Session-14

September 13, 2019

1 Calculator

You can use Python as a calculator. Operators are similar to C or Fortran.

```
In [1]: 3 + 2

Out[1]: 5

In [2]: 3 * 2

Out[2]: 6

In [3]: 3 ** 2

Out[3]: 9

In [4]: 3 - 2

Out[4]: 1
```

Older versions of python perform integer division for / operator. Latest version does float division. To force integer division, use // operator. Make sure either numerator or denominator is a float to ask for float division.

```
In [1]: a = 3 // 2

In [2]: a

Out[2]: 1

In [3]: 3.0 / 2.0

Out[3]: 1.5

In [9]: type(a)

Out[9]: float
```

You can import a module to start using all the features of that module. The math module is important to offer you the features of a scientific calculator.

```
In [10]: import math
```

Using the function dir, you can inspect the methods and the data offered by any object or module. You can use it at any depth of the object heirarchy.

```
In [11]: dir(math)
```

```
Out[11]: ['__doc__',
          '__name__',
          '__package__',
          'acos',
          'acosh',
          'asin',
          'asinh',
          'atan',
          'atan2',
          'atanh',
          'ceil',
          'copysign',
          'cos',
          'cosh',
          'degrees',
          'e',
          'erf',
          'erfc',
          'exp',
          'expm1',
          'fabs',
          'factorial',
          'floor',
          'fmod',
          'frexp',
          'fsum',
          'gamma',
          'hypot',
          'isinf',
          'isnan',
          'ldexp',
          'lgamma',
          'log',
          'log10',
          'log1p',
          'modf',
          'pi',
          'pow',
          'radians',
          'sin',
          'sinh',
          'sqrt',
          'tan',
          'tanh',
          'trunc']

In [13]: math.sin(a)

Out[13]: 0.9974949866040544

In [15]: math.pi

Out[15]: 3.141592653589793

In [16]: type(math.pi)

Out[16]: float

In [17]: math.e

Out[17]: 2.718281828459045
```

The module sys has information about the limits of the system representation of data types.

```
In [18]: import sys

In [19]: dir(sys)

Out[19]: ['__displayhook__',
          '__doc__',
          '__excepthook__',
          '__name__',
          '__package__',
          '__stderr__',
          '__stdin__',
          '__stdout__',
          '_clear_type_cache',
          '_current_frames',
          '_getframe',
          '_git',
          '_multiarch',
          'api_version',
          'argv',
          'builtin_module_names',
          'byteorder',
          'call_tracing',
          'callstats',
          'copyright',
          'displayhook',
          'dont_write_bytecode',
          'exc_clear',
          'exc_info',
          'exc_type',
          'excepthook',
          'exec_prefix',
          'executable',
          'exit',
          'exitfunc',
          'flags',
          'float_info',
          'float_repr_style',
          'getcheckinterval',
          'getdefaultencoding',
          'getdlopenflags',
          'getfilesystemencoding',
          'getprofile',
          'getrecursionlimit',
          'getrefcount',
          'getsizeof',
          'gettrace',
          'hexversion',
          'last_traceback',
          'last_type',
          'last_value',
          'long_info',
          'maxint',
          'maxsize',
          'maxunicode',
          'meta_path',
          'modules',
          'path',
          'path_hooks',
          'path_importer_cache',
          'platform',
          'prefix',
          'ps1',
          'ps2',
          'ps3',
          'py3kwarning',
          'pydebug',
          'setcheckinterval',
          'setdlopenflags',
          'setprofile',
          'setrecursionlimit',
          'settrace',
          'stderr',
          'stdin',
          'stdout',
          'subversion',
          'version',
          'version_info',
          'warnoptions']
```

Use the help function to ask for help on any module or a method or data under it. You can also use it to ask for names of modules that you can import.

```
In [4]: help('modules')
```

```
Please wait a moment while I gather a list of all available modules...



/usr/local/lib/python2.7/dist-packages/matplotlib/cbook.py:136: MatplotlibDeprecationWarning: T
  warnings.warn(message, mplDeprecation, stacklevel=1)
/usr/lib/python2.7/pkgutil.py:122: VisibleDeprecationWarning: zmq.eventloop.minitornado is depr
    Install tornado itself to use zmq with the tornado IOLoop.

  for item in walk_packages(path, name+'.', onerror):
/usr/lib/python2.7/dist-packages/IPython/kernel/__init__.py:13: ShimWarning: The `IPython.kerne
  "You should import from ipykernel or jupyter_client instead.", ShimWarning)


BaseHTTPServer      bisect              jupyter             runpy
Bastion             bleach              jupyter_client      samba
CDROM               bs4                 jupyter_console     scandir
CGIHTTPServer       bsddb               jupyter_core        scanext
Canvas              bz2                 keyring             sched
ConfigParser        cPickle             keyrings            scipy
Cookie              cProfile            keyword             secretstorage
Crypto              cStringIO           ldb                 select
DLFCN               cairo               lib2to3             send2trash
Dialog              calendar            libxml2             sets
DistUtilsExtra      certifi             libxml2mod          setuptools
DocXMLRPCServer     cgi                 linecache           sgmllib
FileDialog          cgitb               linuxaudiodev       sha
FixTk               chardet             locale              shelve
HTMLParser          cherrypy            logging             shlex
IN                  chunk               lsb_release         shutil
IPython             cmath               lxml                signal
MimeWriter          cmd                 macpath             simplegeneric
OpenSSL             code                macurl2path         simplejson
PIL                 codecs              mailbox             singledispatch
PyQt4               codeop              mailcap             singledispatch_helpers
PyQt5               collections         markdown            sip
Queue               colorsys            markupbase          sipconfig
ScrolledText        commands            markupsafe          sipconfig_nd
SimpleDialog        compileall          marshal             site
```

```
SimpleHTTPServer    compiler            math                sitecustomize
SimpleXMLRPCServer  concurrent          matplotlib          six
SocketServer        configparser        md5                 smtpd
StringIO            contextlib          mechanize           smtplib
TYPES               cookielib           mhlib               sndhdr
Tix                 copy                mimetools           socket
Tkconstants         copy_reg            mimetypes           spglib
Tkdnd               crypt               mimify              spwd
Tkinter             cryptography        mistune             sqlite3
UserDict            cssselect           mmap                sre
UserList            cssutils            modulefinder        sre_compile
UserString          csv                 monty               sre_constants
_LWPCookieJar       ctypes              mpi4py              sre_parse
_MozillaCookieJar   cupsext             mpmath              ssl
__builtin__         curses              msgpack             stat
__future__          cycler              multifile           statvfs
_abcoll             cythonmagic         multiprocessing     storemagic
_ast                datetime            mutex               string
_bisect             dateutil            nbconvert           stringold
_bsddb              dbhash              nbformat            stringprep
_cffi_backend       dbm                 netifaces           strop
_codecs             dbus                netrc               struct
_codecs_cn          decimal             new                 subprocess
_codecs_hk          decorator           nis                 subprocess32
_codecs_iso2022     defusedxml          nntplib             sunau
_codecs_jp          difflib             notebook            sunaudio
_codecs_kr          dircache            ntpath              symbol
_codecs_tw          dis                 nturl2path          sympy
_collections        distutils           numbers             sympyprinting
_csv                dns                 numpy               symtable
_ctypes             doctest             olefile             sys
_ctypes_test        drv_libxml2         opcode              sysconfig
_curses             dsextras            operator            syslog
_curses_panel       dumbdbm             optparse            tabnanny
_dbus_bindings      dummy_thread        os                  tabulate
_dbus_glib_bindings dummy_threading     os2emxpath          talloc
_elementtree        easy_install        ossaudiodev         tarfile
_functools          email               palettable          tdb
_hashlib            encodings           pandas              telnetlib
_heapq              encutils            pandocfilters       tempfile
_hotshot            ensurepip           pango               terminado
_io                 entrypoints         pangocairo          termios
_json               enum                parser              test
_ldb_text           errno               pathlib2            test_regex
_locale             exceptions          pcardext            testpath
_lsprof             fcntl               pdb                 textwrap
_md5                feedparser          pexpect             this
_multibytecodec     filecmp             pickle              thread
```

```
_multiprocessing    fileinput           pickleshare         threading
_ordereddict        fnmatch             pickletools         tidy
_osx_support        formatter           pip                 time
_posixsubprocess    fpformat            pipes               timeit
_pyio               fractions           pkg_resources       tkColorChooser
_random             ftplib              pkgutil             tkCommonDialog
_regex              functools           platform            tkFileDialog
_regex_core         functools32         plistlib            tkFont
_ruamel_yaml        future_builtins     popen2              tkMessageBox
_scandir            gc                  poplib              tkSimpleDialog
_sha                genericpath         posix               toaiff
_sha256             getopt              posixfile           token
_sha512             getpass             posixpath           tokenize
_socket             gettext             pprint              tornado
_sqlite3            gi                  profile             trace
_sre                gio                 prometheus_client   traceback
_ssl                glib                prompt_toolkit      traitlets
_strptime           glob                pstats              ttk
_struct             gobject             pty                 tty
_symtable           grp                 ptyprocess          turtle
_sysconfigdata      gtk                 pwd                 types
_sysconfigdata_nd   gtkunixprint        py_compile          unicodedata
_tdb_text           gzip                pyclbr              unittest
_testcapi           hashlib             pydispatch          urllib
_threading_local    heapq               pydoc               urllib2
_tkinter            hmac                pydoc_data          urllib3
_warnings           hotshot             pyexpat             urlparse
_weakref            hpmudext            pygments            user
_weakrefset         html5_parser        pygtk               uu
_yaml               html5lib            pygtkcompat         uuid
abc                 htmlentitydefs      pylab               warnings
aifc                htmllib             pymatgen            wave
antigravity         httplib             pynotify            wcwidth
anydbm              idna                pyparsing           weakref
apsw                ihooks              pytz                webbrowser
argparse            imaplib             qtconsole           webencodings
array               imghdr              quopri              webob
asn1crypto          imp                 random              wheel
ast                 importlib           re                  whichdb
asynchat            imputil             readline            widgetsnbextension
asyncore            indicator_keyboard  regex               wsgiref
atexit              inspect             reportlab           xdg
atk                 io                  repoze              xdrlib
audiodev            ipaddress           repr                xml
audioop             ipykernel           requests            xmllib
autoreload          ipykernel_launcher  resource            xmlrpclib
backports           ipython_genutils    rexec               xxsubtype
backports_abc       ipywidgets          rfc822              yaml
```

```
base64              itertools           rlcompleter         zipfile
bdb                 jinja2              rmagic              zipimport
binascii            json                robotparser         zlib
binhex              jsonschema          routes              zmq

Enter any module name to get more help.  Or, type "modules spam" to search
for modules whose descriptions contain the word "spam".
```

```
In [3]: import sys

In [4]: help(sys.maxint)

Help on int object:

class int(object)
 |  int(x=0) -> int or long
 |  int(x, base=10) -> int or long
 |
 |  Convert a number or string to an integer, or return 0 if no arguments
 |  are given.  If x is floating point, the conversion truncates towards zero.
 |  If x is outside the integer range, the function returns a long instead.
 |
 |  If x is not a number or if base is given, then x must be a string or
 |  Unicode object representing an integer literal in the given base.  The
 |  literal can be preceded by '+' or '-' and be surrounded by whitespace.
 |  The base defaults to 10.  Valid bases are 0 and 2-36.  Base 0 means to
 |  interpret the base from the string as an integer literal.
 |  >>> int('0b100', base=0)
 |  4
 |
 |  Methods defined here:
 |
 |  __abs__(...)
 |      x.__abs__() <==> abs(x)
 |
 |  __add__(...)
 |      x.__add__(y) <==> x+y
 |
 |  __and__(...)
 |      x.__and__(y) <==> x&y
 |
 |  __cmp__(...)
 |      x.__cmp__(y) <==> cmp(x,y)
 |
 |  __coerce__(...)
 |      x.__coerce__(y) <==> coerce(x, y)
```

```
|
|  __div__(...)
|      x.__div__(y) <==> x/y
|
|  __divmod__(...)
|      x.__divmod__(y) <==> divmod(x, y)
|
|  __float__(...)
|      x.__float__() <==> float(x)
|
|  __floordiv__(...)
|      x.__floordiv__(y) <==> x//y
|
|  __format__(...)
|
|  __getattribute__(...)
|      x.__getattribute__('name') <==> x.name
|
|  __getnewargs__(...)
|
|  __hash__(...)
|      x.__hash__() <==> hash(x)
|
|  __hex__(...)
|      x.__hex__() <==> hex(x)
|
|  __index__(...)
|      x[y:z] <==> x[y.__index__():z.__index__()]
|
|  __int__(...)
|      x.__int__() <==> int(x)
|
|  __invert__(...)
|      x.__invert__() <==> ~x
|
|  __long__(...)
|      x.__long__() <==> long(x)
|
|  __lshift__(...)
|      x.__lshift__(y) <==> x<<y
|
|  __mod__(...)
|      x.__mod__(y) <==> x%y
|
|  __mul__(...)
|      x.__mul__(y) <==> x*y
|
|  __neg__(...)
|      x.__neg__() <==> -x
|
|  __nonzero__(...)
|      x.__nonzero__() <==> x != 0
|
|  __oct__(...)
|      x.__oct__() <==> oct(x)
|
|  __or__(...)
|      x.__or__(y) <==> x|y
|
|  __pos__(...)
|      x.__pos__() <==> +x
|
|  __pow__(...)
|      x.__pow__(y[, z]) <==> pow(x, y[, z])
|
|  __radd__(...)
|      x.__radd__(y) <==> y+x
|
|  __rand__(...)
|      x.__rand__(y) <==> y&x
|
|  __rdiv__(...)
|      x.__rdiv__(y) <==> y/x
|
|  __rdivmod__(...)
|      x.__rdivmod__(y) <==> divmod(y, x)
|
|  __repr__(...)
|      x.__repr__() <==> repr(x)
|
|  __rfloordiv__(...)
|      x.__rfloordiv__(y) <==> y//x
|
|  __rlshift__(...)
|      x.__rlshift__(y) <==> y<<x
|
|  __rmod__(...)
|      x.__rmod__(y) <==> y%x
|
|  __rmul__(...)
|      x.__rmul__(y) <==> y*x
|
|  __ror__(...)
|      x.__ror__(y) <==> y|x
|
|  __rpow__(...)
|      y.__rpow__(x[, z]) <==> pow(x, y[, z])
|
|  __rrshift__(...)
|      x.__rrshift__(y) <==> y>>x
|
|  __rshift__(...)
|      x.__rshift__(y) <==> x>>y
|
|  __rsub__(...)
|      x.__rsub__(y) <==> y-x
|
|  __rtruediv__(...)
|      x.__rtruediv__(y) <==> y/x
|
|  __rxor__(...)
|      x.__rxor__(y) <==> y^x
|
|  __str__(...)
|      x.__str__() <==> str(x)
|
|  __sub__(...)
|      x.__sub__(y) <==> x-y
|
|  __truediv__(...)
|      x.__truediv__(y) <==> x/y
|
|  __trunc__(...)
|      Truncating an Integral returns itself.
|
|  __xor__(...)
|      x.__xor__(y) <==> x^y
|
|  bit_length(...)
|      int.bit_length() -> int
|
|      Number of bits necessary to represent self in binary.
|      >>> bin(37)
|      '0b100101'
|      >>> (37).bit_length()
|      6
|
|  conjugate(...)
|      Returns self, the complex conjugate of any int.
|
|  ----------------------------------------------------------------------
|  Data descriptors defined here:
|
|  denominator
```

```
 |      the denominator of a rational number in lowest terms
 |
 |  imag
 |      the imaginary part of a complex number
 |
 |  numerator
 |      the numerator of a rational number in lowest terms
 |
 |  real
 |      the real part of a complex number
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes defined here:
 |
 |  __new__ = <built-in method __new__ of type object>
 |      T.__new__(S, ...) -> a new object with type S, a subtype of T
```

```
In [21]: dir(sys.float_info)
```

```
Out[21]: ['__add__',
          '__class__',
          '__contains__',
          '__delattr__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getitem__',
          '__getslice__',
          '__gt__',
          '__hash__',
          '__init__',
          '__le__',
          '__len__',
          '__lt__',
          '__mul__',
          '__ne__',
          '__new__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__rmul__',
          '__setattr__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          'dig',
          'epsilon',
          'mant_dig',
          'max',
          'max_10_exp',
          'max_exp',
          'min',
          'min_10_exp',
          'min_exp',
          'n_fields',
          'n_sequence_fields',
          'n_unnamed_fields',
          'radix',
          'rounds']
```

```
In [22]: sys.float_info.min

Out[22]: 2.2250738585072014e-308

In [23]: sys.float_info.max

Out[23]: 1.7976931348623157e+308

In [24]: sys.float_info.epsilon

Out[24]: 2.220446049250313e-16

In [25]: 0.1**2

Out[25]: 0.010000000000000002

In [26]: a = 1e14

In [27]: b = 25.44

In [28]: c = 0.74

In [29]: (a+b)+c

Out[29]: 100000000000026.17

In [30]: a+(b+c)

Out[30]: 100000000000026.19
```

1.1 Lists

Sequences or lists can be made easily. They can be indexed in multiple ways. Slices can be made out of lists too.

```
In [5]: a1 = [1.0, 2.0, 3.0, 4.0, 5.0]

In [6]: a1[3]

Out[6]: 4.0

In [8]: a1[0:3]

Out[8]: [1.0, 2.0, 3.0]

In [10]: a1[2:-1]

Out[10]: [3.0, 4.0]

In [14]: a1[3] = 4.4

In [15]: a1

Out[15]: [1.0, 2.0, 3.0, 4.4, 5.0]
```

1.2 Homework

Try editing the cells above and practice simple arithmetic.

```
In [ ]:
```