8.11. `pprint` — Data pretty printer

Source code: Lib/pprint.py

The `pprint` module provides a capability to "pretty-print" arbitrary Python data structures in a form which can be used as input to the interpreter. If the formatted structures include objects which are not fundamental Python types, the representation may not be loadable. This may be the case if objects such as files, sockets or classes are included, as well as many other objects which are not representable as Python literals.

The formatted representation keeps objects on a single line if it can, and breaks them onto multiple lines if they don't fit within the allowed width. Construct `PrettyPrinter` objects explicitly if you need to adjust the width constraint.

Dictionaries are sorted by key before the display is computed.

The `pprint` module defines one class:

class `pprint.`**`PrettyPrinter`**(*indent=1*, *width=80*, *depth=None*, *stream=None*, **, compact=False*)

Construct a `PrettyPrinter` instance. This constructor understands several keyword parameters. An output stream may be set using the *stream* keyword; the only method used on the stream object is the file protocol's `write()` method. If not specified, the `PrettyPrinter` adopts `sys.stdout`. The amount of indentation added for each recursive level is specified by *indent*; the default is one. Other values can cause output to look a little odd, but can make nesting easier to spot. The number of levels which may be printed is controlled by *depth*; if the data structure being printed is too deep, the next contained level is replaced by `...`. By default, there is no constraint on the depth of the objects being formatted. The desired output width is constrained using the *width* parameter; the default is 80 characters. If a structure cannot be formatted within the constrained width, a best effort will be made. If *compact* is false (the default) each item of a long sequence will be formatted on a separate line. If *compact* is true, as many items as will fit within the *width* will be formatted on each output line.

Changed in version 3.4: Added the *compact* parameter.

```
>>> import pprint
>>> stuff = ['spam', 'eggs', 'lumberjack', 'knights', 'ni']
>>> stuff.insert(0, stuff[:])
>>> pp = pprint.PrettyPrinter(indent=4)
```

```
>>> pp.pprint(stuff)
[   ['spam', 'eggs', 'lumberjack', 'knights', 'ni'],
    'spam',
    'eggs',
    'lumberjack',
    'knights',
    'ni']
>>> pp = pprint.PrettyPrinter(width=41, compact=True)
>>> pp.pprint(stuff)
[['spam', 'eggs', 'lumberjack',
  'knights', 'ni'],
 'spam', 'eggs', 'lumberjack', 'knights',
 'ni']
>>> tup = ('spam', ('eggs', ('lumberjack', ('knights', ('ni', ('de
... ('parrot', ('fresh fruit',)))))))
>>> pp = pprint.PrettyPrinter(depth=6)
>>> pp.pprint(tup)
('spam', ('eggs', ('lumberjack', ('knights', ('ni', ('dead', (...)
```

The `pprint` module also provides several shortcut functions:

`pprint.`**`pformat`**(*object*, *indent=1*, *width=80*, *depth=None*, *, *compact=False*)

Return the formatted representation of *object* as a string. *indent*, *width*, *depth* and *compact* will be passed to the `PrettyPrinter` constructor as formatting parameters.

Changed in version 3.4: Added the *compact* parameter.

`pprint.`**`pprint`**(*object*, *stream=None*, *indent=1*, *width=80*, *depth=None*, *, *compact=False*)

Prints the formatted representation of *object* on *stream*, followed by a newline. If *stream* is `None`, `sys.stdout` is used. This may be used in the interactive interpreter instead of the `print()` function for inspecting values (you can even reassign `print = pprint.pprint` for use within a scope). *indent*, *width*, *depth* and *compact* will be passed to the `PrettyPrinter` constructor as formatting parameters.

Changed in version 3.4: Added the *compact* parameter.

```
>>> import pprint
>>> stuff = ['spam', 'eggs', 'lumberjack', 'knights', 'ni']
>>> stuff.insert(0, stuff)
>>> pprint.pprint(stuff)
[<Recursion on list with id=...>,
 'spam',
 'eggs',
 'lumberjack',
```

```
 'knights',
 'ni']
```

`pprint.`**`isreadable`**(*object*)

Determine if the formatted representation of *object* is “readable,” or can be used to reconstruct the value using `eval()`. This always returns `False` for recursive objects.

```
>>> pprint.isreadable(stuff)
False
```

`pprint.`**`isrecursive`**(*object*)

Determine if *object* requires a recursive representation.

One more support function is also defined:

`pprint.`**`saferepr`**(*object*)

Return a string representation of *object*, protected against recursive data structures. If the representation of *object* exposes a recursive entry, the recursive reference will be represented as `<Recursion on typename with id=number>`. The representation is not otherwise formatted.

```
>>> pprint.saferepr(stuff)
"[<Recursion on list with id=...>, 'spam', 'eggs', 'lumberjack', '
```

8.11.1. PrettyPrinter Objects

`PrettyPrinter` instances have the following methods:

`PrettyPrinter.`**`pformat`**(*object*)

Return the formatted representation of *object*. This takes into account the options passed to the `PrettyPrinter` constructor.

`PrettyPrinter.`**`pprint`**(*object*)

Print the formatted representation of *object* on the configured stream, followed by a newline.

The following methods provide the implementations for the corresponding functions of the same names. Using these methods on an instance is slightly more efficient since new `PrettyPrinter` objects don’t need to be created.

`PrettyPrinter.`**`isreadable`**(*object*)

Determine if the formatted representation of the object is “readable,” or can be used to reconstruct the value using `eval()`. Note that this returns `False` for recursive objects. If the *depth* parameter of the `PrettyPrinter` is set and the object is deeper than allowed, this returns `False`.

`PrettyPrinter.`**`isrecursive`**(*object*)

Determine if the object requires a recursive representation.

This method is provided as a hook to allow subclasses to modify the way objects are converted to strings. The default implementation uses the internals of the `saferepr()` implementation.

`PrettyPrinter.`**`format`**(*object*, *context*, *maxlevels*, *level*)

Returns three values: the formatted version of *object* as a string, a flag indicating whether the result is readable, and a flag indicating whether recursion was detected. The first argument is the object to be presented. The second is a dictionary which contains the `id()` of objects that are part of the current presentation context (direct and indirect containers for *object* that are affecting the presentation) as the keys; if an object needs to be presented which is already represented in *context*, the third return value should be `True`. Recursive calls to the `format()` method should add additional entries for containers to this dictionary. The third argument, *maxlevels*, gives the requested limit to recursion; this will be `0` if there is no requested limit. This argument should be passed unmodified to recursive calls. The fourth argument, *level*, gives the current level; recursive calls should be passed a value less than that of the current call.

8.11.2. Example

To demonstrate several uses of the `pprint()` function and its parameters, let’s fetch information about a project from PyPI:

```
>>> import json
>>> import pprint
>>> from urllib.request import urlopen
>>> with urlopen('http://pypi.org/project/Twisted/json') as url:
...     http_info = url.info()
...     raw_data = url.read().decode(http_info.get_content_charset())
>>> project_info = json.loads(raw_data)
```

In its basic form, `pprint()` shows the whole object:

```
>>> pprint.pprint(project_info)
{'info': {'_pypi_hidden': False,
          '_pypi_ordering': 125,
          'author': 'Glyph Lefkowitz',
```

```
          'author_email': 'glyph@twistedmatrix.com',
          'bugtrack_url': '',
          'cheesecake_code_kwalitee_id': None,
          'cheesecake_documentation_id': None,
          'cheesecake_installability_id': None,
          'classifiers': ['Programming Language :: Python :: 2.6',
                          'Programming Language :: Python :: 2.7',
                          'Programming Language :: Python :: 2 :: Only
          'description': 'An extensible framework for Python programmi
                         'special focus\r\n'
                         'on event-based network programming and multi
                         'integration.',
          'docs_url': '',
          'download_url': 'UNKNOWN',
          'home_page': 'http://twistedmatrix.com/',
          'keywords': '',
          'license': 'MIT',
          'maintainer': '',
          'maintainer_email': '',
          'name': 'Twisted',
          'package_url': 'http://pypi.org/project/Twisted',
          'platform': 'UNKNOWN',
          'release_url': 'http://pypi.org/project/Twisted/12.3.0',
          'requires_python': None,
          'stable_version': None,
          'summary': 'An asynchronous networking framework written in
          'version': '12.3.0'},
 'urls': [{'comment_text': '',
           'downloads': 71844,
           'filename': 'Twisted-12.3.0.tar.bz2',
           'has_sig': False,
           'md5_digest': '6e289825f3bf5591cfd670874cc0862d',
           'packagetype': 'sdist',
           'python_version': 'source',
           'size': 2615733,
           'upload_time': '2012-12-26T12:47:03',
           'url': 'https://pypi.org/packages/source/T/Twisted/Twisted-
          {'comment_text': '',
           'downloads': 5224,
           'filename': 'Twisted-12.3.0.win32-py2.7.msi',
           'has_sig': False,
           'md5_digest': '6b778f5201b622a5519a2aca1a2fe512',
           'packagetype': 'bdist_msi',
           'python_version': '2.7',
           'size': 2916352,
           'upload_time': '2012-12-26T12:48:15',
           'url': 'https://pypi.org/packages/2.7/T/Twisted/Twisted-12.
```

The result can be limited to a certain *depth* (ellipsis is used for deeper contents):

```
>>> pprint.pprint(project_info, depth=2)
{'info': {'_pypi_hidden': False,
          '_pypi_ordering': 125,
          'author': 'Glyph Lefkowitz',
          'author_email': 'glyph@twistedmatrix.com',
          'bugtrack_url': '',
          'cheesecake_code_kwalitee_id': None,
          'cheesecake_documentation_id': None,
          'cheesecake_installability_id': None,
          'classifiers': [...],
          'description': 'An extensible framework for Python programmi
                         'special focus\r\n'
                         'on event-based network programming and multi
                         'integration.',
          'docs_url': '',
          'download_url': 'UNKNOWN',
          'home_page': 'http://twistedmatrix.com/',
          'keywords': '',
          'license': 'MIT',
          'maintainer': '',
          'maintainer_email': '',
          'name': 'Twisted',
          'package_url': 'http://pypi.org/project/Twisted',
          'platform': 'UNKNOWN',
          'release_url': 'http://pypi.org/project/Twisted/12.3.0',
          'requires_python': None,
          'stable_version': None,
          'summary': 'An asynchronous networking framework written in
          'version': '12.3.0'},
 'urls': [{...}, {...}]}
```

Additionally, maximum character *width* can be suggested. If a long object cannot be split, the specified width will be exceeded:

```
>>> pprint.pprint(project_info, depth=2, width=50)
{'info': {'_pypi_hidden': False,
          '_pypi_ordering': 125,
          'author': 'Glyph Lefkowitz',
          'author_email': 'glyph@twistedmatrix.com',
          'bugtrack_url': '',
          'cheesecake_code_kwalitee_id': None,
          'cheesecake_documentation_id': None,
          'cheesecake_installability_id': None,
          'classifiers': [...],
          'description': 'An extensible '
                         'framework for Python '
                         'programming, with '
                         'special focus\r\n'
                         'on event-based network '
                         'programming and '
```

```
                          'multiprotocol '
                          'integration.',
          'docs_url': '',
          'download_url': 'UNKNOWN',
          'home_page': 'http://twistedmatrix.com/',
          'keywords': '',
          'license': 'MIT',
          'maintainer': '',
          'maintainer_email': '',
          'name': 'Twisted',
          'package_url': 'http://pypi.org/project/Twisted',
          'platform': 'UNKNOWN',
          'release_url': 'http://pypi.org/project/Twisted/12.3.0',
          'requires_python': None,
          'stable_version': None,
          'summary': 'An asynchronous networking '
                     'framework written in '
                     'Python',
          'version': '12.3.0'},
 'urls': [{...}, {...}]}
```