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

Unknown directive type "highlight".

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

Slice Objects

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

Unknown directive type "c:var".

```
.. c:var:: PyTypeObject PySlice_Type

   The type object for slice objects.  This is the same as :class:`slice` in the
   Python layer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]slice.rst, line 15)

Unknown directive type "c:function".

```
.. c:function:: int PySlice_Check(PyObject *ob)

   Return true if *ob* is a slice object; *ob* must not be ``NULL``.  This
   function always succeeds.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]slice.rst, line 21)

Unknown directive type "c:function".

```
.. c:function:: PyObject* PySlice_New(PyObject *start, PyObject *stop, PyObject *step)

   Return a new slice object with the given values.  The *start*, *stop*, and
   *step* parameters are used as the values of the slice object attributes of
   the same names.  Any of the values may be ``NULL``, in which case the
   ``None`` will be used for the corresponding attribute.  Return ``NULL`` if
   the new object could not be allocated.
```

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

Unknown directive type "c:function".

```
.. c:function:: int PySlice_GetIndices(PyObject *slice, Py_ssize_t length, Py_ssize_t *start, Py_ssize_t *stop, Py_ssize_t *ste

   Retrieve the start, stop and step indices from the slice object *slice*,
   assuming a sequence of length *length*. Treats indices greater than
   *length* as errors.

   Returns ``0`` on success and ``-1`` on error with no exception set (unless one of
   the indices was not :const:`None` and failed to be converted to an integer,
   in which case ``-1`` is returned with an exception set).

   You probably do not want to use this function.

   .. versionchanged:: 3.2
      The parameter type for the *slice* parameter was ``PySliceObject*``
      before.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]slice.rst, line 47)

Unknown directive type "c:function".

```
.. c:function:: int PySlice_GetIndicesEx(PyObject *slice, Py_ssize_t length, Py_ssize_t *start, Py_ssize_t *stop, Py_ssize_t *s

   Usable replacement for :c:func:`PySlice_GetIndices`.  Retrieve the start,
   stop, and step indices from the slice object *slice* assuming a sequence of
   length *length*, and store the length of the slice in *slicelength*.  Out
   of bounds indices are clipped in a manner consistent with the handling of
   normal slices.

   Returns ``0`` on success and ``-1`` on error with exception set.

   .. note::
      This function is considered not safe for resizable sequences.
      Its invocation should be replaced by a combination of
      :c:func:`PySlice_Unpack` and :c:func:`PySlice_AdjustIndices` where ::

         if (PySlice_GetIndicesEx(slice, length, &start, &stop, &step, &slicelength) < 0) {
             // return error
         }

      is replaced by ::

         if (PySlice_Unpack(slice, &start, &stop, &step) < 0) {
             // return error
         }
         slicelength = PySlice_AdjustIndices(length, &start, &stop, step);

   .. versionchanged:: 3.2
      The parameter type for the *slice* parameter was ``PySliceObject*``
      before.

   .. versionchanged:: 3.6.1
      If ``Py_LIMITED_API`` is not set or set to the value between ``0x03050400``
      and ``0x03060000`` (not including) or ``0x03060100`` or higher
      :c:func:`!PySlice_GetIndicesEx` is implemented as a macro using
      :c:func:`!PySlice_Unpack` and :c:func:`!PySlice_AdjustIndices`.
      Arguments *start*, *stop* and *step* are evaluated more than once.

   .. deprecated:: 3.6.1
      If ``Py_LIMITED_API`` is set to the value less than ``0x03050400`` or
      between ``0x03060000`` and ``0x03060100`` (not including)
```

```
   :c:func:`!PySlice_GetIndicesEx` is a deprecated function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]slice.rst, line 90)

Unknown directive type "c:function".

```
.. c:function:: int PySlice_Unpack(PyObject *slice, Py_ssize_t *start, Py_ssize_t *stop, Py_ssize_t *step)

   Extract the start, stop and step data members from a slice object as
   C integers.  Silently reduce values larger than ``PY_SSIZE_T_MAX`` to
   ``PY_SSIZE_T_MAX``, silently boost the start and stop values less than
   ``PY_SSIZE_T_MIN`` to ``PY_SSIZE_T_MIN``, and silently boost the step
   values less than ``-PY_SSIZE_T_MAX`` to ``-PY_SSIZE_T_MAX``.

   Return ``-1`` on error, ``0`` on success.

   .. versionadded:: 3.6.1
```

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

Unknown directive type "c:function".

```
.. c:function:: Py_ssize_t PySlice_AdjustIndices(Py_ssize_t length, Py_ssize_t *start, Py_ssize_t *stop, Py_ssize_t step)

   Adjust start/end slice indices assuming a sequence of the specified length.
   Out of bounds indices are clipped in a manner consistent with the handling
   of normal slices.

   Return the length of the slice.  Always successful.  Doesn't call Python
   code.

   .. versionadded:: 3.6.1
```

Ellipsis Object

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\[cpython-main][Doc][c-api]slice.rst, line 119)

Unknown directive type "c:var".

```
.. c:var:: PyObject *Py_Ellipsis

   The Python ``Ellipsis`` object.  This object has no methods.  It needs to be
   treated just like any other object with respect to reference counts.  Like
   :c:data:`Py_None` it is a singleton object.
```