DSE5002 Module 5 Pair Programming

HD Sheets, July 2024

Intro to some Python Ideas

Basic Variables

Python is dynamically typed, like R, so Python decides how to store variables when you declare them.

Python has floating points, integers, complex numbers, boolean and strings as the basic type elements

It does not have a factor like R does

The basic math operations work the same way they do in R

In [1]:
```
a=5
b=2
print(a+b)
```

```
7
```

In [2]:
```
#We can use the type function to see what type of variable on object is
type(a)
```

Out[2]:
```
int
```

In [3]:
```
a=5.0
type(a)
```

Out[3]:
```
float
```

Question

Why is a an integer in one case and a float in the other?

In [4]:
```
a="5"
type(a)
```

Out[4]:
```
str
```

Storage classes in Python, such as str and more complex types have built in member functions that operate on items in the class

The dir() function will show the variables in the class, which have underscores added, and functions which do not

In [6]:

```
dir(a)
```

```
Out[6]: ['__add__',
          '__class__',
          '__contains__',
          '__delattr__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getitem__',
          '__getnewargs__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__iter__',
          '__le__',
          '__len__',
          '__lt__',
          '__mod__',
          '__mul__',
          '__ne__',
          '__new__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__rmod__',
          '__rmul__',
          '__setattr__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          'capitalize',
          'casefold',
          'center',
          'count',
          'encode',
          'endswith',
          'expandtabs',
          'find',
          'format',
          'format_map',
          'index',
          'isalnum',
          'isalpha',
          'isascii',
          'isdecimal',
          'isdigit',
          'isidentifier',
          'islower',
          'isnumeric',
          'isprintable',
          'isspace',
          'istitle',
          'isupper',
          'join',
          'ljust',
          'lower',
          'lstrip',
          'maketrans',
          'partition',
          'removeprefix',
          'removesuffix',
          'replace',
          'rfind',
          'rindex',
          'rjust',
          'rpartition',
          'rsplit',
          'rstrip',
          'split',
          'splitlines',
          'startswith',
          'strip',
          'swapcase',
          'title',
          'translate',
          'upper',
          'zfill']
```

Looking at a string, there is along list of available functions

We'll define a more interesting string and see what some of these functions do.

Notice how the member functions are called, as the variable name, a period and then the function name and parenthesis

Not all functions in Python are member functions, this is just showing how to call member functions once you find them using dir()

In [7]:

```
a="Joe chased a leaf,"
print(a.upper())
print(a.lower())
print(a.title())
```

```
JOE CHASED A LEAF,
joe chased a leaf,
Joe Chased A Leaf,
```

In [8]:

```
a.split()
```

Out[8]:

```
['Joe', 'chased', 'a', 'leaf,']
```

The key idea here is that dir() can show you a bunch of useful functions available

In [10]:

```
a.__len__
```

Out[10]:

```
<method-wrapper '__len__' of str object at 0x000002623D8AAE90>
```

In [12]:

```
#this is an iPython "magic" command to see the user defined variables in use at the mc
# it is a lift from the Matlab system
# "magic" commands are utility commands in iPython or Jupyter, not part of python

%who
```

```
a        b
```

In [13]:

```
%whos
```

```
Variable   Type    Data/Info
----------------------------
a          str     Joe chased a leaf,
b          int     2
```

To see more about magic commands, see

https://ipython.readthedocs.io/en/stable/interactive/magics.html

Thee are 4 types of built-in data structures in basic python

list, tuple, dictionary and set

We'll talk about them next week

Numpy

Numpy stands for numerical python, it has array and vector data types defined within it

To create matrices and do matric operations in Python, people typically use Numpy

Many other structures and common python packages are built on top of numpy classes

To use Numpy, we have to import the package. Note that the package must already be installed in your environment

np is the classic appreciation or alias for numpy

In [15]:

```
import numpy as np
```

Numpy matrices

-must all have the same type of element

- are indexed by the row and column number

-but like most other language, the first row is 0, and the first column is also 0 with R indices start with 1, in python they start with 0

-think of the indices in python as the distance from the start of an array or matrix

In [16]:

```
x=np.matrix('1 2 3; 4 5 6; 7 8 9')
x
```

Out[16]:

```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

In [17]:

```
x[1,0]
```

Out[17]:

```
4
```

In [18]:

```
Action/Question

What is the index of the "8" in this matrix?

Test your answere
```

```
  Input In [18]
    What is the index of the "8" in this matrix?
                ^
SyntaxError: invalid syntax
```

In [19]:

```
type(x)
```

Out[19]:

```
numpy.matrix
```

In [20]:

```
# dtype is an attribute of a numpy array that indicates the type of elements in the ma

x.dtype
```

Out[20]:

```
dtype('int32')
```

In [21]:

```
x.size
```

Out[21]:

```
9
```

In [22]:

```
x.shape
```

Out[22]:

```
(3, 3)
```

In [23]:

```
dir(x)
```

```
Out[23]: ['A',
          'A1',
          'H',
          'I',
          'T',
          '__abs__',
          '__add__',
          '__and__',
          '__array__',
          '__array_finalize__',
          '__array_function__',
          '__array_interface__',
          '__array_prepare__',
          '__array_priority__',
          '__array_struct__',
          '__array_ufunc__',
          '__array_wrap__',
          '__bool__',
          '__class__',
          '__complex__',
          '__contains__',
          '__copy__',
          '__deepcopy__',
          '__delattr__',
          '__delitem__',
          '__dict__',
          '__dir__',
          '__divmod__',
          '__doc__',
          '__eq__',
          '__float__',
          '__floordiv__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getitem__',
          '__gt__',
          '__hash__',
          '__iadd__',
          '__iand__',
          '__ifloordiv__',
          '__ilshift__',
          '__imatmul__',
          '__imod__',
          '__imul__',
          '__index__',
          '__init__',
          '__init_subclass__',
          '__int__',
          '__invert__',
          '__ior__',
          '__ipow__',
          '__irshift__',
          '__isub__',
          '__iter__',
          '__itruediv__',
          '__ixor__',
          '__le__',
          '__len__',
          '__lshift__',
'__lt__',
'__matmul__',
'__mod__',
'__module__',
'__mul__',
'__ne__',
'__neg__',
'__new__',
'__or__',
'__pos__',
'__pow__',
'__radd__',
'__rand__',
'__rdivmod__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__rfloordiv__',
'__rlshift__',
'__rmatmul__',
'__rmod__',
'__rmul__',
'__ror__',
'__rpow__',
'__rrshift__',
'__rshift__',
'__rsub__',
'__rtruediv__',
'__rxor__',
'__setattr__',
'__setitem__',
'__setstate__',
'__sizeof__',
'__str__',
'__sub__',
'__subclasshook__',
'__truediv__',
'__xor__',
'_align',
'_collapse',
'_getitem',
'all',
'any',
'argmax',
'argmin',
'argpartition',
'argsort',
'astype',
'base',
'byteswap',
'choose',
'clip',
'compress',
'conj',
'conjugate',
'copy',
'ctypes',
'cumprod',
'cumsum',
'data',
 'diagonal',
 'dot',
 'dtype',
 'dump',
 'dumps',
 'fill',
 'flags',
 'flat',
 'flatten',
 'getA',
 'getA1',
 'getH',
 'getI',
 'getT',
 'getfield',
 'imag',
 'item',
 'itemset',
 'itemsize',
 'max',
 'mean',
 'min',
 'nbytes',
 'ndim',
 'newbyteorder',
 'nonzero',
 'partition',
 'prod',
 'ptp',
 'put',
 'ravel',
 'real',
 'repeat',
 'reshape',
 'resize',
 'round',
 'searchsorted',
 'setfield',
 'setflags',
 'shape',
 'size',
 'sort',
 'squeeze',
 'std',
 'strides',
 'sum',
 'swapaxes',
 'take',
 'tobytes',
 'tofile',
 'tolist',
 'tostring',
 'trace',
 'transpose',
 'var',
 'view']
```

Action

Add a cell and try some different operations from the member functions, see what they do

Specialized Matrices

In [24]:
```
x=np.identity(6)
x
```

Out[24]:
```
array([[1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 1.]])
```

In [25]:
```
x=np.ones((4,5))
x
```

Out[25]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

In [26]:
```
x=np.zeros((3,6))
x
```

Out[26]:
```
array([[0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.]])
```

To learn more about using Numpy to do linear algebra see:

https://numpy.org/doc/stable/user/absolute_beginners.html

Another option is

https://www.kaggle.com/code/legendadnan/numpy-tutorial-for-beginners-data-science

Pandas

Pandas is a libary that implies a data frame, much like the dataframes in R, or a data table in SQL

Each row is an observation, each column is a variable.

The columns are actually 1 dimensional numpy arrays (ie n x 1 matrics, for n rows)

There are an immense number of member functions to let us carry out operations on Pandas dataframes

Slicing, sorting and selecting work much like they do in R

We typically import pandas as pd

In [27]:
```
import pandas as pd
```

In [28]:

```
#edit this line so it contains the full path name for the Boston Asssessment_Roll_2024


infile="C:\\Users\\hdavi\\Dropbox\\Merrimack_Data_Science\\DSE5002_R+Python\\DSE5002_M
```

In [29]:

```
#import the file,  place it into a Pandas dataframe

Boston_roll=pd.read_csv(infile)
```

```
C:\Users\hdavi\AppData\Local\Temp\ipykernel_12416\3031399445.py:3: DtypeWarning: Colu
mns (21) have mixed types. Specify dtype option on import or set low_memory=False.
  Boston_roll=pd.read_csv(infile)
```

In [30]:

```
# We have a head function, just as in R

Boston_roll.head(5)
```

Out[30]:

	_id	PID	CM_ID	GIS_ID	ST_NUM	ST_NAME	UNIT_NUM	CITY	ZIP_CODE	BLDG_SE
0	1	100001000	NaN	100001000	104.0	PUTNAM ST	NaN	EAST BOSTON	2128.0	
1	2	100002000	NaN	100002000	197.0	Lexington ST	NaN	EAST BOSTON	2128.0	
2	3	100003000	NaN	100003000	199.0	Lexington ST	NaN	EAST BOSTON	2128.0	
3	4	100004000	NaN	100004000	201.0	Lexington ST	NaN	EAST BOSTON	2128.0	
4	5	100005000	NaN	100005000	203.0	Lexington ST	NaN	EAST BOSTON	2128.0	

5 rows × 66 columns

In [31]:

```
#We can access columns by name, rowns by number
# Notice that 0:5 gives us all values from 0 to 4,  5 is not included

Boston_roll["CITY"][0:5]
```

Out[31]:

```
0    EAST BOSTON
1    EAST BOSTON
2    EAST BOSTON
3    EAST BOSTON
4    EAST BOSTON
Name: CITY, dtype: object
```

In [32]:

```
# we can index using two numbers, note the need to use the .iloc notation to do this

Boston_roll.iloc[5:10,7]
```

Out[32]:

```
5    EAST BOSTON
6    EAST BOSTON
7    EAST BOSTON
8    EAST BOSTON
9    EAST BOSTON
Name: CITY, dtype: object
```

In [33]:

```
#let's look at the member functions

dir(Boston_roll)
```

```
Out[33]: ['AC_TYPE',
          'BDRM_COND',
          'BED_RMS',
          'BLDG_SEQ',
          'BLDG_TYPE',
          'BLDG_VALUE',
          'BTHRM_STYLE1',
          'BTHRM_STYLE2',
          'BTHRM_STYLE3',
          'CD_FLOOR',
          'CITY',
          'CM_ID',
          'COM_UNITS',
          'CORNER_UNIT',
          'EXT_COND',
          'EXT_FNISHED',
          'FIREPLACES',
          'FULL_BTH',
          'GIS_ID',
          'GROSS_AREA',
          'GROSS_TAX',
          'HEAT_SYSTEM',
          'HEAT_TYPE',
          'HLF_BTH',
          'INT_COND',
          'INT_WALL',
          'KITCHENS',
          'KITCHEN_STYLE1',
          'KITCHEN_STYLE2',
          'KITCHEN_STYLE3',
          'KITCHEN_TYPE',
          'LAND_SF',
          'LAND_VALUE',
          'LIVING_AREA',
          'LU',
          'LUC',
          'LU_DESC',
          'MAIL_ADDRESSEE',
          'MAIL_CITY',
          'MAIL_STATE',
          'MAIL_STREET_ADDRESS',
          'MAIL_ZIP_CODE',
          'NUM_BLDGS',
          'NUM_PARKING',
          'ORIENTATION',
          'OVERALL_COND',
          'OWNER',
          'OWN_OCC',
          'PID',
          'PROP_VIEW',
          'RC_UNITS',
          'RES_FLOOR',
          'RES_UNITS',
          'ROOF_COVER',
          'ROOF_STRUCTURE',
          'SFYI_VALUE',
          'STRUCTURE_CLASS',
          'ST_NAME',
          'ST_NUM',
          'T',
'TOTAL_VALUE',
'TT_RMS',
'UNIT_NUM',
'YR_BUILT',
'YR_REMODEL',
'ZIP_CODE',
'_AXIS_LEN',
'_AXIS_ORDERS',
'_AXIS_TO_AXIS_NUMBER',
'_HANDLED_TYPES',
'__abs__',
'__add__',
'__and__',
'__annotations__',
'__array__',
'__array_priority__',
'__array_ufunc__',
'__array_wrap__',
'__bool__',
'__class__',
'__contains__',
'__copy__',
'__deepcopy__',
'__delattr__',
'__delitem__',
'__dict__',
'__dir__',
'__divmod__',
'__doc__',
'__eq__',
'__finalize__',
'__floordiv__',
'__format__',
'__ge__',
'__getattr__',
'__getattribute__',
'__getitem__',
'__getstate__',
'__gt__',
'__hash__',
'__iadd__',
'__iand__',
'__ifloordiv__',
'__imod__',
'__imul__',
'__init__',
'__init_subclass__',
'__invert__',
'__ior__',
'__ipow__',
'__isub__',
'__iter__',
'__itruediv__',
'__ixor__',
'__le__',
'__len__',
'__lt__',
'__matmul__',
'__mod__',
'__module__',
 '__mul__',
 '__ne__',
 '__neg__',
 '__new__',
 '__nonzero__',
 '__or__',
 '__pos__',
 '__pow__',
 '__radd__',
 '__rand__',
 '__rdivmod__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rfloordiv__',
 '__rmatmul__',
 '__rmod__',
 '__rmul__',
 '__ror__',
 '__round__',
 '__rpow__',
 '__rsub__',
 '__rtruediv__',
 '__rxor__',
 '__setattr__',
 '__setitem__',
 '__setstate__',
 '__sizeof__',
 '__str__',
 '__sub__',
 '__subclasshook__',
 '__truediv__',
 '__weakref__',
 '__xor__',
 '_accessors',
 '_accum_func',
 '_add_numeric_operations',
 '_agg_by_level',
 '_agg_examples_doc',
 '_agg_summary_and_see_also_doc',
 '_align_frame',
 '_align_series',
 '_append',
 '_arith_method',
 '_as_manager',
 '_attrs',
 '_box_col_values',
 '_can_fast_transpose',
 '_check_inplace_and_allows_duplicate_labels',
 '_check_inplace_setting',
 '_check_is_chained_assignment_possible',
 '_check_label_or_level_ambiguity',
 '_check_setitem_copy',
 '_clear_item_cache',
 '_clip_with_one_bound',
 '_clip_with_scalar',
 '_cmp_method',
 '_combine_frame',
 '_consolidate',
 '_consolidate_inplace',
 '_construct_axes_dict',
 '_construct_axes_from_arguments',
 '_construct_result',
 '_constructor',
 '_constructor_sliced',
 '_convert',
 '_count_level',
 '_data',
 '_dir_additions',
 '_dir_deletions',
 '_dispatch_frame_op',
 '_drop_axis',
 '_drop_labels_or_levels',
 '_ensure_valid_index',
 '_find_valid_index',
 '_flags',
 '_from_arrays',
 '_from_mgr',
 '_get_agg_axis',
 '_get_axis',
 '_get_axis_name',
 '_get_axis_number',
 '_get_axis_resolvers',
 '_get_block_manager_axis',
 '_get_bool_data',
 '_get_cleaned_column_resolvers',
 '_get_column_array',
 '_get_index_resolvers',
 '_get_item_cache',
 '_get_label_or_level_values',
 '_get_numeric_data',
 '_get_value',
 '_getitem_bool_array',
 '_getitem_multilevel',
 '_gotitem',
 '_hidden_attrs',
 '_id',
 '_indexed_same',
 '_info_axis',
 '_info_axis_name',
 '_info_axis_number',
 '_info_repr',
 '_init_mgr',
 '_inplace_method',
 '_internal_names',
 '_internal_names_set',
 '_is_copy',
 '_is_homogeneous_type',
 '_is_label_or_level_reference',
 '_is_label_reference',
 '_is_level_reference',
 '_is_mixed_type',
 '_is_view',
 '_iset_item',
 '_iset_item_mgr',
 '_iset_not_inplace',
 '_item_cache',
 '_iter_column_arrays',
 '_ixs',
 '_join_compat',
 '_logical_func',
 '_logical_method',
 '_maybe_cache_changed',
 '_maybe_update_cacher',
 '_metadata',
 '_mgr',
 '_min_count_stat_function',
 '_needs_reindex_multi',
 '_protect_consolidate',
 '_reduce',
 '_reduce_axis1',
 '_reindex_axes',
 '_reindex_columns',
 '_reindex_index',
 '_reindex_multi',
 '_reindex_with_indexers',
 '_rename',
 '_replace_columnwise',
 '_repr_data_resource_',
 '_repr_fits_horizontal_',
 '_repr_fits_vertical_',
 '_repr_html_',
 '_repr_latex_',
 '_reset_cache',
 '_reset_cacher',
 '_sanitize_column',
 '_series',
 '_set_axis',
 '_set_axis_name',
 '_set_axis_nocheck',
 '_set_is_copy',
 '_set_item',
 '_set_item_frame_value',
 '_set_item_mgr',
 '_set_value',
 '_setitem_array',
 '_setitem_frame',
 '_setitem_slice',
 '_slice',
 '_stat_axis',
 '_stat_axis_name',
 '_stat_axis_number',
 '_stat_function',
 '_stat_function_ddof',
 '_take_with_is_copy',
 '_to_dict_of_blocks',
 '_typ',
 '_update_inplace',
 '_validate_dtype',
 '_values',
 '_where',
 'abs',
 'add',
 'add_prefix',
 'add_suffix',
 'agg',
 'aggregate',
 'align',
 'all',
 'any',
 'append',
 'apply',
 'applymap',
 'asfreq',
 'asof',
 'assign',
 'astype',
 'at',
 'at_time',
 'attrs',
 'axes',
 'backfill',
 'between_time',
 'bfill',
 'bool',
 'boxplot',
 'clip',
 'columns',
 'combine',
 'combine_first',
 'compare',
 'convert_dtypes',
 'copy',
 'corr',
 'corrwith',
 'count',
 'cov',
 'cummax',
 'cummin',
 'cumprod',
 'cumsum',
 'describe',
 'diff',
 'div',
 'divide',
 'dot',
 'drop',
 'drop_duplicates',
 'droplevel',
 'dropna',
 'dtypes',
 'duplicated',
 'empty',
 'eq',
 'equals',
 'eval',
 'ewm',
 'expanding',
 'explode',
 'ffill',
 'fillna',
 'filter',
 'first',
 'first_valid_index',
 'flags',
 'floordiv',
 'from_dict',
 'from_records',
 'ge',
 'get',
 'groupby',
 'gt',
 'head',
 'hist',
 'iat',
 'idxmax',
 'idxmin',
 'iloc',
 'index',
 'infer_objects',
 'info',
 'insert',
 'interpolate',
 'isin',
 'isna',
 'isnull',
 'items',
 'iteritems',
 'iterrows',
 'itertuples',
 'join',
 'keys',
 'kurt',
 'kurtosis',
 'last',
 'last_valid_index',
 'le',
 'loc',
 'lookup',
 'lt',
 'mad',
 'mask',
 'max',
 'mean',
 'median',
 'melt',
 'memory_usage',
 'merge',
 'min',
 'mod',
 'mode',
 'mul',
 'multiply',
 'ndim',
 'ne',
 'nlargest',
 'notna',
 'notnull',
 'nsmallest',
 'nunique',
 'pad',
 'pct_change',
 'pipe',
 'pivot',
 'pivot_table',
 'plot',
 'pop',
 'pow',
 'prod',
 'product',
 'quantile',
 'query',
 'radd',
 'rank',
 'rdiv',
 'reindex',
 'reindex_like',
 'rename',
 'rename_axis',
 'reorder_levels',
 'replace',
 'resample',
 'reset_index',
 'rfloordiv',
 'rmod',
 'rmul',
 'rolling',
 'round',
 'rpow',
 'rsub',
 'rtruediv',
 'sample',
 'select_dtypes',
 'sem',
 'set_axis',
 'set_flags',
 'set_index',
 'shape',
 'shift',
 'size',
 'skew',
 'slice_shift',
 'sort_index',
 'sort_values',
 'squeeze',
 'stack',
 'std',
 'style',
 'sub',
 'subtract',
 'sum',
 'swapaxes',
 'swaplevel',
 'tail',
 'take',
 'to_clipboard',
 'to_csv',
 'to_dict',
 'to_excel',
 'to_feather',
 'to_gbq',
 'to_hdf',
 'to_html',
 'to_json',
 'to_latex',
 'to_markdown',
 'to_numpy',
 'to_parquet',
 'to_period',
 'to_pickle',
 'to_records',
 'to_sql',
 'to_stata',
 'to_string',
 'to_timestamp',
 'to_xarray',
 'to_xml',
 'transform',
 'transpose',
 'truediv',
 'truncate',
 'tz_convert',
 'tz_localize',
 'unstack',
 'update',
 'value_counts',
 'values',
 'var',
 'where',
 'xs']
```

Just a few options, eh?

Pandas can do a lot of different things....

We will see a lot more of Pandas later

If you want to work ahead a bit

https://www.kaggle.com/learn/pandas

https://pandas.pydata.org/docs/getting_started/tutorials.html