:mod:`copy` --- Shallow and deep copy operations

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 4)

Unknown directive type "module".

```
.. module:: copy
   :synopsis: Shallow and deep copy operations.
```

Source code: :source:`Lib/copy.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 7); *backlink*

Unknown interpreted text role "source".

Assignment statements in Python do not copy objects, they create bindings between a target and an object. For collections that are mutable or contain mutable items, a copy is sometimes needed so one can change one copy without changing the other. This module provides generic shallow and deep copy operations (explained below).

Interface summary:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 20)

Unknown directive type "function".

```
.. function:: copy(x)

   Return a shallow copy of *x*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 25)

Unknown directive type "function".

```
.. function:: deepcopy(x[, memo])

   Return a deep copy of *x*.
```

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

Unknown directive type "exception".

```
.. exception:: Error

   Raised for module specific errors.
```

The difference between shallow and deep copying is only relevant for compound objects (objects that contain other objects, like lists or class instances):

- A *shallow copy* constructs a new compound object and then (to the extent possible) inserts *references* into it to the objects found in the original.
- A *deep copy* constructs a new compound object and then, recursively, inserts *copies* into it of the objects found in the original.

Two problems often exist with deep copy operations that don't exist with shallow copy operations:

- Recursive objects (compound objects that, directly or indirectly, contain a reference to themselves) may cause a recursive loop.
- Because deep copy copies everything it may copy too much, such as data which is intended to be shared between copies.

The :func:`deepcopy` function avoids these problems by:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 54); *backlink*

Unknown interpreted text role "func".

- keeping a `memo` dictionary of objects already copied during the current copying pass; and
- letting user-defined classes override the copying operation or the set of components copied.

This module does not copy types like module, method, stack trace, stack frame, file, socket, window, or any similar types. It does "copy" functions and classes (shallow and deeply), by returning the original object unchanged; this is compatible with the way these are treated by the :mod:`pickle` module.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 62); *backlink*

Unknown interpreted text role "mod".

Shallow copies of dictionaries can be made using :meth:`dict.copy`, and of lists by assigning a slice of the entire list, for example, `copied_list = original_list[:]`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 67); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 71)

Unknown directive type "index".

```
.. index:: module: pickle
```

Classes can use the same interfaces to control copying that they use to control pickling. See the description of module :mod:`pickle` for information on these methods. In fact, the :mod:`copy` module uses the registered pickle functions from the :mod:`copyreg` module.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 73); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 73); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 73); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 78)

Unknown directive type "index".

```
.. index::
   single: __copy__() (copy protocol)
   single: __deepcopy__() (copy protocol)
```

In order for a class to define its own copy implementation, it can define special methods :meth:`__copy__` and :meth:`__deepcopy__`. The former is called to implement the shallow copy operation; no additional arguments are passed. The latter is called to implement the deep copy operation; it is passed one argument, the `memo` dictionary. If the :meth:`__deepcopy__` implementation needs to make a deep copy of a component, it should call the :func:`deepcopy` function with the component as first argument and the memo dictionary as second argument. The memo dictionary should be treated as an opaque object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 82); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 82); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 82); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 82); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]copy.rst, line 92)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`pickle`
      Discussion of the special methods used to support object state retrieval and
      restoration.
```