Function Objects

There are a few functions specific to Python functions.

```
.. c:function:: PyObject* PyFunction_NewWithQualName(PyObject *code, PyObject *globals, PyObject *qual
```

As :c:func:`PyFunction_New`, but also allows setting the function object's
``__qualname__`` attribute. *qualname* should be a unicode object or ``NULL``;
if ``NULL``, the ``__qualname__`` attribute is set to the same value as the
code object's ``co_qualname`` field.

.. versionadded:: 3.3

```
.. c:function:: PyObject* PyFunction_GetCode(PyObject *op)
```

Return the code object associated with the function object *op*.

```
.. c:function:: PyObject* PyFunction_GetGlobals(PyObject *op)
```

Return the globals dictionary associated with the function object *op*.

```
.. c:function:: PyObject* PyFunction_GetModule(PyObject *op)
```

Return a :term:`borrowed reference` to the *__module__* attribute of the
function object *op*. It can be *NULL*.

This is normally a string containing the module name, but can be set to any
other object by Python code.

```
.. c:function:: PyObject* PyFunction_GetDefaults(PyObject *op)
```

Return the argument default values of the function object *op*. This can be a
tuple of arguments or ``NULL``.

```
.. c:function:: int PyFunction_SetDefaults(PyObject *op, PyObject *defaults)
```

Set the argument default values for the function object *op*. *defaults* must be
``Py_None`` or a tuple.

Raises :exc:`SystemError` and returns ``-1`` on failure.

```
.. c:function:: PyObject* PyFunction_GetClosure(PyObject *op)
```

Return the closure associated with the function object *op*. This can be ``NULL``
or a tuple of cell objects.

```
.. c:function:: int PyFunction_SetClosure(PyObject *op, PyObject *closure)

   Set the closure associated with the function object *op*. *closure* must be
   ``Py_None`` or a tuple of cell objects.

   Raises :exc:`SystemError` and returns ``-1`` on failure.
```

```
.. c:function:: PyObject *PyFunction_GetAnnotations(PyObject *op)

   Return the annotations of the function object *op*. This can be a
   mutable dictionary or ``NULL``.
```

```
.. c:function:: int PyFunction_SetAnnotations(PyObject *op, PyObject *annotations)

   Set the annotations for the function object *op*. *annotations*
   must be a dictionary or ``Py_None``.

   Raises :exc:`SystemError` and returns ``-1`` on failure.
```