backlink* Unknown interpreted text role "c:type".	**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 909); *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 910); *backlink* Unknown interpreted text role "c:type". **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 911); *backlink* Unknown interpreted text role "c:type".	int

See :ref:`slot-typedefs` below for more detail.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst, line 460); *backlink*
Unknown interpreted text role "ref".

PyTypeObject Definition

The structure definition for :c:type:`PyTypeObject` can be found in :file:`Include/object.h`. For convenience of reference, this repeats the definition found there:

PyObject Slots

The type object structure extends the :c:type:`PyVarObject` structure. The :attr:`ob_size` field is used for dynamic types (created by :func:`type_new`, usually called from a class statement). Note that :c:data:`PyType_Type` (the metatype) initializes :c:member:`~PyTypeObject.tp_itemsize`, which means that its instances (i.e. type objects) *must* have the :attr:`ob_size` field.

```
start of the module's initialization function, before doing anything else.  This
is typically done like this::

    Foo_Type.ob_type = &PyType_Type;

This should be done before any instances of the type are created.
:c:func:`PyType_Ready` checks if :attr:`ob_type` is ``NULL``, and if so,
initializes it to the :attr:`ob_type` field of the base class.
:c:func:`PyType_Ready` will not change this field if it is non-zero.

**Inheritance:**

This field is inherited by subtypes.
```

```
.. c:member:: PyObject* PyObject._ob_next
              PyObject* PyObject._ob_prev

   These fields are only present when the macro ``Py_TRACE_REFS`` is defined
   (see the :option:`configure --with-trace-refs option <--with-trace-refs>`).

   Their initialization to ``NULL`` is taken care of by the
   ``PyObject_HEAD_INIT`` macro.  For :ref:`statically allocated objects
   <static-types>`, these fields always remain ``NULL``.  For :ref:`dynamically
   allocated objects <heap-types>`, these two fields are used to link the
   object into a doubly-linked list of *all* live objects on the heap.

   This could be used for various debugging purposes; currently the only uses
   are the :func:`sys.getobjects` function and to print the objects that are
   still alive at the end of a run when the environment variable
   :envvar:`PYTHONDUMPREFS` is set.

   **Inheritance:**

   These fields are not inherited by subtypes.
```

PyVarObject Slots

```
.. c:member:: Py_ssize_t PyVarObject.ob_size

   For :ref:`statically allocated type objects <static-types>`, this should be
   initialized to zero. For :ref:`dynamically allocated type objects
   <heap-types>`, this field has a special internal meaning.

   **Inheritance:**

   This field is not inherited by subtypes.
```

PyTypeObject Slots

Each slot has a section describing inheritance. If :c:func:`PyType_Ready` may set a value when the field is set to NULL then there will also be a "Default" section. (Note that many fields set on :c:type:`PyBaseObject_Type` and :c:type:`PyType_Type` effectively act as defaults.)

```
.. c:member:: const char* PyTypeObject.tp_name

   Pointer to a NUL-terminated string containing the name of the type. For types
   that are accessible as module globals, the string should be the full module
   name, followed by a dot, followed by the type name; for built-in types, it
   should be just the type name.  If the module is a submodule of a package, the
   full package name is part of the full module name.  For example, a type named
   :class:`T` defined in module :mod:`M` in subpackage :mod:`Q` in package :mod:`P`
   should have the :c:member:`~PyTypeObject.tp_name` initializer ``"P.Q.M.T"``.

   For :ref:`dynamically allocated type objects <heap-types>`,
```

this should just be the type name, and
the module name explicitly stored in the type dict as the value for key
``'__module__'``.

For :ref:`statically allocated type objects <static-types>`,
the *tp_name* field should contain a dot.
Everything before the last dot is made accessible as the :attr:`__module__`
attribute, and everything after the last dot is made accessible as the
:attr:`~definition.__name__` attribute.

If no dot is present, the entire :c:member:`~PyTypeObject.tp_name` field is made accessible as the
:attr:`~definition.__name__` attribute, and the :attr:`__module__` attribute is undefined
(unless explicitly set in the dictionary, as explained above). This means your
type will be impossible to pickle. Additionally, it will not be listed in
module documentations created with pydoc.

This field must not be ``NULL``. It is the only required field
in :c:func:`PyTypeObject` (other than potentially
:c:member:`~PyTypeObject.tp_itemsize`).

Inheritance:

This field is not inherited by subtypes.

    .. c:member:: Py_ssize_t PyTypeObject.tp_basicsize
                  Py_ssize_t PyTypeObject.tp_itemsize

    These fields allow calculating the size in bytes of instances of the type.

    There are two kinds of types: types with fixed-length instances have a zero
    :c:member:`~PyTypeObject.tp_itemsize` field, types with variable-length instances have a non-zero
    :c:member:`~PyTypeObject.tp_itemsize` field. For a type with fixed-length instances, all
    instances have the same size, given in :c:member:`~PyTypeObject.tp_basicsize`.

    For a type with variable-length instances, the instances must have an
    :attr:`ob_size` field, and the instance size is :c:member:`~PyTypeObject.tp_basicsize` plus N
    times :c:member:`~PyTypeObject.tp_itemsize`, where N is the "length" of the object. The value of
    N is typically stored in the instance's :attr:`ob_size` field. There are
    exceptions: for example, ints use a negative :attr:`ob_size` to indicate a
    negative number, and N is ``abs(ob_size)`` there. Also, the presence of an
    :attr:`ob_size` field in the instance layout doesn't mean that the instance
    structure is variable-length (for example, the structure for the list type has
    fixed-length instances, yet those instances have a meaningful :attr:`ob_size`
    field).

    The basic size includes the fields in the instance declared by the macro
    :c:macro:`PyObject_HEAD` or :c:macro:`PyObject_VAR_HEAD` (whichever is used to
    declare the instance struct) and this in turn includes the :attr:`_ob_prev` and
    :attr:`_ob_next` fields if they are present. This means that the only correct
    way to get an initializer for the :c:member:`~PyTypeObject.tp_basicsize` is to use the
    ``sizeof`` operator on the struct used to declare the instance layout.
    The basic size does not include the GC header size.

    A note about alignment: if the variable items require a particular alignment,
    this should be taken care of by the value of :c:member:`~PyTypeObject.tp_basicsize`. Example:
    suppose a type implements an array of ``double``. :c:member:`~PyTypeObject.tp_itemsize` is
    ``sizeof(double)``. It is the programmer's responsibility that
    :c:member:`~PyTypeObject.tp_basicsize` is a multiple of ``sizeof(double)`` (assuming this is the
    alignment requirement for ``double``).

    For any type with variable-length instances, this field must not be ``NULL``.

    Inheritance:

    These fields are inherited separately by subtypes. If the base type has a
    non-zero :c:member:`~PyTypeObject.tp_itemsize`, it is generally not safe to set
    :c:member:`~PyTypeObject.tp_itemsize` to a different non-zero value in a subtype (though this
    depends on the implementation of the base type).

    .. c:member:: destructor PyTypeObject.tp_dealloc

    A pointer to the instance destructor function. This function must be defined
    unless the type guarantees that its instances will never be deallocated (as is
    the case for the singletons ``None`` and ``Ellipsis``). The function signature is::

        void tp_dealloc(PyObject *self);

    The destructor function is called by the :c:func:`Py_DECREF` and
    :c:func:`Py_XDECREF` macros when the new reference count is zero. At this point,
    the instance is still in existence, but there are no references to it. The
    destructor function should free all references which the instance owns, free all
    memory buffers owned by the instance (using the freeing function corresponding
    to the allocation function used to allocate the buffer), and call the type's
    :c:member:`~PyTypeObject.tp_free` function. If the type is not subtypable
    (doesn't have the :const:`Py_TPFLAGS_BASETYPE` flag bit set), it is
    permissible to call the object deallocator directly instead of via
    :c:member:`~PyTypeObject.tp_free`. The object deallocator should be the one used to allocate the
    instance; this is normally :c:func:`PyObject_Del` if the instance was allocated
    using :c:func:`PyObject_New` or :c:func:`PyObject_VarNew`, or
    :c:func:`PyObject_GC_Del` if the instance was allocated using
    :c:func:`PyObject_GC_New` or :c:func:`PyObject_GC_NewVar`.

    If the type supports garbage collection (has the :const:`Py_TPFLAGS_HAVE_GC`

```
flag bit set), the destructor should call :c:func:`PyObject_GC_UnTrack`
before clearing any member fields.

.. code-block:: c

  static void foo_dealloc(foo_object *self) {
      PyObject_GC_UnTrack(self);
      Py_CLEAR(self->ref);
      Py_TYPE(self)->tp_free((PyObject *)self);
  }

Finally, if the type is heap allocated (:const:`Py_TPFLAGS_HEAPTYPE`), the
deallocator should decrement the reference count for its type object after
calling the type deallocator. In order to avoid dangling pointers, the
recommended way to achieve this is:

.. code-block:: c

  static void foo_dealloc(foo_object *self) {
      PyTypeObject *tp = Py_TYPE(self);
      // free references and buffers here
      tp->tp_free(self);
      Py_DECREF(tp);
  }


**Inheritance:**

This field is inherited by subtypes.
```

```
   .. c:member:: Py_ssize_t PyTypeObject.tp_vectorcall_offset

   An optional offset to a per-instance function that implements calling
   the object using the :ref:`vectorcall protocol <vectorcall>`,
   a more efficient alternative
   of the simpler :c:member:`~PyTypeObject.tp_call`.

   This field is only used if the flag :const:`Py_TPFLAGS_HAVE_VECTORCALL`
   is set. If so, this must be a positive integer containing the offset in the
   instance of a :c:type:`vectorcallfunc` pointer.

   The *vectorcallfunc* pointer may be ``NULL``, in which case the instance behaves
   as if :const:`Py_TPFLAGS_HAVE_VECTORCALL` was not set: calling the instance
   falls back to :c:member:`~PyTypeObject.tp_call`.

   Any class that sets ``Py_TPFLAGS_HAVE_VECTORCALL`` must also set
   :c:member:`~PyTypeObject.tp_call` and make sure its behaviour is consistent
   with the *vectorcallfunc* function.
   This can be done by setting *tp_call* to :c:func:`PyVectorcall_Call`.

   .. warning::

      It is not recommended for :ref:`mutable heap types <heap-types>` to implement
      the vectorcall protocol.
      When a user sets :attr:`__call__` in Python code, only *tp_call* is updated,
      likely making it inconsistent with the vectorcall function.

   .. note::

      The semantics of the ``tp_vectorcall_offset`` slot are provisional and
      expected to be finalized in Python 3.9.
      If you use vectorcall, plan for updating your code for Python 3.9.

   .. versionchanged:: 3.8

      Before version 3.8, this slot was named ``tp_print``.
      In Python 2.x, it was used for printing to a file.
      In Python 3.0 to 3.7, it was unused.

   **Inheritance:**

   This field is always inherited.
   However, the :const:`Py_TPFLAGS_HAVE_VECTORCALL` flag is not
   always inherited. If it's not, then the subclass won't use
   :ref:`vectorcall <vectorcall>`, except when
   :c:func:`PyVectorcall_Call` is explicitly called.
   This is in particular the case for types without the
   :const:`Py_TPFLAGS_IMMUTABLETYPE` flag set (including subclasses defined in
   Python).
```

```
   .. c:member:: getattrfunc PyTypeObject.tp_getattr

   An optional pointer to the get-attribute-string function.

   This field is deprecated.  When it is defined, it should point to a function
   that acts the same as the :c:member:`~PyTypeObject.tp_getattro` function, but taking a C string
   instead of a Python string object to give the attribute name.

   **Inheritance:**

   Group: :attr:`tp_getattr`, :attr:`tp_getattro`

   This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_getattro`: a subtype
```

```
inherits both :c:member:`~PyTypeObject.tp_getattr` and :c:member:`~PyTypeObject.tp_getattro` from its base type
the subtype's :c:member:`~PyTypeObject.tp_getattr` and :c:member:`~PyTypeObject.tp_getattro` are both ``NULL``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 771)

Unknown directive type "c:member".

```
.. c:member:: setattrfunc PyTypeObject.tp_setattr

   An optional pointer to the function for setting and deleting attributes.

   This field is deprecated.  When it is defined, it should point to a function
   that acts the same as the :c:member:`~PyTypeObject.tp_setattro` function, but taking a C string
   instead of a Python string object to give the attribute name.

   **Inheritance:**

   Group: :attr:`tp_setattr`, :attr:`tp_setattro`

   This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_setattro`: a subtype
   inherits both :c:member:`~PyTypeObject.tp_setattr` and :c:member:`~PyTypeObject.tp_setattro` from its base type
   the subtype's :c:member:`~PyTypeObject.tp_setattr` and :c:member:`~PyTypeObject.tp_setattro` are both ``NULL``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 788)

Unknown directive type "c:member".

```
.. c:member:: PyAsyncMethods* PyTypeObject.tp_as_async

   Pointer to an additional structure that contains fields relevant only to
   objects which implement :term:`awaitable` and :term:`asynchronous iterator`
   protocols at the C-level.  See :ref:`async-structs` for details.

   .. versionadded:: 3.5
      Formerly known as ``tp_compare`` and ``tp_reserved``.

   **Inheritance:**

   The :c:member:`~PyTypeObject.tp_as_async` field is not inherited,
   but the contained fields are inherited individually.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 803)

Unknown directive type "c:member".

```
.. c:member:: reprfunc PyTypeObject.tp_repr

   .. index:: builtin: repr

   An optional pointer to a function that implements the built-in function
   :func:`repr`.

   The signature is the same as for :c:func:`PyObject_Repr`::

      PyObject *tp_repr(PyObject *self);

   The function must return a string or a Unicode object.  Ideally,
   this function should return a string that, when passed to
   :func:`eval`, given a suitable environment, returns an object with the
   same value.  If this is not feasible, it should return a string starting with
   ``'<'`` and ending with ``'>'`` from which both the type and the value of the
   object can be deduced.

   **Inheritance:**

   This field is inherited by subtypes.

   **Default:**

   When this field is not set, a string of the form ``<%s object at %p>`` is
   returned, where ``%s`` is replaced by the type name, and ``%p`` by the object's
   memory address.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 832)

Unknown directive type "c:member".

```
.. c:member:: PyNumberMethods* PyTypeObject.tp_as_number

   Pointer to an additional structure that contains fields relevant only to
   objects which implement the number protocol.  These fields are documented in
   :ref:`number-structs`.

   **Inheritance:**

   The :c:member:`~PyTypeObject.tp_as_number` field is not inherited, but the contained fields are
   inherited individually.
```

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

Unknown directive type "c:member".

```
.. c:member:: PySequenceMethods* PyTypeObject.tp_as_sequence

   Pointer to an additional structure that contains fields relevant only to
   objects which implement the sequence protocol.  These fields are documented
   in :ref:`sequence-structs`.

   **Inheritance:**

   The :c:member:`~PyTypeObject.tp_as_sequence` field is not inherited, but the contained fields
   are inherited individually.
```

Unknown directive type "c:member".

```
.. c:member:: PyMappingMethods* PyTypeObject.tp_as_mapping

   Pointer to an additional structure that contains fields relevant only to
   objects which implement the mapping protocol.  These fields are documented in
   :ref:`mapping-structs`.

   **Inheritance:**

   The :c:member:`~PyTypeObject.tp_as_mapping` field is not inherited, but the contained fields
   are inherited individually.
```

Unknown directive type "c:member".

```
.. c:member:: hashfunc PyTypeObject.tp_hash

   .. index:: builtin: hash

   An optional pointer to a function that implements the built-in function
   :func:`hash`.

   The signature is the same as for :c:func:`PyObject_Hash`::

      Py_hash_t tp_hash(PyObject *);

   The value ``-1`` should not be returned as a
   normal return value; when an error occurs during the computation of the hash
   value, the function should set an exception and return ``-1``.

   When this field is not set (*and* :attr:`tp_richcompare` is not set),
   an attempt to take the hash of the object raises :exc:`TypeError`.
   This is the same as setting it to :c:func:`PyObject_HashNotImplemented`.

   This field can be set explicitly to :c:func:`PyObject_HashNotImplemented` to
   block inheritance of the hash method from a parent type. This is interpreted
   as the equivalent of ``__hash__ = None`` at the Python level, causing
   ``isinstance(o, collections.Hashable)`` to correctly return ``False``. Note
   that the converse is also true - setting ``__hash__ = None`` on a class at
   the Python level will result in the ``tp_hash`` slot being set to
   :c:func:`PyObject_HashNotImplemented`.

   **Inheritance:**

   Group: :attr:`tp_hash`, :attr:`tp_richcompare`

   This field is inherited by subtypes together with
   :c:member:`~PyTypeObject.tp_richcompare`: a subtype inherits both of
   :c:member:`~PyTypeObject.tp_richcompare` and :c:member:`~PyTypeObject.tp_hash`, when the subtype's
   :c:member:`~PyTypeObject.tp_richcompare` and :c:member:`~PyTypeObject.tp_hash` are both ``NULL``.
```

Unknown directive type "c:member".

```
.. c:member:: ternaryfunc PyTypeObject.tp_call

   An optional pointer to a function that implements calling the object.  This
   should be ``NULL`` if the object is not callable.  The signature is the same as
   for :c:func:`PyObject_Call`::

      PyObject *tp_call(PyObject *self, PyObject *args, PyObject *kwargs);

   **Inheritance:**

   This field is inherited by subtypes.
```

Unknown directive type "c:member".

```
.. c:member:: reprfunc PyTypeObject.tp_str

   An optional pointer to a function that implements the built-in operation
```

:func:`str`. (Note that :class:`str` is a type now, and :func:`str` calls the
constructor for that type. This constructor calls :c:func:`PyObject_Str` to do
the actual work, and :c:func:`PyObject_Str` will call this handler.)

The signature is the same as for :c:func:`PyObject_Str`::

    PyObject *tp_str(PyObject *self);

The function must return a string or a Unicode object. It should be a "friendly" string
representation of the object, as this is the representation that will be used,
among other things, by the :func:`print` function.

Inheritance:

This field is inherited by subtypes.

Default:

When this field is not set, :c:func:`PyObject_Repr` is called to return a string
representation.

```
.. c:member:: getattrofunc PyTypeObject.tp_getattro

   An optional pointer to the get-attribute function.

   The signature is the same as for :c:func:`PyObject_GetAttr`::

      PyObject *tp_getattro(PyObject *self, PyObject *attr);

   It is usually convenient to set this field to :c:func:`PyObject_GenericGetAttr`,
   which implements the normal way of looking for object attributes.

   **Inheritance:**

   Group: :attr:`tp_getattr`, :attr:`tp_getattro`

   This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_getattr`: a subtype
   inherits both :c:member:`~PyTypeObject.tp_getattr` and :c:member:`~PyTypeObject.tp_getattro` from its base type w
   the subtype's :c:member:`~PyTypeObject.tp_getattr` and :c:member:`~PyTypeObject.tp_getattro` are both ``NULL``.

   **Default:**

   :c:type:`PyBaseObject_Type` uses :c:func:`PyObject_GenericGetAttr`.
```

```
.. c:member:: setattrofunc PyTypeObject.tp_setattro

   An optional pointer to the function for setting and deleting attributes.

   The signature is the same as for :c:func:`PyObject_SetAttr`::

      int tp_setattro(PyObject *self, PyObject *attr, PyObject *value);

   In addition, setting *value* to ``NULL`` to delete an attribute must be
   supported.  It is usually convenient to set this field to
   :c:func:`PyObject_GenericSetAttr`, which implements the normal
   way of setting object attributes.

   **Inheritance:**

   Group: :attr:`tp_setattr`, :attr:`tp_setattro`

   This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_setattr`: a subtype
   inherits both :c:member:`~PyTypeObject.tp_setattr` and :c:member:`~PyTypeObject.tp_setattro` from its base type w
   the subtype's :c:member:`~PyTypeObject.tp_setattr` and :c:member:`~PyTypeObject.tp_setattro` are both ``NULL``.

   **Default:**

   :c:type:`PyBaseObject_Type` uses :c:func:`PyObject_GenericSetAttr`.
```

```
.. c:member:: PyBufferProcs* PyTypeObject.tp_as_buffer

   Pointer to an additional structure that contains fields relevant only to objects
   which implement the buffer interface.  These fields are documented in
   :ref:`buffer-structs`.

   **Inheritance:**

   The :c:member:`~PyTypeObject.tp_as_buffer` field is not inherited,
   but the contained fields are inherited individually.
```

Unknown directive type "c:member".

```
.. c:member:: unsigned long PyTypeObject.tp_flags

   This field is a bit mask of various flags.  Some flags indicate variant
   semantics for certain situations; others are used to indicate that certain
   fields in the type object (or in the extension structures referenced via
   :c:member:`~PyTypeObject.tp_as_number`, :c:member:`~PyTypeObject.tp_as_sequence`, :c:member:`~PyTypeObject.tp_as_
   :c:member:`~PyTypeObject.tp_as_buffer`) that were historically not always present are valid; if
   such a flag bit is clear, the type fields it guards must not be accessed and
   must be considered to have a zero or ``NULL`` value instead.

   **Inheritance:**

   Inheritance of this field is complicated.  Most flag bits are inherited
   individually, i.e. if the base type has a flag bit set, the subtype inherits
   this flag bit.  The flag bits that pertain to extension structures are strictly
   inherited if the extension structure is inherited, i.e. the base type's value of
   the flag bit is copied into the subtype together with a pointer to the extension
   structure.  The :const:`Py_TPFLAGS_HAVE_GC` flag bit is inherited together with
   the :c:member:`~PyTypeObject.tp_traverse` and :c:member:`~PyTypeObject.tp_clear` fields, i.e. if the
   :const:`Py_TPFLAGS_HAVE_GC` flag bit is clear in the subtype and the
   :c:member:`~PyTypeObject.tp_traverse` and :c:member:`~PyTypeObject.tp_clear` fields in the subtype exist and have
   ``NULL`` values.

   .. XXX are most flag bits *really* inherited individually?

   **Default:**

   :c:type:`PyBaseObject_Type` uses
   ``Py_TPFLAGS_DEFAULT | Py_TPFLAGS_BASETYPE``.

   **Bit Masks:**

   The following bit masks are currently defined; these can be ORed together using
   the ``|`` operator to form the value of the :c:member:`~PyTypeObject.tp_flags` field.  The macro
   :c:func:`PyType_HasFeature` takes a type and a flags value, *tp* and *f*, and
   checks whether ``tp->tp_flags & f`` is non-zero.

   .. data:: Py_TPFLAGS_HEAPTYPE

      This bit is set when the type object itself is allocated on the heap, for
      example, types created dynamically using :c:func:`PyType_FromSpec`.  In this
      case, the :attr:`ob_type` field of its instances is considered a reference to
      the type, and the type object is INCREF'ed when a new instance is created, and
      DECREF'ed when an instance is destroyed (this does not apply to instances of
      subtypes; only the type referenced by the instance's ob_type gets INCREF'ed or
      DECREF'ed).

      **Inheritance:**

      ???


   .. data:: Py_TPFLAGS_BASETYPE

      This bit is set when the type can be used as the base type of another type.  If
      this bit is clear, the type cannot be subtyped (similar to a "final" class in
      Java).

      **Inheritance:**

      ???


   .. data:: Py_TPFLAGS_READY

      This bit is set when the type object has been fully initialized by
      :c:func:`PyType_Ready`.

      **Inheritance:**

      ???


   .. data:: Py_TPFLAGS_READYING

      This bit is set while :c:func:`PyType_Ready` is in the process of initializing
      the type object.

      **Inheritance:**

      ???


   .. data:: Py_TPFLAGS_HAVE_GC

      This bit is set when the object supports garbage collection.  If this bit
      is set, instances must be created using :c:func:`PyObject_GC_New` and
      destroyed using :c:func:`PyObject_GC_Del`.  More information in section
      :ref:`supporting-cycle-detection`.  This bit also implies that the
      GC-related fields :c:member:`~PyTypeObject.tp_traverse` and :c:member:`~PyTypeObject.tp_clear` are present in
      the type object.

      **Inheritance:**

      Group: :const:`Py_TPFLAGS_HAVE_GC`, :attr:`tp_traverse`, :attr:`tp_clear`

      The :const:`Py_TPFLAGS_HAVE_GC` flag bit is inherited
      together with the :attr:`tp_traverse` and :attr:`tp_clear`
      fields, i.e.  if the :const:`Py_TPFLAGS_HAVE_GC` flag bit is
      clear in the subtype and the :attr:`tp_traverse` and
      :attr:`tp_clear` fields in the subtype exist and have ``NULL``
      values.


   .. data:: Py_TPFLAGS_DEFAULT
```

This is a bitmask of all the bits that pertain to the existence of certain
fields in the type object and its extension structures. Currently, it includes
the following bits: :const:`Py_TPFLAGS_HAVE_STACKLESS_EXTENSION`.

Inheritance:

???

.. data:: Py_TPFLAGS_METHOD_DESCRIPTOR

   This bit indicates that objects behave like unbound methods.

   If this flag is set for ``type(meth)``, then:

   - ``meth.__get__(obj, cls)(*args, **kwds)`` (with ``obj`` not None)
     must be equivalent to ``meth(obj, *args, **kwds)``.

   - ``meth.__get__(None, cls)(*args, **kwds)``
     must be equivalent to ``meth(*args, **kwds)``.

   This flag enables an optimization for typical method calls like
   ``obj.meth()``: it avoids creating a temporary "bound method" object for
   ``obj.meth``.

   .. versionadded:: 3.8

   Inheritance:

   This flag is never inherited by types without the
   :const:`Py_TPFLAGS_IMMUTABLETYPE` flag set. For extension types, it is
   inherited whenever :c:member:`~PyTypeObject.tp_descr_get` is inherited.

.. XXX Document more flags here?

.. data:: Py_TPFLAGS_LONG_SUBCLASS
.. data:: Py_TPFLAGS_LIST_SUBCLASS
.. data:: Py_TPFLAGS_TUPLE_SUBCLASS
.. data:: Py_TPFLAGS_BYTES_SUBCLASS
.. data:: Py_TPFLAGS_UNICODE_SUBCLASS
.. data:: Py_TPFLAGS_DICT_SUBCLASS
.. data:: Py_TPFLAGS_BASE_EXC_SUBCLASS
.. data:: Py_TPFLAGS_TYPE_SUBCLASS

   These flags are used by functions such as
   :c:func:`PyLong_Check` to quickly determine if a type is a subclass
   of a built-in type; such specific checks are faster than a generic
   check, like :c:func:`PyObject_IsInstance`. Custom types that inherit
   from built-ins should have their :c:member:`~PyTypeObject.tp_flags`
   set appropriately, or the code that interacts with such types
   will behave differently depending on what kind of check is used.

.. data:: Py_TPFLAGS_HAVE_FINALIZE

   This bit is set when the :c:member:`~PyTypeObject.tp_finalize` slot is present in the
   type structure.

   .. versionadded:: 3.4

   .. deprecated:: 3.8
      This flag isn't necessary anymore, as the interpreter assumes the
      :c:member:`~PyTypeObject.tp_finalize` slot is always present in the
      type structure.

.. data:: Py_TPFLAGS_HAVE_VECTORCALL

   This bit is set when the class implements
   the :ref:`vectorcall protocol <vectorcall>`.
   See :c:member:`~PyTypeObject.tp_vectorcall_offset` for details.

   Inheritance:

   This bit is inherited for types with the
   :const:`Py_TPFLAGS_IMMUTABLETYPE` flag set, if
   :c:member:`~PyTypeObject.tp_call` is also inherited.

   .. versionadded:: 3.9

.. data:: Py_TPFLAGS_IMMUTABLETYPE

   This bit is set for type objects that are immutable: type attributes cannot be set nor deleted.

   :c:func:`PyType_Ready` automatically applies this flag to
   :ref:`static types <static-types>`.

   Inheritance:

   This flag is not inherited.

   .. versionadded:: 3.10

.. data:: Py_TPFLAGS_DISALLOW_INSTANTIATION

   Disallow creating instances of the type: set
   :c:member:`~PyTypeObject.tp_new` to NULL and don't create the ``__new__``
   key in the type dictionary.

   The flag must be set before creating the type, not after. For example, it
   must be set before :c:func:`PyType_Ready` is called on the type.

   The flag is set automatically on :ref:`static types <static-types>` if
   :c:member:`~PyTypeObject.tp_base` is NULL or ``&PyBaseObject_Type`` and
   :c:member:`~PyTypeObject.tp_new` is NULL.

Inheritance:

This flag is not inherited.

.. versionadded:: 3.10

.. data:: Py_TPFLAGS_MAPPING

This bit indicates that instances of the class may match mapping patterns
when used as the subject of a :keyword:`match` block. It is automatically
set when registering or subclassing :class:`collections.abc.Mapping`, and
unset when registering :class:`collections.abc.Sequence`.

.. note::

   :const:`Py_TPFLAGS_MAPPING` and :const:`Py_TPFLAGS_SEQUENCE` are
   mutually exclusive; it is an error to enable both flags simultaneously.

Inheritance:

This flag is inherited by types that do not already set
:const:`Py_TPFLAGS_SEQUENCE`.

.. seealso:: :pep:`634` -- Structural Pattern Matching: Specification

.. versionadded:: 3.10

.. data:: Py_TPFLAGS_SEQUENCE

This bit indicates that instances of the class may match sequence patterns
when used as the subject of a :keyword:`match` block. It is automatically
set when registering or subclassing :class:`collections.abc.Sequence`, and
unset when registering :class:`collections.abc.Mapping`.

.. note::

   :const:`Py_TPFLAGS_MAPPING` and :const:`Py_TPFLAGS_SEQUENCE` are
   mutually exclusive; it is an error to enable both flags simultaneously.

Inheritance:

This flag is inherited by types that do not already set
:const:`Py_TPFLAGS_MAPPING`.

.. seealso:: :pep:`634` -- Structural Pattern Matching: Specification

.. versionadded:: 3.10

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst`, line 1270)

Unknown directive type "c:member".

```
.. c:member:: const char* PyTypeObject.tp_doc

   An optional pointer to a NUL-terminated C string giving the docstring for this
   type object.  This is exposed as the :attr:`__doc__` attribute on the type and
   instances of the type.

   **Inheritance:**

   This field is *not* inherited by subtypes.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main) (Doc) (c-api) typeobj.rst`, line 1281)

Unknown directive type "c:member".

```
.. c:member:: traverseproc PyTypeObject.tp_traverse

   An optional pointer to a traversal function for the garbage collector.  This is
   only used if the :const:`Py_TPFLAGS_HAVE_GC` flag bit is set.  The signature is::

      int tp_traverse(PyObject *self, visitproc visit, void *arg);

   More information about Python's garbage collection scheme can be found
   in section :ref:`supporting-cycle-detection`.

   The :c:member:`~PyTypeObject.tp_traverse` pointer is used by the garbage collector to detect
   reference cycles. A typical implementation of a :c:member:`~PyTypeObject.tp_traverse` function
   simply calls :c:func:`Py_VISIT` on each of the instance's members that are Python
   objects that the instance owns. For example, this is function :c:func:`local_traverse` from the
   :mod:`_thread` extension module::

      static int
      local_traverse(localobject *self, visitproc visit, void *arg)
      {
          Py_VISIT(self->args);
          Py_VISIT(self->kw);
          Py_VISIT(self->dict);
          return 0;
      }

   Note that :c:func:`Py_VISIT` is called only on those members that can participate
   in reference cycles.  Although there is also a ``self->key`` member, it can only
   be ``NULL`` or a Python string and therefore cannot be part of a reference cycle.

   On the other hand, even if you know a member can never be part of a cycle, as a
   debugging aid you may want to visit it anyway just so the :mod:`gc` module's
   :func:`~gc.get_referents` function will include it.
```

```
.. warning::
    When implementing :c:member:`~PyTypeObject.tp_traverse`, only the
    members that the instance *owns* (by having :term:`strong references
    <strong reference>` to them) must be
    visited. For instance, if an object supports weak references via the
    :c:member:`~PyTypeObject.tp_weaklist` slot, the pointer supporting
    the linked list (what *tp_weaklist* points to) must **not** be
    visited as the instance does not directly own the weak references to itself
    (the weakreference list is there to support the weak reference machinery,
    but the instance has no strong reference to the elements inside it, as they
    are allowed to be removed even if the instance is still alive).

Note that :c:func:`Py_VISIT` requires the *visit* and *arg* parameters to
:c:func:`local_traverse` to have these specific names; don't name them just
anything.

Instances of :ref:`heap-allocated types <heap-types>` hold a reference to
their type. Their traversal function must therefore either visit
:c:func:`Py_TYPE(self) <Py_TYPE>`, or delegate this responsibility by
calling ``tp_traverse`` of another heap-allocated type (such as a
heap-allocated superclass).
If they do not, the type object may not be garbage-collected.

.. versionchanged:: 3.9

    Heap-allocated types are expected to visit ``Py_TYPE(self)`` in
    ``tp_traverse``.  In earlier versions of Python, due to
    `bug 40217 <https://bugs.python.org/issue40217>`_, doing this
    may lead to crashes in subclasses.

**Inheritance:**

Group: :const:`Py_TPFLAGS_HAVE_GC`, :attr:`tp_traverse`, :attr:`tp_clear`

This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_clear` and the
:const:`Py_TPFLAGS_HAVE_GC` flag bit: the flag bit, :c:member:`~PyTypeObject.tp_traverse`, and
:c:member:`~PyTypeObject.tp_clear` are all inherited from the base type if they are all zero in
the subtype.
```

```
.. c:member:: inquiry PyTypeObject.tp_clear

An optional pointer to a clear function for the garbage collector. This is only
used if the :const:`Py_TPFLAGS_HAVE_GC` flag bit is set.  The signature is::

    int tp_clear(PyObject *);

The :c:member:`~PyTypeObject.tp_clear` member function is used to break reference cycles in cyclic
garbage detected by the garbage collector.  Taken together, all :c:member:`~PyTypeObject.tp_clear`
functions in the system must combine to break all reference cycles.  This is
subtle, and if in any doubt supply a :c:member:`~PyTypeObject.tp_clear` function.  For example,
the tuple type does not implement a :c:member:`~PyTypeObject.tp_clear` function, because it's
possible to prove that no reference cycle can be composed entirely of tuples.
Therefore the :c:member:`~PyTypeObject.tp_clear` functions of other types must be sufficient to
break any cycle containing a tuple.  This isn't immediately obvious, and there's
rarely a good reason to avoid implementing :c:member:`~PyTypeObject.tp_clear`.

Implementations of :c:member:`~PyTypeObject.tp_clear` should drop the instance's references to
those of its members that may be Python objects, and set its pointers to those
members to ``NULL``, as in the following example::

    static int
    local_clear(localobject *self)
    {
        Py_CLEAR(self->key);
        Py_CLEAR(self->args);
        Py_CLEAR(self->kw);
        Py_CLEAR(self->dict);
        return 0;
    }

The :c:func:`Py_CLEAR` macro should be used, because clearing references is
delicate:  the reference to the contained object must not be decremented until
after the pointer to the contained object is set to ``NULL``.  This is because
decrementing the reference count may cause the contained object to become trash,
triggering a chain of reclamation activity that may include invoking arbitrary
Python code (due to finalizers, or weakref callbacks, associated with the
contained object). If it's possible for such code to reference *self* again,
it's important that the pointer to the contained object be ``NULL`` at that time,
so that *self* knows the contained object can no longer be used.  The
:c:func:`Py_CLEAR` macro performs the operations in a safe order.

Note that :c:member:`~PyTypeObject.tp_clear` is not *always* called
before an instance is deallocated. For example, when reference counting
is enough to determine that an object is no longer used, the cyclic garbage
collector is not involved and :c:member:`~PyTypeObject.tp_dealloc` is
called directly.

Because the goal of :c:member:`~PyTypeObject.tp_clear` functions is to break reference cycles,
it's not necessary to clear contained objects like Python strings or Python
integers, which can't participate in reference cycles. On the other hand, it may
be convenient to clear all contained Python objects, and write the type's
:c:member:`~PyTypeObject.tp_dealloc` function to invoke :c:member:`~PyTypeObject.tp_clear`.

More information about Python's garbage collection scheme can be found in
section :ref:`supporting-cycle-detection`.

**Inheritance:**

Group: :const:`Py_TPFLAGS_HAVE_GC`, :attr:`tp_traverse`, :attr:`tp_clear`

This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_traverse` and the
```

:const:`Py_TPFLAGS_HAVE_GC` flag bit: the flag bit, :c:member:`~PyTypeObject.tp_traverse`, and
        :c:member:`~PyTypeObject.tp_clear` are all inherited from the base type if they are all zero in
        the subtype.

    .. c:member:: richcmpfunc PyTypeObject.tp_richcompare

        An optional pointer to the rich comparison function, whose signature is::

            PyObject *tp_richcompare(PyObject *self, PyObject *other, int op);

        The first parameter is guaranteed to be an instance of the type
        that is defined by :c:type:`PyTypeObject`.

        The function should return the result of the comparison (usually ``Py_True``
        or ``Py_False``). If the comparison is undefined, it must return
        ``Py_NotImplemented``, if another error occurred it must return ``NULL`` and
        set an exception condition.

        The following constants are defined to be used as the third argument for
        :c:member:`~PyTypeObject.tp_richcompare` and for :c:func:`PyObject_RichCompare`:

        +-----------------+------------+
        | Constant | Comparison |
        +=================+============+
        | :const:`Py_LT` | ``<`` |
        +-----------------+------------+
        | :const:`Py_LE` | ``<=`` |
        +-----------------+------------+
        | :const:`Py_EQ` | ``==`` |
        +-----------------+------------+
        | :const:`Py_NE` | ``!=`` |
        +-----------------+------------+
        | :const:`Py_GT` | ``>`` |
        +-----------------+------------+
        | :const:`Py_GE` | ``>=`` |
        +-----------------+------------+

        The following macro is defined to ease writing rich comparison functions:

        .. c:macro:: Py_RETURN_RICHCOMPARE(VAL_A, VAL_B, op)

            Return ``Py_True`` or ``Py_False`` from the function, depending on the
            result of a comparison.
            VAL_A and VAL_B must be orderable by C comparison operators (for example,
            they may be C ints or floats). The third argument specifies the requested
            operation, as for :c:func:`PyObject_RichCompare`.

            The return value's reference count is properly incremented.

            On error, sets an exception and returns ``NULL`` from the function.

            .. versionadded:: 3.7

        Inheritance:

        Group: :attr:`tp_hash`, :attr:`tp_richcompare`

        This field is inherited by subtypes together with :c:member:`~PyTypeObject.tp_hash`:
        a subtype inherits :c:member:`~PyTypeObject.tp_richcompare` and :c:member:`~PyTypeObject.tp_hash` when
        the subtype's :c:member:`~PyTypeObject.tp_richcompare` and :c:member:`~PyTypeObject.tp_hash` are both
        ``NULL``.

        Default:

        :c:type:`PyBaseObject_Type` provides a :attr:`tp_richcompare`
        implementation, which may be inherited. However, if only
        :attr:`tp_hash` is defined, not even the inherited function is used
        and instances of the type will not be able to participate in any
        comparisons.

    .. c:member:: Py_ssize_t PyTypeObject.tp_weaklistoffset

        If the instances of this type are weakly referenceable, this field is greater
        than zero and contains the offset in the instance structure of the weak
        reference list head (ignoring the GC header, if present); this offset is used by
        :c:func:`PyObject_ClearWeakRefs` and the :c:func:`PyWeakref_\*` functions. The
        instance structure needs to include a field of type :c:type:`PyObject*` which is
        initialized to ``NULL``.

        Do not confuse this field with :c:member:`~PyTypeObject.tp_weaklist`; that is the list head for
        weak references to the type object itself.

        Inheritance:

        This field is inherited by subtypes, but see the rules listed below. A subtype
        may override this offset; this means that the subtype uses a different weak
        reference list head than the base type. Since the list head is always found via
        :c:member:`~PyTypeObject.tp_weaklistoffset`, this should not be a problem.

        When a type defined by a class statement has no :attr:`~object.__slots__` declaration,
        and none of its base types are weakly referenceable, the type is made weakly
        referenceable by adding a weak reference list head slot to the instance layout
        and setting the :c:member:`~PyTypeObject.tp_weaklistoffset` of that slot's offset.

When a type's :attr:`__slots__` declaration contains a slot named
:attr:`__weakref__`, that slot becomes the weak reference list head for
instances of the type, and the slot's offset is stored in the type's
:c:member:`~PyTypeObject.tp_weaklistoffset`.

When a type's :attr:`__slots__` declaration does not contain a slot named
:attr:`__weakref__`, the type inherits its :c:member:`~PyTypeObject.tp_weaklistoffset` from its
base type.

Unknown directive type "c:member".

```
.. c:member:: getiterfunc PyTypeObject.tp_iter

    An optional pointer to a function that returns an :term:`iterator` for the
    object.  Its presence normally signals that the instances of this type are
    :term:`iterable` (although sequences may be iterable without this function).

    This function has the same signature as :c:func:`PyObject_GetIter`::

        PyObject *tp_iter(PyObject *self);

    **Inheritance:**

    This field is inherited by subtypes.
```

Unknown directive type "c:member".

```
.. c:member:: iternextfunc PyTypeObject.tp_iternext

    An optional pointer to a function that returns the next item in an
    :term:`iterator`. The signature is::

        PyObject *tp_iternext(PyObject *self);

    When the iterator is exhausted, it must return ``NULL``; a :exc:`StopIteration`
    exception may or may not be set.  When another error occurs, it must return
    ``NULL`` too.  Its presence signals that the instances of this type are
    iterators.

    Iterator types should also define the :c:member:`~PyTypeObject.tp_iter` function, and that
    function should return the iterator instance itself (not a new iterator
    instance).

    This function has the same signature as :c:func:`PyIter_Next`.

    **Inheritance:**

    This field is inherited by subtypes.
```

Unknown directive type "c:member".

```
.. c:member:: struct PyMethodDef* PyTypeObject.tp_methods

    An optional pointer to a static ``NULL``-terminated array of :c:type:`PyMethodDef`
    structures, declaring regular methods of this type.

    For each entry in the array, an entry is added to the type's dictionary (see
    :c:member:`~PyTypeObject.tp_dict` below) containing a method descriptor.

    **Inheritance:**

    This field is not inherited by subtypes (methods are inherited through a
    different mechanism).
```

Unknown directive type "c:member".

```
.. c:member:: struct PyMemberDef* PyTypeObject.tp_members

    An optional pointer to a static ``NULL``-terminated array of :c:type:`PyMemberDef`
    structures, declaring regular data members (fields or slots) of instances of
    this type.

    For each entry in the array, an entry is added to the type's dictionary (see
    :c:member:`~PyTypeObject.tp_dict` below) containing a member descriptor.

    **Inheritance:**

    This field is not inherited by subtypes (members are inherited through a
    different mechanism).
```

Unknown directive type "c:member".

```
.. c:member:: struct PyGetSetDef* PyTypeObject.tp_getset

   An optional pointer to a static ``NULL``-terminated array of :c:type:`PyGetSetDef`
   structures, declaring computed attributes of instances of this type.

   For each entry in the array, an entry is added to the type's dictionary (see
   :c:member:`~PyTypeObject.tp_dict` below) containing a getset descriptor.

   **Inheritance:**

   This field is not inherited by subtypes (computed attributes are inherited
   through a different mechanism).
```

Unknown directive type "c:member".

```
.. c:member:: PyTypeObject* PyTypeObject.tp_base

   An optional pointer to a base type from which type properties are inherited.  At
   this level, only single inheritance is supported; multiple inheritance require
   dynamically creating a type object by calling the metatype.

   .. note::

      .. from Modules/xxmodule.c

      Slot initialization is subject to the rules of initializing globals.
      C99 requires the initializers to be "address constants".  Function
      designators like :c:func:`PyType_GenericNew`, with implicit conversion
      to a pointer, are valid C99 address constants.

      However, the unary '&' operator applied to a non-static variable
      like :c:func:`PyBaseObject_Type` is not required to produce an address
      constant.  Compilers may support this (gcc does), MSVC does not.
      Both compilers are strictly standard conforming in this particular
      behavior.

      Consequently, :c:member:`~PyTypeObject.tp_base` should be set in
      the extension module's init function.

   **Inheritance:**

   This field is not inherited by subtypes (obviously).

   **Default:**

   This field defaults to ``&PyBaseObject_Type`` (which to Python
   programmers is known as the type :class:`object`).
```

Unknown directive type "c:member".

```
.. c:member:: PyObject* PyTypeObject.tp_dict

   The type's dictionary is stored here by :c:func:`PyType_Ready`.

   This field should normally be initialized to ``NULL`` before PyType_Ready is
   called; it may also be initialized to a dictionary containing initial attributes
   for the type.  Once :c:func:`PyType_Ready` has initialized the type, extra
   attributes for the type may be added to this dictionary only if they don't
   correspond to overloaded operations (like :meth:`__add__`).

   **Inheritance:**

   This field is not inherited by subtypes (though the attributes defined in here
   are inherited through a different mechanism).

   **Default:**

   If this field is ``NULL``, :c:func:`PyType_Ready` will assign a new
   dictionary to it.

   .. warning::

      It is not safe to use :c:func:`PyDict_SetItem` on or otherwise modify
      :c:member:`~PyTypeObject.tp_dict` with the dictionary C-API.
```

Unknown directive type "c:member".

```
.. c:member:: descrgetfunc PyTypeObject.tp_descr_get

   An optional pointer to a "descriptor get" function.

   The function signature is::

      PyObject * tp_descr_get(PyObject *self, PyObject *obj, PyObject *type);
```

```
.. XXX explain more?

**Inheritance:**

This field is inherited by subtypes.
```

```
.. c:member:: descrsetfunc PyTypeObject.tp_descr_set

    An optional pointer to a function for setting and deleting
    a descriptor's value.

    The function signature is::

        int tp_descr_set(PyObject *self, PyObject *obj, PyObject *value);

    The *value* argument is set to ``NULL`` to delete the value.

    .. XXX explain more?

    **Inheritance:**

    This field is inherited by subtypes.
```

```
.. c:member:: Py_ssize_t PyTypeObject.tp_dictoffset

    If the instances of this type have a dictionary containing instance variables,
    this field is non-zero and contains the offset in the instances of the type of
    the instance variable dictionary; this offset is used by
    :c:func:`PyObject_GenericGetAttr`.

    Do not confuse this field with :c:member:`~PyTypeObject.tp_dict`; that is the dictionary for
    attributes of the type object itself.

    If the value of this field is greater than zero, it specifies the offset from
    the start of the instance structure.  If the value is less than zero, it
    specifies the offset from the *end* of the instance structure.  A negative
    offset is more expensive to use, and should only be used when the instance
    structure contains a variable-length part.  This is used for example to add an
    instance variable dictionary to subtypes of :class:`str` or :class:`tuple`. Note
    that the :c:member:`~PyTypeObject.tp_basicsize` field should account for the dictionary added to
    the end in that case, even though the dictionary is not included in the basic
    object layout.  On a system with a pointer size of 4 bytes,
    :c:member:`~PyTypeObject.tp_dictoffset` should be set to ``-4`` to indicate that the dictionary is
    at the very end of the structure.

    The real dictionary offset in an instance can be computed from a negative
    :c:member:`~PyTypeObject.tp_dictoffset` as follows::

        dictoffset = tp_basicsize + abs(ob_size)*tp_itemsize + tp_dictoffset
        if dictoffset is not aligned on sizeof(void*):
            round up to sizeof(void*)

    where :c:member:`~PyTypeObject.tp_basicsize`, :c:member:`~PyTypeObject.tp_itemsize` and :c:member:`~PyTypeObject.
    taken from the type object, and :attr:`ob_size` is taken from the instance.  The
    absolute value is taken because ints use the sign of :attr:`ob_size` to
    store the sign of the number.  (There's never a need to do this calculation
    yourself; it is done for you by :c:func:`_PyObject_GetDictPtr`.)

    **Inheritance:**

    This field is inherited by subtypes, but see the rules listed below. A subtype
    may override this offset; this means that the subtype instances store the
    dictionary at a difference offset than the base type.  Since the dictionary is
    always found via :c:member:`~PyTypeObject.tp_dictoffset`, this should not be a problem.

    When a type defined by a class statement has no :attr:`~object.__slots__` declaration,
    and none of its base types has an instance variable dictionary, a dictionary
    slot is added to the instance layout and the :c:member:`~PyTypeObject.tp_dictoffset` is set to
    that slot's offset.

    When a type defined by a class statement has a :attr:`__slots__` declaration,
    the type inherits its :c:member:`~PyTypeObject.tp_dictoffset` from its base type.

    (Adding a slot named :attr:`~object.__dict__` to the :attr:`__slots__` declaration does
    not have the expected effect, it just causes confusion.  Maybe this should be
    added as a feature just like :attr:`__weakref__` though.)

    **Default:**

    This slot has no default. For :ref:`static types <static-types>`, if the
    field is ``NULL`` then no :attr:`__dict__` gets created for instances.
```

```
.. c:member:: initproc PyTypeObject.tp_init

    An optional pointer to an instance initialization function.
```

This function corresponds to the :meth:`__init__` method of classes. Like
:meth:`__init__`, it is possible to create an instance without calling
:meth:`__init__`, and it is possible to reinitialize an instance by calling its
:meth:`__init__` method again.

The function signature is::

    int tp_init(PyObject *self, PyObject *args, PyObject *kwds);

The self argument is the instance to be initialized; the *args* and *kwds*
arguments represent positional and keyword arguments of the call to
:meth:`__init__`.

The :c:member:`~PyTypeObject.tp_init` function, if not ``NULL``, is called when an instance is
created normally by calling its type, after the type's :c:member:`~PyTypeObject.tp_new` function
has returned an instance of the type. If the :c:member:`~PyTypeObject.tp_new` function returns an
instance of some other type that is not a subtype of the original type, no
:c:member:`~PyTypeObject.tp_init` function is called; if :c:member:`~PyTypeObject.tp_new` returns an instance of
subtype of the original type, the subtype's :c:member:`~PyTypeObject.tp_init` is called.

Returns ``0`` on success, ``-1`` and sets an exception on error.

Inheritance:

This field is inherited by subtypes.

Default:

For :ref:`static types <static-types>` this field does not have a default.

```
.. c:member:: allocfunc PyTypeObject.tp_alloc
```

An optional pointer to an instance allocation function.

The function signature is::

    PyObject *tp_alloc(PyTypeObject *self, Py_ssize_t nitems);

Inheritance:

This field is inherited by static subtypes, but not by dynamic
subtypes (subtypes created by a class statement).

Default:

For dynamic subtypes, this field is always set to
:c:func:`PyType_GenericAlloc`, to force a standard heap
allocation strategy.

For static subtypes, :c:type:`PyBaseObject_Type` uses
:c:func:`PyType_GenericAlloc`. That is the recommended value
for all statically defined types.

```
.. c:member:: newfunc PyTypeObject.tp_new
```

An optional pointer to an instance creation function.

The function signature is::

    PyObject *tp_new(PyTypeObject *subtype, PyObject *args, PyObject *kwds);

The *subtype* argument is the type of the object being created; the *args* and
kwds arguments represent positional and keyword arguments of the call to the
type. Note that *subtype* doesn't have to equal the type whose :c:member:`~PyTypeObject.tp_new`
function is called; it may be a subtype of that type (but not an unrelated
type).

The :c:member:`~PyTypeObject.tp_new` function should call ``subtype->tp_alloc(subtype, nitems)``
to allocate space for the object, and then do only as much further
initialization as is absolutely necessary. Initialization that can safely be
ignored or repeated should be placed in the :c:member:`~PyTypeObject.tp_init` handler. A good
rule of thumb is that for immutable types, all initialization should take place
in :c:member:`~PyTypeObject.tp_new`, while for mutable types, most initialization should be
deferred to :c:member:`~PyTypeObject.tp_init`.

Set the :const:`Py_TPFLAGS_DISALLOW_INSTANTIATION` flag to disallow creating
instances of the type in Python.

Inheritance:

This field is inherited by subtypes, except it is not inherited by
:ref:`static types <static-types>` whose :c:member:`~PyTypeObject.tp_base`
is ``NULL`` or ``&PyBaseObject_Type``.

Default:

For :ref:`static types <static-types>` this field has no default.
This means if the slot is defined as ``NULL``, the type cannot be called
to create new instances; presumably there is some other way to create
instances, like a factory function.

```
.. c:member:: freefunc PyTypeObject.tp_free

   An optional pointer to an instance deallocation function.  Its signature is::

      void tp_free(void *self);

   An initializer that is compatible with this signature is :c:func:`PyObject_Free`.

   **Inheritance:**

   This field is inherited by static subtypes, but not by dynamic
   subtypes (subtypes created by a class statement)

   **Default:**

   In dynamic subtypes, this field is set to a deallocator suitable to
   match :c:func:`PyType_GenericAlloc` and the value of the
   :const:`Py_TPFLAGS_HAVE_GC` flag bit.

   For static subtypes, :c:type:`PyBaseObject_Type` uses PyObject_Del.
```

```
.. c:member:: inquiry PyTypeObject.tp_is_gc

   An optional pointer to a function called by the garbage collector.

   The garbage collector needs to know whether a particular object is collectible
   or not.  Normally, it is sufficient to look at the object's type's
   :c:member:`~PyTypeObject.tp_flags` field, and check the :const:`Py_TPFLAGS_HAVE_GC` flag bit.  But
   some types have a mixture of statically and dynamically allocated instances, and
   the statically allocated instances are not collectible.  Such types should
   define this function; it should return ``1`` for a collectible instance, and
   ``0`` for a non-collectible instance. The signature is::

      int tp_is_gc(PyObject *self);

   (The only example of this are types themselves.  The metatype,
   :c:data:`PyType_Type`, defines this function to distinguish between statically
   and :ref:`dynamically allocated types <heap-types>`.)

   **Inheritance:**

   This field is inherited by subtypes.

   **Default:**

   This slot has no default.  If this field is ``NULL``,
   :const:`Py_TPFLAGS_HAVE_GC` is used as the functional equivalent.
```

```
.. c:member:: PyObject* PyTypeObject.tp_bases

   Tuple of base types.

   This is set for types created by a class statement.  It should be ``NULL`` for
   statically defined types.

   **Inheritance:**

   This field is not inherited.
```

```
.. c:member:: PyObject* PyTypeObject.tp_mro

   Tuple containing the expanded set of base types, starting with the type itself
   and ending with :class:`object`, in Method Resolution Order.


   **Inheritance:**

   This field is not inherited; it is calculated fresh by
   :c:func:`PyType_Ready`.
```

```
.. c:member:: PyObject* PyTypeObject.tp_cache

Unused.  Internal use only.

**Inheritance:**

This field is not inherited.
```

```
.. c:member:: PyObject* PyTypeObject.tp_subclasses

List of weak references to subclasses.  Internal use only.

**Inheritance:**

This field is not inherited.
```

```
.. c:member:: PyObject* PyTypeObject.tp_weaklist

Weak reference list head, for weak references to this type object.  Not
inherited.  Internal use only.

**Inheritance:**

This field is not inherited.
```

```
.. c:member:: destructor PyTypeObject.tp_del

This field is deprecated.  Use :c:member:`~PyTypeObject.tp_finalize` instead.
```

```
.. c:member:: unsigned int PyTypeObject.tp_version_tag

Used to index into the method cache.  Internal use only.

**Inheritance:**

This field is not inherited.
```

```
.. c:member:: destructor PyTypeObject.tp_finalize

An optional pointer to an instance finalization function.  Its signature is::

    void tp_finalize(PyObject *self);

If :c:member:`~PyTypeObject.tp_finalize` is set, the interpreter calls it once when
finalizing an instance.  It is called either from the garbage
collector (if the instance is part of an isolated reference cycle) or
just before the object is deallocated.  Either way, it is guaranteed
to be called before attempting to break reference cycles, ensuring
that it finds the object in a sane state.

:c:member:`~PyTypeObject.tp_finalize` should not mutate the current exception status;
therefore, a recommended way to write a non-trivial finalizer is::

    static void
    local_finalize(PyObject *self)
    {
        PyObject *error_type, *error_value, *error_traceback;

        /* Save the current exception, if any. */
        PyErr_Fetch(&error_type, &error_value, &error_traceback);

        /* ... */

        /* Restore the saved exception. */
        PyErr_Restore(error_type, error_value, error_traceback);
    }

For this field to be taken into account (even through inheritance),
you must also set the :const:`Py_TPFLAGS_HAVE_FINALIZE` flags bit.
```

```
    Also, note that, in a garbage collected Python,
    :c:member:`~PyTypeObject.tp_dealloc` may be called from
    any Python thread, not just the thread which created the object (if the object
    becomes part of a refcount cycle, that cycle might be collected by a garbage
    collection on any thread).  This is not a problem for Python API calls, since
    the thread on which tp_dealloc is called will own the Global Interpreter Lock
    (GIL). However, if the object being destroyed in turn destroys objects from some
    other C or C++ library, care should be taken to ensure that destroying those
    objects on the thread which called tp_dealloc will not violate any assumptions
    of the library.

    **Inheritance:**

    This field is inherited by subtypes.

    .. versionadded:: 3.4

    .. seealso:: "Safe object finalization" (:pep:`442`)
```

```
    .. c:member:: vectorcallfunc PyTypeObject.tp_vectorcall

    Vectorcall function to use for calls of this type object.
    In other words, it is used to implement
    :ref:`vectorcall <vectorcall>` for ``type.__call__``.
    If ``tp_vectorcall`` is ``NULL``, the default call implementation
    using :attr:`__new__` and :attr:`__init__` is used.

    **Inheritance:**

    This field is never inherited.

    .. versionadded:: 3.9 (the field exists since 3.8 but it's only used since 3.9)
```

Static Types

Traditionally, types defined in C code are *static*, that is, a static :c:type:`PyTypeObject` structure is defined directly in code and initialized using :c:func:`PyType_Ready`.

This results in types that are limited relative to types defined in Python:

- Static types are limited to one base, i.e. they cannot use multiple inheritance.
- Static type objects (but not necessarily their instances) are immutable. It is not possible to add or modify the type object's attributes from Python.
- Static type objects are shared across :ref:`sub-interpreters <sub-interpreter-support>`, so they should not include any subinterpreter-specific state.

Also, since :c:type:`PyTypeObject` is not part of the :ref:`stable ABI <stable>`, any extension modules using static types must be compiled for a specific Python minor version.

Heap Types

An alternative to :ref:`static types <static-types>` is *heap-allocated types*, or *heap types* for short, which correspond closely to classes created by Python's `class` statement. Heap types have the :const:`Py_TPFLAGS_HEAPTYPE` flag set.

This is done by filling a :c:type:`PyType_Spec` structure and calling :c:func:`PyType_FromSpec`, :c:func:`PyType_FromSpecWithBases`, or :c:func:`PyType_FromModuleAndSpec`.

Number Object Structures

Binary and ternary functions must check the type of all their operands,
and implement the necessary conversions (at least one of the operands is
an instance of the defined type). If the operation is not defined for the
given operands, binary and ternary functions must return
``Py_NotImplemented``, if another error occurred they must return ``NULL``
and set an exception.

.. note::

   The :c:data:`nb_reserved` field should always be ``NULL``. It
   was previously called :c:data:`nb_long`, and was renamed in
   Python 3.0.1.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2153)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_add
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2154)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_subtract
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2155)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_multiply
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2156)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_remainder
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2157)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_divmod
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2158)

Unknown directive type "c:member".

```
.. c:member:: ternaryfunc PyNumberMethods.nb_power
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2159)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_negative
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2160)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_positive
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2161)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_absolute
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2162)

Unknown directive type "c:member".

```
.. c:member:: inquiry PyNumberMethods.nb_bool
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2163)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_invert
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2164**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_lshift
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2165**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_rshift
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2166**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_and
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2167**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_xor
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2168**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_or
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2169**)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_int
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2170**)

Unknown directive type "c:member".

```
.. c:member:: void *PyNumberMethods.nb_reserved
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2171**)

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_float
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2172**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_add
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2173**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_subtract
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2174**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_multiply
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, **line 2175**)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_remainder
```

Unknown directive type "c:member".

```
.. c:member:: ternaryfunc PyNumberMethods.nb_inplace_power
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_lshift
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_rshift
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_and
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_xor
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_or
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_floor_divide
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_true_divide
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_floor_divide
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_true_divide
```

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyNumberMethods.nb_index
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_matrix_multiply
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyNumberMethods.nb_inplace_matrix_multiply
```

Mapping Object Structures

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Amaury Forgeot d'Arc
```

Unknown directive type "c:type".

```
.. c:type:: PyMappingMethods

   This structure holds pointers to the functions which an object uses to
   implement the mapping protocol.  It has three members:
```

Unknown directive type "c:member".

```
.. c:member:: lenfunc PyMappingMethods.mp_length

   This function is used by :c:func:`PyMapping_Size` and
   :c:func:`PyObject_Size`, and has the same signature.  This slot may be set to
   ``NULL`` if the object has no defined length.
```

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PyMappingMethods.mp_subscript

   This function is used by :c:func:`PyObject_GetItem` and
   :c:func:`PySequence_GetSlice`, and has the same signature as
   :c:func:`!PyObject_GetItem`.  This slot must be filled for the
   :c:func:`PyMapping_Check` function to return ``1``, it can be ``NULL``
   otherwise.
```

Unknown directive type "c:member".

```
.. c:member:: objobjargproc PyMappingMethods.mp_ass_subscript

   This function is used by :c:func:`PyObject_SetItem`,
   :c:func:`PyObject_DelItem`, :c:func:`PyObject_SetSlice` and
   :c:func:`PyObject_DelSlice`.  It has the same signature as
   :c:func:`!PyObject_SetItem`, but *v* can also be set to ``NULL`` to delete
   an item.  If this slot is ``NULL``, the object does not support item
   assignment and deletion.
```

Sequence Object Structures

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Amaury Forgeot d'Arc
```

Unknown directive type "c:type".

```
.. c:type:: PySequenceMethods

   This structure holds pointers to the functions which an object uses to
   implement the sequence protocol.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2241)

Unknown directive type "c:member".

```
.. c:member:: lenfunc PySequenceMethods.sq_length

   This function is used by :c:func:`PySequence_Size` and
   :c:func:`PyObject_Size`, and has the same signature.  It is also used for
   handling negative indices via the :c:member:`~PySequenceMethods.sq_item`
   and the :c:member:`~PySequenceMethods.sq_ass_item` slots.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2248)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PySequenceMethods.sq_concat

   This function is used by :c:func:`PySequence_Concat` and has the same
   signature.  It is also used by the ``+`` operator, after trying the numeric
   addition via the :c:member:`~PyNumberMethods.nb_add` slot.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2254)

Unknown directive type "c:member".

```
.. c:member:: ssizeargfunc PySequenceMethods.sq_repeat

   This function is used by :c:func:`PySequence_Repeat` and has the same
   signature.  It is also used by the ``*`` operator, after trying numeric
   multiplication via the :c:member:`~PyNumberMethods.nb_multiply` slot.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2260)

Unknown directive type "c:member".

```
.. c:member:: ssizeargfunc PySequenceMethods.sq_item

   This function is used by :c:func:`PySequence_GetItem` and has the same
   signature.  It is also used by :c:func:`PyObject_GetItem`, after trying
   the subscription via the :c:member:`~PyMappingMethods.mp_subscript` slot.
   This slot must be filled for the :c:func:`PySequence_Check`
   function to return ``1``, it can be ``NULL`` otherwise.

   Negative indexes are handled as follows: if the :attr:`sq_length` slot is
   filled, it is called and the sequence length is used to compute a positive
   index which is passed to :attr:`sq_item`.  If :attr:`sq_length` is ``NULL``,
   the index is passed as is to the function.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2273)

Unknown directive type "c:member".

```
.. c:member:: ssizeobjargproc PySequenceMethods.sq_ass_item

   This function is used by :c:func:`PySequence_SetItem` and has the same
   signature.  It is also used by :c:func:`PyObject_SetItem` and
   :c:func:`PyObject_DelItem`, after trying the item assignment and deletion
   via the :c:member:`~PyMappingMethods.mp_ass_subscript` slot.
   This slot may be left to ``NULL`` if the object does not support
   item assignment and deletion.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2282)

Unknown directive type "c:member".

```
.. c:member:: objobjproc PySequenceMethods.sq_contains

   This function may be used by :c:func:`PySequence_Contains` and has the same
   signature.  This slot may be left to ``NULL``, in this case
   :c:func:`!PySequence_Contains` simply traverses the sequence until it
   finds a match.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2289)

Unknown directive type "c:member".

```
.. c:member:: binaryfunc PySequenceMethods.sq_inplace_concat

   This function is used by :c:func:`PySequence_InPlaceConcat` and has the same
   signature.  It should modify its first operand, and return it.  This slot
   may be left to ``NULL``, in this case :c:func:`!PySequence_InPlaceConcat`
   will fall back to :c:func:`PySequence_Concat`.  It is also used by the
   augmented assignment ``+=``, after trying numeric in-place addition
   via the :c:member:`~PyNumberMethods.nb_inplace_add` slot.
```

Buffer Object Structures

:c:func:`PyBuffer_FillInfo` provides an easy way of exposing a simple
bytes buffer while dealing correctly with all request types.

:c:func:`PyObject_GetBuffer` is the interface for the consumer that
wraps this function.

```
.. c:member:: releasebufferproc PyBufferProcs.bf_releasebuffer

   The signature of this function is::

       void (PyObject *exporter, Py_buffer *view);

   Handle a request to release the resources of the buffer. If no resources
   need to be released, :c:member:`PyBufferProcs.bf_releasebuffer` may be
   ``NULL``. Otherwise, a standard implementation of this function will take
   these optional steps:

   (1) Decrement an internal counter for the number of exports.

   (2) If the counter is ``0``, free all memory associated with *view*.

   The exporter MUST use the :c:member:`~Py_buffer.internal` field to keep
   track of buffer-specific resources. This field is guaranteed to remain
   constant, while a consumer MAY pass a copy of the original buffer as the
   *view* argument.


   This function MUST NOT decrement :c:data:`view->obj`, since that is
   done automatically in :c:func:`PyBuffer_Release` (this scheme is
   useful for breaking reference cycles).


   :c:func:`PyBuffer_Release` is the interface for the consumer that
   wraps this function.
```

Async Object Structures

```
.. sectionauthor:: Yury Selivanov <yselivanov@sprymix.com>
```

```
.. versionadded:: 3.5
```

```
.. c:type:: PyAsyncMethods

   This structure holds pointers to the functions required to implement
   :term:`awaitable` and :term:`asynchronous iterator` objects.

   Here is the structure definition::

       typedef struct {
           unaryfunc am_await;
           unaryfunc am_aiter;
           unaryfunc am_anext;
           sendfunc am_send;
       } PyAsyncMethods;
```

```
.. c:member:: unaryfunc PyAsyncMethods.am_await

   The signature of this function is::

       PyObject *am_await(PyObject *self);

   The returned object must be an :term:`iterator`, i.e. :c:func:`PyIter_Check`
   must return ``1`` for it.

   This slot may be set to ``NULL`` if an object is not an :term:`awaitable`.
```

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyAsyncMethods.am_aiter

   The signature of this function is::

       PyObject *am_aiter(PyObject *self);

   Must return an :term:`asynchronous iterator` object.
   See :meth:`__anext__` for details.

   This slot may be set to ``NULL`` if an object does not implement
   asynchronous iteration protocol.
```

Unknown directive type "c:member".

```
.. c:member:: unaryfunc PyAsyncMethods.am_anext

   The signature of this function is::

       PyObject *am_anext(PyObject *self);

   Must return an :term:`awaitable` object.  See :meth:`__anext__` for details.
   This slot may be set to ``NULL``.
```

Unknown directive type "c:member".

```
.. c:member:: sendfunc PyAsyncMethods.am_send

   The signature of this function is::

       PySendResult am_send(PyObject *self, PyObject *arg, PyObject **result);

   See :c:func:`PyIter_Send` for details.
   This slot may be set to ``NULL``.

   .. versionadded:: 3.10
```

Slot Type typedefs

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*allocfunc)(PyTypeObject *cls, Py_ssize_t nitems)

   The purpose of this function is to separate memory allocation from memory
   initialization.  It should return a pointer to a block of memory of adequate
   length for the instance, suitably aligned, and initialized to zeros, but with
   :attr:`ob_refcnt` set to ``1`` and :attr:`ob_type` set to the type argument.  If
   the type's :c:member:`~PyTypeObject.tp_itemsize` is non-zero, the object's :attr:`ob_size` field
   should be initialized to *nitems* and the length of the allocated memory block
   should be ``tp_basicsize + nitems*tp_itemsize``, rounded up to a multiple of
   ``sizeof(void*)``; otherwise, *nitems* is not used and the length of the block
   should be :c:member:`~PyTypeObject.tp_basicsize`.

   This function should not do any other instance initialization, not even to
   allocate additional memory; that should be done by :c:member:`~PyTypeObject.tp_new`.
```

Unknown directive type "c:type".

```
.. c:type:: void (*destructor)(PyObject *)
```

Unknown directive type "c:type".

```
.. c:type:: void (*freefunc)(void *)

   See :c:member:`~PyTypeObject.tp_free`.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*newfunc)(PyObject *, PyObject *, PyObject *)

   See :c:member:`~PyTypeObject.tp_new`.
```

Unknown directive type "c:type".

```
.. c:type:: int (*initproc)(PyObject *, PyObject *, PyObject *)

    See :c:member:`~PyTypeObject.tp_init`.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*reprfunc)(PyObject *)

    See :c:member:`~PyTypeObject.tp_repr`.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*getattrfunc)(PyObject *self, char *attr)

    Return the value of the named attribute for the object.
```

Unknown directive type "c:type".

```
.. c:type:: int (*setattrfunc)(PyObject *self, char *attr, PyObject *value)

    Set the value of the named attribute for the object.
    The value argument is set to ``NULL`` to delete the attribute.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*getattrofunc)(PyObject *self, PyObject *attr)

    Return the value of the named attribute for the object.

    See :c:member:`~PyTypeObject.tp_getattro`.
```

Unknown directive type "c:type".

```
.. c:type:: int (*setattrofunc)(PyObject *self, PyObject *attr, PyObject *value)

    Set the value of the named attribute for the object.
    The value argument is set to ``NULL`` to delete the attribute.

    See :c:member:`~PyTypeObject.tp_setattro`.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*descrgetfunc)(PyObject *, PyObject *, PyObject *)

    See :c:member:`~PyTypeObject.tp_descrget`.
```

Unknown directive type "c:type".

```
.. c:type:: int (*descrsetfunc)(PyObject *, PyObject *, PyObject *)

    See :c:member:`~PyTypeObject.tp_descrset`.
```

Unknown directive type "c:type".

```
.. c:type:: Py_hash_t (*hashfunc)(PyObject *)

    See :c:member:`~PyTypeObject.tp_hash`.
```

Unknown directive type "c:type".

```
.. c:type:: PyObject *(*richcmpfunc)(PyObject *, PyObject *, int)

    See :c:member:`~PyTypeObject.tp_richcompare`.
```

```
.. c:type:: PyObject *(*getiterfunc)(PyObject *)

    See :c:member:`~PyTypeObject.tp_iter`.
```

```
.. c:type:: PyObject *(*iternextfunc)(PyObject *)

    See :c:member:`~PyTypeObject.tp_iternext`.
```

```
.. c:type:: Py_ssize_t (*lenfunc)(PyObject *)
```

```
.. c:type:: int (*getbufferproc)(PyObject *, Py_buffer *, int)
```

```
.. c:type:: void (*releasebufferproc)(PyObject *, Py_buffer *)
```

```
.. c:type:: PyObject *(*unaryfunc)(PyObject *)
```

```
.. c:type:: PyObject *(*binaryfunc)(PyObject *, PyObject *)
```

```
.. c:type:: PySendResult (*sendfunc)(PyObject *, PyObject *, PyObject **)

    See :c:member:`~PyAsyncMethods.am_send`.
```

```
.. c:type:: PyObject *(*ternaryfunc)(PyObject *, PyObject *, PyObject *)
```

```
.. c:type:: PyObject *(*ssizeargfunc)(PyObject *, Py_ssize_t)
```

```
.. c:type:: int (*ssizeobjargproc)(PyObject *, Py_ssize_t)
```

Examples

The following are simple examples of Python type definitions. They include common usage you may encounter. Some demonstrate tricky corner cases. For more examples, practical info, and a tutorial, see :ref:`defining-new-types` and :ref:`new-types-topics`.

A basic :ref:`static type <static-types>`:

```
typedef struct {
    PyObject_HEAD
    const char *data;
} MyObject;

static PyTypeObject MyObject_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    .tp_name = "mymod.MyObject",
    .tp_basicsize = sizeof(MyObject),
    .tp_doc = "My objects",
    .tp_new = myobj_new,
    .tp_dealloc = (destructor)myobj_dealloc,
    .tp_repr = (reprfunc)myobj_repr,
};
```

You may also find older code (especially in the CPython code base) with a more verbose initializer:

```
static PyTypeObject MyObject_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    "mymod.MyObject",            /* tp_name */
    sizeof(MyObject),            /* tp_basicsize */
    0,                           /* tp_itemsize */
    (destructor)myobj_dealloc,   /* tp_dealloc */
    0,                           /* tp_vectorcall_offset */
    0,                           /* tp_getattr */
    0,                           /* tp_setattr */
    0,                           /* tp_as_async */
    (reprfunc)myobj_repr,        /* tp_repr */
    0,                           /* tp_as_number */
    0,                           /* tp_as_sequence */
    0,                           /* tp_as_mapping */
    0,                           /* tp_hash */
    0,                           /* tp_call */
    0,                           /* tp_str */
    0,                           /* tp_getattro */
    0,                           /* tp_setattro */
    0,                           /* tp_as_buffer */
    0,                           /* tp_flags */
    "My objects",                /* tp_doc */
    0,                           /* tp_traverse */
    0,                           /* tp_clear */
    0,                           /* tp_richcompare */
    0,                           /* tp_weaklistoffset */
    0,                           /* tp_iter */
    0,                           /* tp_iternext */
    0,                           /* tp_methods */
    0,                           /* tp_members */
    0,                           /* tp_getset */
    0,                           /* tp_base */
    0,                           /* tp_dict */
    0,                           /* tp_descr_get */
    0,                           /* tp_descr_set */
    0,                           /* tp_dictoffset */
    0,                           /* tp_init */
    0,                           /* tp_alloc */
    myobj_new,                   /* tp_new */
};
```

A type that supports weakrefs, instance dicts, and hashing:

```
typedef struct {
```

```
    PyObject_HEAD
    const char *data;
    PyObject *inst_dict;
    PyObject *weakreflist;
} MyObject;

static PyTypeObject MyObject_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    .tp_name = "mymod.MyObject",
    .tp_basicsize = sizeof(MyObject),
    .tp_doc = "My objects",
    .tp_weaklistoffset = offsetof(MyObject, weakreflist),
    .tp_dictoffset = offsetof(MyObject, inst_dict),
    .tp_flags = Py_TPFLAGS_DEFAULT | Py_TPFLAGS_BASETYPE | Py_TPFLAGS_HAVE_GC,
    .tp_new = myobj_new,
    .tp_traverse = (traverseproc)myobj_traverse,
    .tp_clear = (inquiry)myobj_clear,
    .tp_alloc = PyType_GenericNew,
    .tp_dealloc = (destructor)myobj_dealloc,
    .tp_repr = (reprfunc)myobj_repr,
    .tp_hash = (hashfunc)myobj_hash,
    .tp_richcompare = PyBaseObject_Type.tp_richcompare,
};
```

A str subclass that cannot be subclassed and cannot be called to create instances (e.g. uses a separate factory func) using :c:data:`Py_TPFLAGS_DISALLOW_INSTANTIATION` flag:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2677); *backlink***
>
> Unknown interpreted text role "c:data".

```
typedef struct {
    PyUnicodeObject raw;
    char *extra;
} MyStr;

static PyTypeObject MyStr_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    .tp_name = "mymod.MyStr",
    .tp_basicsize = sizeof(MyStr),
    .tp_base = NULL,  // set to &PyUnicode_Type in module init
    .tp_doc = "my custom str",
    .tp_flags = Py_TPFLAGS_DEFAULT | Py_TPFLAGS_DISALLOW_INSTANTIATION,
    .tp_repr = (reprfunc)myobj_repr,
};
```

The simplest :ref:`static type <static-types>` with fixed-length instances:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2696); *backlink***
>
> Unknown interpreted text role "ref".

```
typedef struct {
    PyObject_HEAD
} MyObject;

static PyTypeObject MyObject_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    .tp_name = "mymod.MyObject",
};
```

The simplest :ref:`static type <static-types>` with variable-length instances:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\(cpython-main)(Doc)(c-api)typeobj.rst`, line 2707); *backlink***
>
> Unknown interpreted text role "ref".

```
typedef struct {
    PyObject_VAR_HEAD
    const char *data[1];
} MyObject;

static PyTypeObject MyObject_Type = {
    PyVarObject_HEAD_INIT(NULL, 0)
    .tp_name = "mymod.MyObject",
    .tp_basicsize = sizeof(MyObject) - sizeof(char *),
    .tp_itemsize = sizeof(char *),
};
```