02.01-a-tour-of-python

February 21, 2020

1 Python

1.1

```
In [1]: 2 + 2
```

```
Out[1]: 4
```

```
In [2]: 2.0 + 2.5
```

```
Out[2]: 4.5
```

```
In [3]: 2 + 2.5
```

```
Out[3]: 4.5
```

1.2

Python<>=<>

```
In [4]: a = 0.2
```

1.3 String

```
In [5]: s = "hello world"
        s
```

```
Out[5]: 'hello world'
```

```
In [6]: s = 'hello world'
        s
```

```
Out[6]: 'hello world'
```

```
In [7]: s = """hello
        world"""
        print (s)
```

```
hello
world
```

```
In [8]: s = '''hello
        world'''
        print (s)
```

```
hello
world
```

```
In [9]: s = "hello" + " world"
        s
```

```
Out[9]: 'hello world'
```

```
In [10]: s[0]
```

```
Out[10]: 'h'
```

```
In [11]: s[-1]
```

```
Out[11]: 'd'
```

```
In [12]: s[0:5]
```

```
Out[12]: 'hello'
```

```
In [13]: s = "hello world"
         s.split()
```

```
Out[13]: ['hello', 'world']
```

```
In [14]: len(s)
```

```
Out[14]: 11
```

1.4 List

Python[]

```
In [15]: a = [1, 2.0, 'hello', 5 + 1.0]
         a

Out[15]: [1, 2.0, 'hello', 6.0]
```

```
In [16]: a + a

Out[16]: [1, 2.0, 'hello', 6.0, 1, 2.0, 'hello', 6.0]
```

```
In [17]: a[1]

Out[17]: 2.0
```

```
In [18]: len(a)

Out[18]: 4
```

```
In [19]: a.append("world")
         a

Out[19]: [1, 2.0, 'hello', 6.0, 'world']
```

1.5 Set

Python{}

```
In [20]: s = {2, 3, 4, 2}
         s

Out[20]: {2, 3, 4}
```

```
In [21]: len(s)

Out[21]: 3
```

```
In [22]: s.add(1)
         s

Out[22]: {1, 2, 3, 4}
```

```
In [23]: a = {1, 2, 3, 4}
         b = {2, 3, 4, 5}
         a & b

Out[23]: {2, 3, 4}
```

```
In [24]: a | b

Out[24]: {1, 2, 3, 4, 5}
```

```
In [25]: a - b

Out[25]: {1}
```

```
In [26]: a ^ b

Out[26]: {1, 5}
```

1.6 Dictionary

Python{key:value}Dictionary

```
In [27]: d = {'dogs':5, 'cats':4}
         d

Out[27]: {'dogs': 5, 'cats': 4}
```

```
In [28]: len(d)

Out[28]: 2
```

```
In [29]: d["dogs"]

Out[29]: 5
```

```
In [30]: d["dogs"] = 2
         d

Out[30]: {'dogs': 2, 'cats': 4}
```

```
In [31]: d["pigs"] = 7
         d

Out[31]: {'dogs': 2, 'cats': 4, 'pigs': 7}
```

```
In [32]: d.keys()

Out[32]: dict_keys(['dogs', 'cats', 'pigs'])
```

```
In [33]: d.values()

Out[33]: dict_values([2, 4, 7])
```

```
In [34]: d.items()

Out[34]: dict_items([('dogs', 2), ('cats', 4), ('pigs', 7)])
```

1.7 Numpy Arrays

Numpy

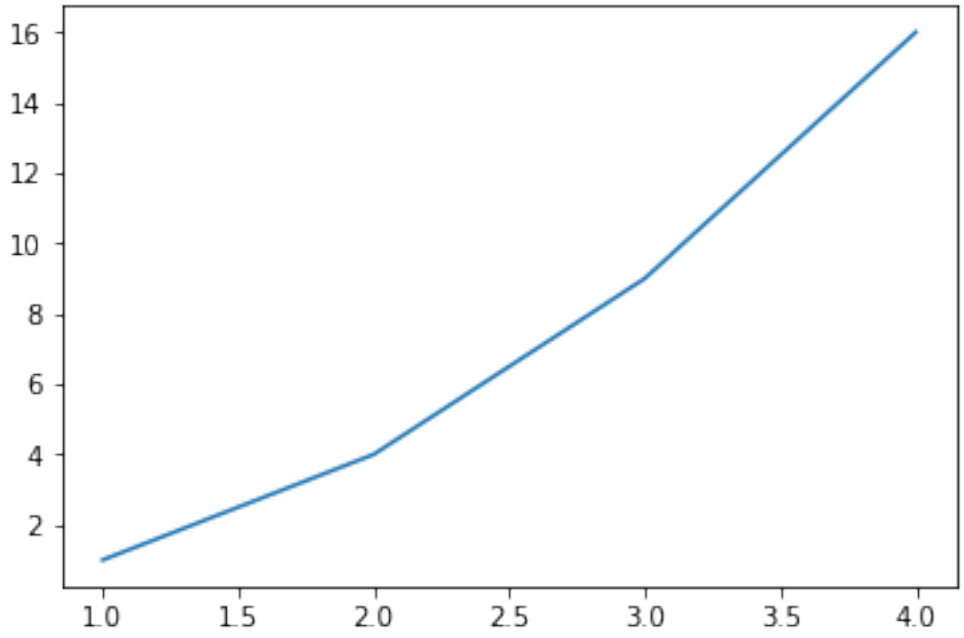
```
In [35]: from numpy import array
         a = array([1, 2, 3, 4])
         a

Out[35]: array([1, 2, 3, 4])
```

```
In [36]: a + 2

Out[36]: array([3, 4, 5, 6])

In [37]: a + a

Out[37]: array([2, 4, 6, 8])
```

1.8 Plot

PythonMATLAB

```
In [38]: %matplotlib inline
         from matplotlib.pyplot import plot
         plot(a, a**2)

Out[38]: [<matplotlib.lines.Line2D at 0x7f8dc4d13e80>]
```



1.9 Loop

```
In [39]: line = '1 2 3 4 5'
         fields = line.split()
         fields

Out[39]: ['1', '2', '3', '4', '5']

In [40]: total = 0
         for field in fields:
             total += int(field)
         total

Out[40]: 15
```

Python(List comprehension)

```
In [41]: numbers = [int(field) for field in fields]
         numbers
```

```
Out[41]: [1, 2, 3, 4, 5]
```

```
In [42]: sum(numbers)
```

```
Out[42]: 15
```

```
In [43]: sum([int(field) for field in line.split()])
```

```
Out[43]: 15
```

1.10 File IO

```
In [44]: cd ~
```

```
/home/jovyan
```

```
In [45]: f = open('data.txt', 'w')
         f.write('1 2 3 4\n')
         f.write('2 3 4 5\n')
         f.close()
```

```
In [46]: f = open('data.txt')
         data = []
         for line in f:
             data.append([int(field) for field in line.split()])
         f.close()
         data
```

```
Out[46]: [[1, 2, 3, 4], [2, 3, 4, 5]]
```

```
In [47]: for row in data:
             print (row)
```

```
[1, 2, 3, 4]
[2, 3, 4, 5]
```

```
In [48]: import os
         os.remove('data.txt')
```

1.11 Function

Pythondef

```
In [49]: def poly(x, a, b, c):
             y = a * x ** 2 + b * x + c
             return y

         x = 1
         poly(x, 1, 2, 3)
```

```
Out[49]: 6
```

Numpyx

```
In [50]: x = array([1, 2, 3])
         poly(x, 1, 2, 3)
```

```
Out[50]: array([ 6, 11, 18])
```

```
In [51]: from numpy import arange

         def poly(x, a = 1, b = 2, c = 3):
             y = a*x**2 + b*x + c
             return y

         x = arange(10)
         x
         array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
Out[51]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [52]: poly(x)
```

```
Out[52]: array([  3,   6,  11,  18,  27,  38,  51,  66,  83, 102])
```

```
In [53]: poly(x, b = 1)
```

```
Out[53]: array([ 3,  5,  9, 15, 23, 33, 45, 59, 75, 93])
```

1.12 Module

Pythonimport

```
In [54]: import os
```

```
In [55]: os.getpid()
```

```
Out[55]: 17
```

```
In [56]: os.sep
```

```
Out[56]: '/'
```

1.13 - Class

class Person(object)object __init__ selfC Javathis

```
In [57]: class Person(object):
             def __init__(self, first, last, age):
                 self.first = first
                 self.last = last
                 self.age = age
             def full_name(self):
                 return self.first + ' ' + self.last
```

```
In [58]: person = Person('Mertle', 'Sedgewick', 52)
```

```
In [59]: person.first
```

```
Out[59]: 'Mertle'
```

```
In [60]: person.full_name()
```

```
Out[60]: 'Mertle Sedgewick'
```

```
In [61]: person.last = 'Smith'
```

d

```
In [62]: person.critters = d
         person.critters
```

```
Out[62]: {'dogs': 2, 'cats': 4, 'pigs': 7}
```

1.14 Data from Web

```
In [63]: url = 'https://raw.githubusercontent.com/real-time-machine-learning/1-pandas-intro/mas
```

```
In [64]: # docker
         import urllib.request
         aapl_csv = urllib.request.urlopen(url)
         data = []
         for line in aapl_csv:
             line = line.decode() #
             line=line.strip('\n') #
             data.append(line.split(',')) #
         data[:4] #
```

```
---------------------------------------------------------------------------

gaierror                                  Traceback (most recent call last)

/opt/conda/lib/python3.6/urllib/request.py in do_open(self, http_class, req, **http_co
1317                 h.request(req.get_method(), req.selector, req.data, headers,
-> 1318                           encode_chunked=req.has_header('Transfer-encoding'))
1319             except OSError as err: # timeout error


/opt/conda/lib/python3.6/http/client.py in request(self, method, url, body, headers, e
1238         """Send a complete request to the server."""
-> 1239         self._send_request(method, url, body, headers, encode_chunked)
1240


/opt/conda/lib/python3.6/http/client.py in _send_request(self, method, url, body, head
1284             body = _encode(body, 'body')
-> 1285         self.endheaders(body, encode_chunked=encode_chunked)
1286


/opt/conda/lib/python3.6/http/client.py in endheaders(self, message_body, encode_chunk
1233             raise CannotSendHeader()
-> 1234         self._send_output(message_body, encode_chunked=encode_chunked)
1235


/opt/conda/lib/python3.6/http/client.py in _send_output(self, message_body, encode_chu
1025         del self._buffer[:]
-> 1026         self.send(msg)
1027


/opt/conda/lib/python3.6/http/client.py in send(self, data)
963             if self.auto_open:
--> 964                 self.connect()
965             else:


/opt/conda/lib/python3.6/http/client.py in connect(self)
1391
-> 1392             super().connect()
1393


/opt/conda/lib/python3.6/http/client.py in connect(self)
```

```
    935         self.sock = self._create_connection(
--> 936             (self.host,self.port), self.timeout, self.source_address)
    937         self.sock.setsockopt(socket.IPPROTO_TCP, socket.TCP_NODELAY, 1)


    /opt/conda/lib/python3.6/socket.py in create_connection(address, timeout, source_addres
    703     err = None
--> 704     for res in getaddrinfo(host, port, 0, SOCK_STREAM):
    705         af, socktype, proto, canonname, sa = res


    /opt/conda/lib/python3.6/socket.py in getaddrinfo(host, port, family, type, proto, flag
    744     addrlist = []
--> 745     for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
    746         af, socktype, proto, canonname, sa = res


    gaierror: [Errno -3] Temporary failure in name resolution


During handling of the above exception, another exception occurred:


    URLError                                  Traceback (most recent call last)

    <ipython-input-64-0e8507ee7da7> in <module>
      1 # docker
      2 import urllib.request
----> 3 aapl_csv = urllib.request.urlopen(url)
      4 data = []
      5 for line in aapl_csv:


    /opt/conda/lib/python3.6/urllib/request.py in urlopen(url, data, timeout, cafile, capat
    221     else:
    222         opener = _opener
--> 223     return opener.open(url, data, timeout)
    224
    225 def install_opener(opener):


    /opt/conda/lib/python3.6/urllib/request.py in open(self, fullurl, data, timeout)
    524             req = meth(req)
    525
--> 526         response = self._open(req, data)
    527
    528         # post-process response
```

```
    /opt/conda/lib/python3.6/urllib/request.py in _open(self, req, data)
    542         protocol = req.type
    543         result = self._call_chain(self.handle_open, protocol, protocol +
--> 544                                   '_open', req)
    545         if result:
    546             return result


    /opt/conda/lib/python3.6/urllib/request.py in _call_chain(self, chain, kind, meth_name
    502         for handler in handlers:
    503             func = getattr(handler, meth_name)
--> 504             result = func(*args)
    505             if result is not None:
    506                 return result


    /opt/conda/lib/python3.6/urllib/request.py in https_open(self, req)
   1359         def https_open(self, req):
   1360             return self.do_open(http.client.HTTPSConnection, req,
-> 1361                 context=self._context, check_hostname=self._check_hostname)
   1362
   1363         https_request = AbstractHTTPHandler.do_request_


    /opt/conda/lib/python3.6/urllib/request.py in do_open(self, http_class, req, **http_co
   1318                           encode_chunked=req.has_header('Transfer-encoding'))
   1319             except OSError as err: # timeout error
-> 1320                 raise URLError(err)
   1321             r = h.getresponse()
   1322         except:


    URLError: <urlopen error [Errno -3] Temporary failure in name resolution>
```

pandas

```
In [65]: ge_csv = urllib.request.urlopen(url)
         import pandas
         ge = pandas.read_csv(ge_csv, index_col=0, parse_dates=True)
         ge.head()


         ---------------------------------------------------------------------------

         gaierror                                  Traceback (most recent call last)
```

```
    /opt/conda/lib/python3.6/urllib/request.py in do_open(self, http_class, req, **http_co
   1317                 h.request(req.get_method(), req.selector, req.data, headers,
-> 1318                           encode_chunked=req.has_header('Transfer-encoding'))
   1319             except OSError as err: # timeout error


    /opt/conda/lib/python3.6/http/client.py in request(self, method, url, body, headers, e
   1238         """Send a complete request to the server."""
-> 1239         self._send_request(method, url, body, headers, encode_chunked)
   1240


    /opt/conda/lib/python3.6/http/client.py in _send_request(self, method, url, body, head
   1284             body = _encode(body, 'body')
-> 1285         self.endheaders(body, encode_chunked=encode_chunked)
   1286


    /opt/conda/lib/python3.6/http/client.py in endheaders(self, message_body, encode_chunk
   1233             raise CannotSendHeader()
-> 1234         self._send_output(message_body, encode_chunked=encode_chunked)
   1235


    /opt/conda/lib/python3.6/http/client.py in _send_output(self, message_body, encode_chu
   1025         del self._buffer[:]
-> 1026         self.send(msg)
   1027


    /opt/conda/lib/python3.6/http/client.py in send(self, data)
    963             if self.auto_open:
--> 964                 self.connect()
    965             else:


    /opt/conda/lib/python3.6/http/client.py in connect(self)
   1391
-> 1392             super().connect()
   1393


    /opt/conda/lib/python3.6/http/client.py in connect(self)
    935         self.sock = self._create_connection(
--> 936             (self.host,self.port), self.timeout, self.source_address)
    937         self.sock.setsockopt(socket.IPPROTO_TCP, socket.TCP_NODELAY, 1)
```

```
    /opt/conda/lib/python3.6/socket.py in create_connection(address, timeout, source_addres
    703     err = None
--> 704     for res in getaddrinfo(host, port, 0, SOCK_STREAM):
    705         af, socktype, proto, canonname, sa = res


    /opt/conda/lib/python3.6/socket.py in getaddrinfo(host, port, family, type, proto, flag
    744     addrlist = []
--> 745     for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
    746         af, socktype, proto, canonname, sa = res


    gaierror: [Errno -3] Temporary failure in name resolution


During handling of the above exception, another exception occurred:


    URLError                                  Traceback (most recent call last)

    <ipython-input-65-85d570c7e973> in <module>
----> 1 ge_csv = urllib.request.urlopen(url)
      2 import pandas
      3 ge = pandas.read_csv(ge_csv, index_col=0, parse_dates=True)
      4 ge.head()


    /opt/conda/lib/python3.6/urllib/request.py in urlopen(url, data, timeout, cafile, capat
    221     else:
    222         opener = _opener
--> 223     return opener.open(url, data, timeout)
    224
    225 def install_opener(opener):


    /opt/conda/lib/python3.6/urllib/request.py in open(self, fullurl, data, timeout)
    524             req = meth(req)
    525
--> 526         response = self._open(req, data)
    527
    528         # post-process response


    /opt/conda/lib/python3.6/urllib/request.py in _open(self, req, data)
    542         protocol = req.type
    543         result = self._call_chain(self.handle_open, protocol, protocol +
--> 544                                   '_open', req)
    545         if result:
```

```
    546             return result


    /opt/conda/lib/python3.6/urllib/request.py in _call_chain(self, chain, kind, meth_name
    502         for handler in handlers:
    503             func = getattr(handler, meth_name)
--> 504             result = func(*args)
    505             if result is not None:
    506                 return result


    /opt/conda/lib/python3.6/urllib/request.py in https_open(self, req)
   1359         def https_open(self, req):
   1360             return self.do_open(http.client.HTTPSConnection, req,
-> 1361                 context=self._context, check_hostname=self._check_hostname)
   1362
   1363         https_request = AbstractHTTPHandler.do_request_


    /opt/conda/lib/python3.6/urllib/request.py in do_open(self, http_class, req, **http_co
   1318                             encode_chunked=req.has_header('Transfer-encoding'))
   1319             except OSError as err: # timeout error
-> 1320                 raise URLError(err)
   1321             r = h.getresponse()
   1322         except:


    URLError: <urlopen error [Errno -3] Temporary failure in name resolution>
```

In [66]: #