01-Scope-and-iterators

November 21, 2019

1 Assignment semantics

```
In [3]: a=[1,2,3]
        b=a
        b[0]=3
        print(a)
```

```
[3, 2, 3]
```

2 Classes

`__init__(self,...)` is the constructor `classname()`
Variables set in class body are class members
Object members are initialized in constructor as `self.member = value`
No member hiding!

```
In [33]: class counter:
                 '''counter is a class used for counting'''
                 # number of counter objects
                 ncounter = 0
                 def __init__(self):
                         self.count = 0
                         print('__init__',self.ncounter) # read only access
                         # self.ncounter += 1 # tries to create a new member variable
                         counter.ncounter += 1
                 def get(self):
                         '''get the value of the counter'''
                         return self.count

                 def incr(self):
                         '''increment the counter
         only by one'''
                         self.count += 1

                 #def __str__(self):
```

```
                #    return 'counter: ' + str(self.count)

                def __repr__(self):
                    return '<object counter: ' + str(self.count) + '>'
```

```
In [34]: c=counter()
         c.incr()
         c.get()
         print(counter.ncounter, c.get())
         print(c)
         c
```

```
__init__ 0
1 1
<object counter: 1>
```

```
Out[34]: <object counter: 1>
```

3 Operator Overloading

x * y	x.__mult__(y)
x / y	x.__truediv__(y)
x // y	x.__floordiv(y)
x > y	x.__gt__(y)
x[y]	x.__getitem__(y)
x[y]=z	x.__setitem__(y,z)
del x[y]	x.__delitem__(y)
x in y	x.__contains__(y)
x += y	x.__iadd__(y)
x.y	x.__getattribute__(y)
x.y = z	x.__setattr__(y,z)
del x.y	x.__delattr__(y)

```
In [7]: class Compleks:
            '''Alternative to complex to indicate operator overloading'''
            def __init__(self, r=0.0,i=0.0):
                self.real = r
                self.img = i
            def __add__(self,rhs):                          # x + y
                return Compleks(self.real+rhs.real, self.img+rhs.img)
            def __sub__(self,rhs):                          # x - y
                return Compleks(self.real-rhs.real, self.img-rhs.img)
            def __mul__(self,rhs):                          # x * y
                return Compleks(self.real*rhs.real-self.img*rhs.img, self.img*rhs.real+rhs.img*s
            def __truediv__(self,rhs):                      # x / y
                t = rhs.real**2 + rhs.img **2
                return Compleks((self.real*rhs.real+self.img*rhs.img)/t, (self.img*rhs.real-rhs.
            def __eq__(self,rhs):                             # x == y
                return self.real == rhs.real and self.img == rhs.img
            def __ne__(self,rhs):                             # x != y
                return not (self == rhs)
            def __pow__(self,i):                             #  x ** n
                if not isinstance(i,int):
                    return None
                if i < 0:
                    return Compleks(1,0)/(self ** -i)
                if i == 0:
                    return Compleks(1,0)
                if i == 1:
                    return Compleks(self.real, self.img)
                if i % 2 == 0:
                    nv = self ** (i // 2)
                    return nv*nv
                else:
                    nv = self ** (i // 2)
                    return nv*nv*self
            def __repr__(self):                             # repr(x)
                return str(self.real) + ("-" if self.img < 0 else "+") + str(abs(self.img)) + "i
```

```
In [11]: a = Compleks(3,4)
         b = Compleks(2,3)
         print(a,b)
         print(a+b)
         c=a*b
         print(c)
         print(c/a)
         print(a==c,a==a,a != b)
         print(a**20)
         a+=c                   # maps to a = a + c --> a = a.__add__(c)
         b*=c
         c/=c
         print(a,b,c)
```

```
3+4i 2+3i
5+7i
-6+17i
2.0+3.0i
False True True
91004468168113-28515500892816i
-3+21i -63+16i 1.0+0.0i
```

```
In [11]: class LList:
             '''Linked list implementation. Each node is a binary list [value,next]'''
             class Node:
```

```
        '''Just illustrates nested classes'''
        def __init__(self, v, n):
            self.val = v
            self.next = n
        def __str__(self):
            return "( " + str(self.val) + ", " + str(self.next) + " )"

    def __init__(self,vals=[]):
        self.head = self.last = None
        for v in vals:
            self.append(v)
    def append(self,v):
        if self.last == None:
            # very first element
            self.head = self.last = LList.Node(v,None)     # how to use nested classes!
        else:
            self.last.next=LList.Node(v,None)
            self.last = self.last.next
    def __getitem__(self,no):
        count = 0
        ptr = self.head
        while count < no:
            if ptr:
                ptr = ptr.next
            else:
                raise IndexError
            count += 1
        if ptr:
            return ptr.val
        else:
            raise IndexError
    def __setitem__(self,no,val):
        count = 0
        ptr = self.head
        while count < no:
            if ptr:
                ptr = ptr.next
            else:
                raise IndexError
            count += 1
        if ptr:
            ptr.val=val
            return ptr.val
        else:
            raise IndexError
    def __delitem__(self,no):
        count = 0
        prev = ptr = self.head
```

```
            while count < no:
                if ptr:
                    prev = ptr
                    ptr = ptr.next
                else:
                    raise IndexError
                count += 1
            if ptr:
                if ptr is self.head:
                    if self.head is self.last:
                        self.head = self.last = None
                    else:
                        self.head = self.head.next
                else:
                    if ptr == self.last:
                        self.last = prev
                    prev.next = ptr.next
            else:
                raise IndexError
```

```
In [10]: a=LList([1,2,3,4,5])
         a.append(2)
         a.append(4)
         print(a.head,a.last)
         print(a[1],a[0],a[3],a[5])
         a[5]=110
         print(a.head, a.last)
         del a[0]
         del a[5]
         print(a.head,a.last)
```

```
( 1, ( 2, ( 3, ( 4, ( 5, ( 2, ( 4, None ) ) ) ) ) ) ) ( 4, None )
2 1 4 2
( 1, ( 2, ( 3, ( 4, ( 5, ( 110, ( 4, None ) ) ) ) ) ) ) ( 4, None )
( 2, ( 3, ( 4, ( 5, ( 110, None ) ) ) ) ) ( 110, None )
```

4 Writing Iterators

Following loops are equivalent

```
In [17]: v = [1,5,6,2]
         for i in v:
             #loop body
             print(i)
         #--------------
         vit = iter(v)
         try:
             while True:
                 i = next(vit)
                 #loop body
                 print(i)
         except StopIteration:
             pass   # do nothing
```

```
1
5
6
2
1
5
6
2
```

4.1 Fibonacci example

```
In [12]: class Fibonacci:
             def __init__(self,n):
                 self.a, self.b = 0,1
                 self.icount = 0
                 self.n = n
             def __iter__(self):
                 '''returning the iterator. Always returns the same iterator in this case'''
                 return self
             def __next__(self):
                 self.icount += 1
                 if self.icount > self.n:
                     raise StopIteration
                 else:
                     self.a , self.b = self.b , self.a + self.b
                     return self.b
```

```
In [13]: for i in Fibonacci(10):
             print(i)
```

```
1
2
3
5
8
13
21
34
```

```
55
89
```

Returning same object in `__iter__` causes problems when same iterator is active multiple times

```
In [21]: a=Fibonacci(6)
         for i in a:
             for j in a:
                 print(i,j,i*j)
```

```
1 2 2
1 3 3
1 5 5
1 8 8
1 13 13
```

5 Iterator for a Data Structure

Implement `__iter__(self)`, `__next__(self)`, raise `StopIteration` at the end. Following is returning itself as the iterator which has problems.

```
In [5]: class LList:
            '''Linked list implementation. An iterator is added.'''
            class Node:
                def __init__(self, v,n):
                    self.val, self.next = v, n
                def __str__(self):
                    return "( " + str(self.val) + ", " + str(self.next) + " )"

            def __init__(self,vals=[]):
                self.head = self.last = None
                for v in vals:
                    self.append(v)
            def append(self,v):
                if self.last == None:
                    # very first element
                    self.head = self.last = LList.Node(v,None)
                else:
                    self.last.next = LList.Node(v,None)
                    self.last = self.last.next
            def __getitem__(self,no):
                count = 0
                ptr = self.head
                while count < no:
                    if ptr:
```

```
                ptr = ptr.next    # next
            else:
                raise IndexError
            count += 1
        if ptr:
            return ptr.val
        else:
            raise IndexError
    def __setitem__(self,no,val):
        count = 0
        ptr = self.head
        while count < no:
            if ptr:
                ptr = ptr.next
            else:
                raise IndexError
            count += 1
        if ptr:
            ptr.val=val
            return ptr.val
        else:
            raise IndexError
    def __delitem__(self,no):
        count = 0
        prev = ptr = self.head
        while count < no:
            if ptr:
                prev = ptr
                ptr = ptr.next
            else:
                raise IndexError
            count += 1
        if ptr:
            if ptr is self.head:
                if self.head is self.last:
                    self.head = self.last = None
                else:
                    self.head = self.head.val
            else:
                if ptr == self.last:
                    self.last = prev
                prev.next = ptr.next
        else:
            raise IndexError
    def __str__(self):
        ret="["
        ptr = self.head
        while True:
```

```
                    if ptr:
                        ret += str(ptr.val)
                    else:
                        break
                    ptr = ptr.next
                    if ptr:
                        ret += " -> "
                ret += ']\n'
                return ret
            def __iter__(self):
                self.itptr = self.head
                return self
            def __next__(self):
                if self.itptr == None:
                    raise StopIteration
                else:
                    val=self.itptr.val
                    self.itptr = self.itptr.next
                    return val
```

```
In [6]: a=LList([1,2,3,4,5])
        a.append(110)
        a[2]=10
        del a[3]
        print(a[4])
        print(a)
        for i in a:
                print(i)
```

```
110
[1 -> 2 -> 10 -> 5 -> 110]

1
2
10
5
110
```

Iterator works in a single loop but when same structure is used multiple times:

```
In [7]: for i in a:
            for j in a:
                print(i,j,i*j)
```

```
1 1 1
1 2 2
1 10 10
1 5 5
```

```
1 110 110
```

Fix. Return a distinct object per iterator request:

```
In [4]: class LList2:
            '''Linked list implementation. Iterator reuse is fixed'''
            class Node:
                def __init__(self, v,n):
                    self.val, self.next = v, n
                def __str__(self):
                    return "( " + str(self.val) + ", " + str(self.next) + " )"


            def __init__(self,vals=[]):
                self.head = self.last = None
                for v in vals:
                    self.append(v)

            def append(self,v):
                if self.last == None:
                    # very first element
                    self.head = self.last = LList2.Node(v,None)
                else:
                    self.last.next = LList2.Node(v,None)
                    self.last = self.last.next

            def __getitem__(self,no):
                count = 0
                ptr = self.head
                while count < no:
                    if ptr:
                        ptr = ptr.next    # next
                    else:
                        raise IndexError
                    count += 1
                if ptr:
                    return ptr.val
                else:
                    raise IndexError

            def __setitem__(self,no,val):
                count = 0
                ptr = self.head
                while count < no:
                    if ptr:
                        ptr = ptr.next
                    else:
```

```
                raise IndexError
            count += 1
        if ptr:
            ptr.val=val
            return ptr.val
        else:
            raise IndexError

    def __delitem__(self,no):
        count = 0
        prev = ptr = self.head
        while count < no:
            if ptr:
                prev = ptr
                ptr = ptr.next
            else:
                raise IndexError
            count += 1
        if ptr:
            if ptr is self.head:
                if self.head is self.last:
                    self.head = self.last = None
                else:
                    self.head = self.head.val
            else:
                if ptr == self.last:
                    self.last = prev
                prev.next = ptr.next
        else:
            raise IndexError

    def __str__(self):
        ret="["
        ptr = self.head
        while True:
            if ptr:
                ret += str(ptr.val)
            else:
                break
            ptr = ptr.next
            if ptr:
                ret += " -> "
        ret += "]"
        return ret

    def __iter__(self):
        '''return a brand new iterator'''
        return self.LListIterator(self)
    # yes, nested iterators possible
    class LListIterator:
        def __init__(self,llist):
            self.llist = llist
            self.itptr = llist.head

        def __next__(self):
            if self.itptr == None:
                raise StopIteration
            else:
                val=self.itptr.val
                self.itptr = self.itptr.next
                return val
```

```
In [5]: a=LList2([1,20,32])
        a.append(110)
        a[2]=10
        print(a)
```

```
[1 -> 20 -> 10 -> 110]
```

```
In [6]: for i in a:
            for j in a:
                print(i,j,i*j)
```

```
1 1 1
1 20 20
1 10 10
1 110 110
20 1 20
20 20 400
20 10 200
20 110 2200
10 1 10
10 20 200
10 10 100
10 110 1100
110 1 110
110 20 2200
110 10 1100
110 110 12100
```

5.1 An Iterator for a Tree

Following is a Binary Search Tree implementation. Implementation of iterator is tricky. Because computation of next value of a value **v** requires a repeated search as “find smallest element > **v**”.

Which is possible but not time efficient. A solution is to use next pointers on each node which requires extra storage and insertion/deletion becomes complicated.

5.2 Generators

A better solution is to use a generator. Generators are objects keeping the state of a computation independent from the current run time stack. You can go back to a generator, generator makes computation and yields back a value. The computation in the generator co exists with the existing computation. A recursive traversal like a tree is put into a generator in the following example. Any function using `yield` returns a `Generator` object immediately. When you call `next()` on a generator it will resume execution until it yields a value. When it yields execution returns to function calling `next()`.

```
In [7]: class BSTree:
            def __init__(self):
                self.node = None # empty tree
            def __getitem__(self, key):
                if self.node == None:
                    raise KeyError
                elif key < self.node[0]:  # search in left
                    return self.left[key]
                elif key > self.node[0]: # search in right
                    return self.right[key]
                else:
                    return self.node[1]  # return node content

            def __setitem__(self, key, val):
                if self.node == None:
                    self.node = (key,val)
                    self.left = BSTree()        # empty tree
                    self.right = BSTree()        # empty tree
                elif key < self.node[0]:
                    self.left[key] = val  # insert to left
                elif key > self.node[0]:
                    self.right[key] = val # insert to right
                else:
                    self.node = (key,val) # update node
            def __str__(self):
                if self.node == None:
                    return '*'
                else:
                    return '[' + str(self.left) + ', ' + \
                        str(self.node) + ', ' + \
                        str(self.right) + ']'

            def traverse(self):
                '''Generator function returning traverse'''
                if self.node != None:
                    # this is how you recurse in generators
                    # > python 3.4  you can replace loop by:
                    #   yield from self.left.traverse()
                    # which is more efficient
                    for vals in self.left.traverse():
                        yield vals
                    yield self.node
                    for vals in self.right.traverse():
                        yield vals
```

```
In [8]: c = BSTree()
        for (k,v) in [(5,4),(8,6),(4,3),(2,6),(7,12)]:
                c[k] = v
        print(str(c))
        print('value for 2 is ', str(c[2]))
```

```
[[[*, (2, 6), *], (4, 3), *], (5, 4), [[*, (7, 12), *], (8, 6), *]]
value for 2 is  6
```

```
In [9]: def fibonacci(n):
            a,b = 0,1
            counter = 0
            while counter < n:
                a,b = b, a+b
                yield a
                counter = counter+1
```

```
In [10]: a=fibonacci(10)
         dir(a)
```

```
Out[10]: ['__class__',
          '__del__',
          '__delattr__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__gt__',
          '__hash__',
          '__init__',
          '__iter__',
          '__le__',
          '__lt__',
          '__name__',
          '__ne__',
          '__new__',
          '__next__',
          '__qualname__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__setattr__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          'close',
          'gi_code',
          'gi_frame',
          'gi_running',
          'gi_yieldfrom',
          'send',
          'throw']
```

```
In [11]: a=fibonacci(5)
         for i in a:
             for b in a:
                 print(i,b,i*b)
```

```
1 1 1
1 2 2
1 3 3
1 5 5
```

```
In [12]: for i in c.traverse():
             for j in c.traverse():
                 print(i,j)
```

```
(2, 6) (2, 6)
(2, 6) (4, 3)
(2, 6) (5, 4)
(2, 6) (7, 12)
(2, 6) (8, 6)
(4, 3) (2, 6)
(4, 3) (4, 3)
(4, 3) (5, 4)
(4, 3) (7, 12)
(4, 3) (8, 6)
(5, 4) (2, 6)
(5, 4) (4, 3)
(5, 4) (5, 4)
(5, 4) (7, 12)
(5, 4) (8, 6)
(7, 12) (2, 6)
```

```
(7, 12) (4, 3)
(7, 12) (5, 4)
(7, 12) (7, 12)
(7, 12) (8, 6)
(8, 6) (2, 6)
(8, 6) (4, 3)
(8, 6) (5, 4)
(8, 6) (7, 12)
(8, 6) (8, 6)
```

```
In [13]: mylist=list(c.traverse())
         print(mylist)
```

```
[(2, 6), (4, 3), (5, 4), (7, 12), (8, 6)]
```

5.3 General usage of iterators

Iterators are used in many places for different objects like reading a file (following example), getting query results. Also many constructors or library functions use iteratable objects to get values like constructors (example above, `list` is initialized from tree traversal).

```
In [14]: fp=open("/etc/protocols","r")
         for line in fp:
             print(line,end='')
```

```
# Internet (IP) protocols
#
# Updated from http://www.iana.org/assignments/protocol-numbers and other
# sources.
# New protocols will be added on request if they have been officially
# assigned by IANA and are not historical.
# If you need a huge list of used numbers please install the nmap package.

ip        0        IP                # internet protocol, pseudo protocol number
hopopt        0        HOPOPT                # IPv6 Hop-by-Hop Option [RFC1883]
icmp        1        ICMP                # internet control message protocol
igmp        2        IGMP                # Internet Group Management
ggp        3        GGP                # gateway-gateway protocol
ipencap        4        IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
st        5        ST                # ST datagram mode
tcp        6        TCP                # transmission control protocol
egp        8        EGP                # exterior gateway protocol
igp        9        IGP                # any private interior gateway (Cisco)
pup        12        PUP                # PARC universal packet protocol
udp        17        UDP                # user datagram protocol
hmp        20        HMP                # host monitoring protocol
xns-idp        22        XNS-IDP                # Xerox NS IDP
```

```
rdp        27        RDP                # "reliable datagram" protocol
iso-tp4        29        ISO-TP4                # ISO Transport Protocol class 4 [RFC905]
dccp        33        DCCP                # Datagram Congestion Control Prot. [RFC4340]
xtp        36        XTP                # Xpress Transfer Protocol
ddp        37        DDP                # Datagram Delivery Protocol
idpr-cmtp 38        IDPR-CMTP        # IDPR Control Message Transport
ipv6        41        IPv6                # Internet Protocol, version 6
ipv6-route 43        IPv6-Route        # Routing Header for IPv6
ipv6-frag 44        IPv6-Frag        # Fragment Header for IPv6
idrp        45        IDRP                # Inter-Domain Routing Protocol
rsvp        46        RSVP                # Reservation Protocol
gre        47        GRE                # General Routing Encapsulation
esp        50        IPSEC-ESP        # Encap Security Payload [RFC2406]
ah        51        IPSEC-AH        # Authentication Header [RFC2402]
skip        57        SKIP                # SKIP
ipv6-icmp 58        IPv6-ICMP        # ICMP for IPv6
ipv6-nonxt 59        IPv6-NoNxt        # No Next Header for IPv6
ipv6-opts 60        IPv6-Opts        # Destination Options for IPv6
rspf        73        RSPF CPHB        # Radio Shortest Path First (officially CPHB)
vmtp        81        VMTP                # Versatile Message Transport
eigrp        88        EIGRP                # Enhanced Interior Routing Protocol (Cisco)
ospf        89        OSPFIGP                # Open Shortest Path First IGP
ax.25        93        AX.25                # AX.25 frames
ipip        94        IPIP                # IP-within-IP Encapsulation Protocol
etherip        97        ETHERIP                # Ethernet-within-IP Encapsulation [RFC3378]
encap        98        ENCAP                # Yet Another IP encapsulation [RFC1241]
#        99                        # any private encryption scheme
pim        103        PIM                # Protocol Independent Multicast
ipcomp        108        IPCOMP                # IP Payload Compression Protocol
vrrp        112        VRRP                # Virtual Router Redundancy Protocol [RFC5798]
l2tp        115        L2TP                # Layer Two Tunneling Protocol [RFC2661]
isis        124        ISIS                # IS-IS over IPv4
sctp        132        SCTP                # Stream Control Transmission Protocol
fc        133        FC                # Fibre Channel
mobility-header 135 Mobility-Header # Mobility Support for IPv6 [RFC3775]
udplite        136        UDPLite                # UDP-Lite [RFC3828]
mpls-in-ip 137        MPLS-in-IP        # MPLS-in-IP [RFC4023]
manet        138                        # MANET Protocols [RFC5498]
hip        139        HIP                # Host Identity Protocol
shim6        140        Shim6                # Shim6 Protocol [RFC5533]
wesp        141        WESP                # Wrapped Encapsulating Security Payload
rohc        142        ROHC                # Robust Header Compression
```

5.4 with

Creates a context of execution for a block. Context block will have a object value that is only valid in the context. Context is initialized on entry, invalidated when context is over. Context can be over either when end of block is reached or there is an exception. In any case invalidation is done. This is useful in scenarios like: * File, network connection handlers. * Database connections * Concurrent code locks

In this way, automatic closing of files, committing/rolling back database connection depending on success or exception within the block, releasing locks when critical regions are exited, are achieved.

```
In [2]: with open("/etc/protocols","r") as fp:
            print(fp.readline(),end='')

# Internet (IP) protocols


In [3]: print(fp.readline())


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

        ValueError                                Traceback (most recent call last)

        <ipython-input-3-4d0b66eada2a> in <module>()
    ----> 1 print(fp.readline())


        ValueError: I/O operation on closed file.


In [6]: class F:
            def __init__(self,x):
                self.x = x
            def __str__(self):
                return str(self.x)
            def __enter__(self):
                print("entered:",self.x)
            def __exit__(self,extype,exval,traceback):
                print("exitted:",self.x,extype,exval,traceback)
                #raise StopIteration

In [7]: with F(10) as f:
            print("hello")
            a={}
            print(a['no key like this'])
            print("world")

entered: 10
hello
exitted: 10 <class 'KeyError'> 'no key like this' <traceback object at 0x7f7c84476448>
```

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

KeyError                                  Traceback (most recent call last)

<ipython-input-7-05fbbe918faf> in <module>()
  3     print("hello")
  4     a={}
----> 5     print(a['no key like this'])
  6     print("world")


KeyError: 'no key like this'
```

6 Simple string processing

```
In [35]: type("abc")
```

```
Out[35]: str
```

```
In [5]:     a="onur tolg sehitoglu"
        a.split(" ")
```

```
Out[5]: ['onur', 'tolg', 'sehitoglu']
```

```
In [40]: "".join(["a","b","c"])
```

```
Out[40]: 'abc'
```

```
In [6]: print(a)
        print(a.find("sehit"))
        print(a.find("nothing"))  # returns -1
        print(a.rindex("o"), a.index("o"))    # right to left search
        try:
            print(a.index("nothing"))
        except ValueError as v:
            print(v)
        print(a.upper())
        a[2:8]                  # substring
```

```
onur tolg sehitoglu
10
-1
15 0
substring not found
ONUR TOLG SEHITOGLU
```

```
Out[6]: 'ur tol'
```

7 Formatted strings

- old format: `%` operator, `format % (args)`
- new format: `formatstring.format(args)`

```
In [3]: "%d %-10d %5.2f %20s" % (3,4,1.5,"onur")

Out[3]: '3 4           1.50                 onur'

In [50]: print("{} {} {}".format(3,4,"onur"))
         print("{2} {1} {0}".format(3,4,"onur"))

3 4 onur
onur 4 3


In [5]: "{0:5d} {1:5.2f} {2:20s} {surname} {name}".format(3,4,"onur",
                                   name="cin",surname="ali")

Out[5]: '    3  4.00 onur                 ali cin'
```

8 Array map and filter

- `filter(f,iteratable)` returns an iterator giving only elements i returning True for f(i)
- `map(f, iteratable)` returns an iterator giving f(i) for all elements
- `[f(i) for i in iteratable]` also works similar to `map`
- `[f(i) for i in iteratable if g(i)]` is general form

```
In [54]: list(filter(lambda x:x<5, [1,2,3,4,5,6,7]))

Out[54]: [1, 2, 3, 4]

In [2]: list(map(lambda x:x*x, [1,2,3,4,5,6,7]))

Out[2]: [1, 4, 9, 16, 25, 36, 49]

In [4]: ":".join(map(str, [1,2,3,4,5,6,7]))

Out[4]: '1:2:3:4:5:6:7'

In [5]: ":".join(map(str, filter(lambda x: x % 2 == 0,[1,2,3,4,5,6,7])))

Out[5]: '2:4:6'

In [9]: a=[2,3,5,7,11,13,17]
        print([i*i for i in a])
        print([i*i for i in a if i < 5])
        print([i*j for i in a for j in a])

[4, 9, 25, 49, 121, 169, 289]
[4, 9]
[4, 6, 10, 14, 22, 26, 34, 6, 9, 15, 21, 33, 39, 51, 10, 15, 25, 35, 55, 65, 85, 14, 21, 35, 49,
```

9 File and I/O

- `fp = open(path,"rw")` returns a file handle
- `fp.readline()` read a line
- `fp.read(n)` read `n` bytes
- `fp.read()` whole read
- `fp.seek(n)` seek to a position in file
- `fp.close()` close file
- file handle is also an iterator, `next()` reads next line

```
In [6]: input1=input()
```

```
32132141
```

```
In [7]: input1

Out[7]: '32132141'

In [8]: fp=open("/etc/services","r")

In [9]: for line in fp:
            print(line,end='')
        fp.readline()
```

```
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux                1/tcp                             # TCP port service multiplexer
echo                7/tcp
echo                7/udp
discard               9/tcp               sink null
discard               9/udp               sink null
systat                11/tcp              users
daytime               13/tcp
daytime               13/udp
netstat               15/tcp
qotd                17/tcp              quote
msp                18/tcp                          # message send protocol
msp                18/udp
```

```
chargen                19/tcp                ttytst source
chargen                19/udp                ttytst source
ftp-data        20/tcp
ftp                21/tcp
fsp                21/udp                fspd
ssh                22/tcp                                # SSH Remote Login Protocol
telnet                23/tcp
smtp                25/tcp                mail
time                37/tcp                timserver
time                37/udp                timserver
rlp                39/udp                resource        # resource location
nameserver        42/tcp                name                # IEN 116
whois                43/tcp                nicname
tacacs                49/tcp                                # Login Host Protocol (TACACS)
tacacs                49/udp
re-mail-ck        50/tcp                                # Remote Mail Checking Protocol
re-mail-ck        50/udp
domain                53/tcp                                # Domain Name Server
domain                53/udp
tacacs-ds        65/tcp                                # TACACS-Database Service
tacacs-ds        65/udp
bootps                67/tcp                                # BOOTP server
bootps                67/udp
bootpc                68/tcp                                # BOOTP client
bootpc                68/udp
tftp                69/udp
gopher                70/tcp                                # Internet Gopher
finger                79/tcp
http                80/tcp                www                # WorldWideWeb HTTP
link                87/tcp                ttylink
kerberos        88/tcp                kerberos5 krb5 kerberos-sec        # Kerberos v5
kerberos        88/udp                kerberos5 krb5 kerberos-sec        # Kerberos v5
supdup                95/tcp
hostnames        101/tcp                hostname        # usually from sri-nic
iso-tsap        102/tcp                tsap                # part of ISODE
acr-nema        104/tcp                dicom                # Digital Imag. & Comm. 300
acr-nema        104/udp                dicom
csnet-ns        105/tcp                cso-ns                # also used by CSO name server
csnet-ns        105/udp                cso-ns
rtelnet                107/tcp                                # Remote Telnet
rtelnet                107/udp
pop3                110/tcp                pop-3                # POP version 3
sunrpc                111/tcp                portmapper        # RPC 4.0 portmapper
sunrpc                111/udp                portmapper
auth                113/tcp                authentication tap ident
sftp                115/tcp
nntp                119/tcp                readnews untp        # USENET News Transfer Protocol
ntp                123/tcp
ntp                123/udp                                # Network Time Protocol
pwdgen                129/tcp                                # PWDGEN service
pwdgen                129/udp
loc-srv                135/tcp                epmap                # Location Service
loc-srv                135/udp                epmap
netbios-ns        137/tcp                                # NETBIOS Name Service
netbios-ns        137/udp
netbios-dgm        138/tcp                                # NETBIOS Datagram Service
netbios-dgm        138/udp
netbios-ssn        139/tcp                                # NETBIOS session service
netbios-ssn        139/udp
imap2                143/tcp                imap                # Interim Mail Access P 2 and 4
snmp                161/tcp                                # Simple Net Mgmt Protocol
snmp                161/udp
snmp-trap        162/tcp                snmptrap        # Traps for SNMP
snmp-trap        162/udp                snmptrap
cmip-man        163/tcp                                # ISO mgmt over IP (CMOT)
cmip-man        163/udp
cmip-agent        164/tcp
cmip-agent        164/udp
mailq                174/tcp                        # Mailer transport queue for Zmailer
mailq                174/udp
xdmcp                177/tcp                                # X Display Mgr. Control Proto
xdmcp                177/udp
nextstep        178/tcp                NeXTStep NextStep        # NeXTStep window
nextstep        178/udp                NeXTStep NextStep        #  server
bgp                179/tcp                                # Border Gateway Protocol
irc                194/tcp                                # Internet Relay Chat
irc                194/udp
smux                199/tcp                                # SNMP Unix Multiplexer
smux                199/udp
at-rtmp                201/tcp                                # AppleTalk routing
at-rtmp                201/udp
at-nbp                202/tcp                                # AppleTalk name binding
at-nbp                202/udp
at-echo                204/tcp                                # AppleTalk echo
at-echo                204/udp
at-zis                206/tcp                                # AppleTalk zone information
at-zis                206/udp
qmtp                209/tcp                                # Quick Mail Transfer Protocol
qmtp                209/udp
z3950                210/tcp                wais                # NISO Z39.50 database
z3950                210/udp                wais
ipx                213/tcp                                # IPX
ipx                213/udp
pawserv                345/tcp                                # Perf Analysis Workbench
pawserv                345/udp
zserv                346/tcp                                # Zebra server
zserv                346/udp
fatserv                347/tcp                                  # Fatmen Server
fatserv                347/udp
rpc2portmap        369/tcp
rpc2portmap        369/udp                                  # Coda portmapper
codaauth2        370/tcp
codaauth2        370/udp                                  # Coda authentication server
clearcase        371/tcp                Clearcase
clearcase        371/udp                Clearcase
ulistserv        372/tcp                                  # UNIX Listserv
ulistserv        372/udp
ldap                389/tcp                         # Lightweight Directory Access Protocol
ldap                389/udp
imsp                406/tcp                         # Interactive Mail Support Protocol
imsp                406/udp
svrloc                427/tcp                                 # Server Location
svrloc                427/udp
https                443/tcp                                 # http protocol over TLS/SSL
snpp                444/tcp                                 # Simple Network Paging Protocol
snpp                444/udp
microsoft-ds        445/tcp                                 # Microsoft Naked CIFS
microsoft-ds        445/udp
kpasswd                464/tcp
kpasswd                464/udp
urd                465/tcp                  ssmtp smtps  # URL Rendesvous Directory for SSM
saft                487/tcp                         # Simple Asynchronous File Transfer
saft                487/udp
isakmp                500/tcp                          # IPsec - Internet Security Association
isakmp                500/udp                          #  and Key Management Protocol
rtsp                554/tcp                         # Real Time Stream Control Protocol
rtsp                554/udp
nqs                607/tcp                                  # Network Queuing system
nqs                607/udp
npmp-local        610/tcp                dqs313_qmaster                # npmp-local / DQS
npmp-local        610/udp                dqs313_qmaster
npmp-gui        611/tcp                dqs313_execd                # npmp-gui / DQS
npmp-gui        611/udp                dqs313_execd
hmmp-ind        612/tcp                dqs313_intercell        # HMMP Indication / DQS
hmmp-ind        612/udp                dqs313_intercell
asf-rmcp        623/udp                # ASF Remote Management and Control Protocol
qmqp                628/tcp
qmqp                628/udp
ipp                631/tcp                                  # Internet Printing Protocol
ipp                631/udp
#
# UNIX specific services
#
exec                512/tcp
```

```
biff                512/udp                comsat
login                513/tcp
who                513/udp                whod
shell                513/tcp                cmd                # no passwords used
syslog                514/udp
printer                515/tcp                spooler                # line printer spooler
talk                517/udp
ntalk                518/udp
route                520/udp                router routed        # RIP
timed                525/udp                timeserver
tempo                526/tcp                newdate
courier                530/tcp                rpc
conference        531/tcp                chat
netnews                532/tcp                readnews
netwall                533/udp                                # for emergency broadcasts
gdomap                538/tcp                                # GNUstep distributed objects
gdomap                538/udp
uucp                540/tcp                uucpd                # uucp daemon
klogin                543/tcp                                # Kerberized `rlogin' (v5)
kshell                544/tcp                krcmd                # Kerberized `rsh' (v5)
dhcpv6-client        546/tcp
dhcpv6-client        546/udp
dhcpv6-server        547/tcp
dhcpv6-server        547/udp
afpovertcp        548/tcp                                # AFP over TCP
afpovertcp        548/udp
idfp                549/tcp
idfp                549/udp
remotefs        556/tcp                rfs_server rfs        # Brunhoff remote filesystem
nntps                563/tcp                snntp                # NNTP over SSL
submission        587/tcp                                # Submission [RFC4409]
ldaps                636/tcp                                # LDAP over SSL
ldaps                636/udp
tinc                655/tcp                                # tinc control port
tinc                655/udp
silc                706/tcp
silc                706/udp
kerberos-adm        749/tcp                                # Kerberos `kadmin' (v5)
#
webster                765/tcp                                # Network dictionary
webster                765/udp
rsync                873/tcp
ftps-data        989/tcp                                # FTP over SSL (data)
ftps                990/tcp
telnets                992/tcp                                # Telnet over SSL
imaps                993/tcp                                # IMAP over SSL
pop3s                995/tcp                                # POP-3 over SSL
#
```

```
# From ``Assigned Numbers'':
#
#> The Registered Ports are not controlled by the IANA and on most systems
#> can be used by ordinary user processes or programs executed by ordinary
#> users.
#
#> Ports are used in the TCP [45,106] to name the ends of logical
#> connections which carry long term conversations.  For the purpose of
#> providing services to unknown callers, a service contact port is
#> defined.  This list specifies the port used by the server process as its
#> contact port.  While the IANA can not control uses of these ports it
#> does register or list uses of these ports as a convienence to the
#> community.
#
socks                1080/tcp                        # socks proxy server
socks                1080/udp
proofd                1093/tcp
proofd                1093/udp
rootd                1094/tcp
rootd                1094/udp
openvpn               1194/tcp
openvpn               1194/udp
rmiregistry        1099/tcp                        # Java RMI Registry
rmiregistry        1099/udp
kazaa                1214/tcp
kazaa                1214/udp
nessus                1241/tcp                        # Nessus vulnerability
nessus                1241/udp                        #  assessment scanner
lotusnote        1352/tcp        lotusnotes        # Lotus Note
lotusnote        1352/udp        lotusnotes
ms-sql-s        1433/tcp                        # Microsoft SQL Server
ms-sql-s        1433/udp
ms-sql-m        1434/tcp                        # Microsoft SQL Monitor
ms-sql-m        1434/udp
ingreslock        1524/tcp
ingreslock        1524/udp
datametrics        1645/tcp        old-radius
datametrics        1645/udp        old-radius
sa-msg-port        1646/tcp        old-radacct
sa-msg-port        1646/udp        old-radacct
kermit                1649/tcp
kermit                1649/udp
groupwise        1677/tcp
groupwise        1677/udp
l2f                1701/tcp        l2tp
l2f                1701/udp        l2tp
radius                1812/tcp
radius                1812/udp
```

```
radius-acct        1813/tcp        radacct                # Radius Accounting
radius-acct        1813/udp        radacct
msnp                1863/tcp                        # MSN Messenger
msnp                1863/udp
unix-status        1957/tcp                        # remstats unix-status server
log-server         1958/tcp                        # remstats log server
remoteping         1959/tcp                        # remstats remoteping server
cisco-sccp         2000/tcp                        # Cisco SCCP
cisco-sccp         2000/udp
search                 2010/tcp        ndtp
pipe-server        2010/tcp        pipe_server
nfs                2049/tcp                        # Network File System
nfs                2049/udp                        # Network File System
gnunet                 2086/tcp
gnunet                 2086/udp
rtcm-sc104         2101/tcp                        # RTCM SC-104 IANA 1/29/99
rtcm-sc104         2101/udp
gsigatekeeper         2119/tcp
gsigatekeeper         2119/udp
gris                2135/tcp               # Grid Resource Information Server
gris                2135/udp
cvspserver         2401/tcp                        # CVS client/server operations
cvspserver         2401/udp
venus                 2430/tcp                        # codacon port
venus                 2430/udp                        # Venus callback/wbc interface
venus-se        2431/tcp                        # tcp side effects
venus-se        2431/udp                        # udp sftp side effect
codasrv                 2432/tcp                        # not used
codasrv                 2432/udp                        # server port
codasrv-se         2433/tcp                        # tcp side effects
codasrv-se         2433/udp                        # udp sftp side effect
mon                2583/tcp                        # MON traps
mon                2583/udp
dict                2628/tcp                        # Dictionary server
dict                2628/udp
f5-globalsite         2792/tcp
f5-globalsite         2792/udp
gsiftp                 2811/tcp
gsiftp                 2811/udp
gpsd                2947/tcp
gpsd                2947/udp
gds-db                 3050/tcp        gds_db                # InterBase server
gds-db                 3050/udp        gds_db
icpv2                 3130/tcp        icp                # Internet Cache Protocol
icpv2                 3130/udp        icp
isns                3205/tcp                        # iSNS Server Port
isns                3205/udp                        # iSNS Server Port
iscsi-target        3260/tcp
```

```
mysql                3306/tcp
mysql                3306/udp
nut                3493/tcp                       # Network UPS Tools
nut                3493/udp
distcc                3632/tcp                       # distributed compiler
distcc                3632/udp
daap                3689/tcp                        # Digital Audio Access Protocol
daap                3689/udp
svn                3690/tcp        subversion        # Subversion protocol
svn                3690/udp        subversion
suucp                4031/tcp                        # UUCP over SSL
suucp                4031/udp
sysrqd                4094/tcp                        # sysrq daemon
sysrqd                4094/udp
sieve                4190/tcp                        # ManageSieve Protocol
epmd                4369/tcp                        # Erlang Port Mapper Daemon
epmd                4369/udp
remctl                4373/tcp                # Remote Authenticated Command Service
remctl                4373/udp
f5-iquery        4353/tcp                        # F5 iQuery
f5-iquery        4353/udp
ipsec-nat-t        4500/udp                        # IPsec NAT-Traversal [RFC3947]
iax                4569/tcp                        # Inter-Asterisk eXchange
iax                4569/udp
mtn                4691/tcp                        # monotone Netsync Protocol
mtn                4691/udp
radmin-port        4899/tcp                        # RAdmin Port
radmin-port        4899/udp
rfe                5002/udp                        # Radio Free Ethernet
rfe                5002/tcp
mmcc                5050/tcp        # multimedia conference control tool (Yahoo IM)
mmcc                5050/udp
sip                5060/tcp                        # Session Initiation Protocol
sip                5060/udp
sip-tls                5061/tcp
sip-tls                5061/udp
aol                5190/tcp                        # AIM
aol                5190/udp
xmpp-client        5222/tcp        jabber-client        # Jabber Client Connection
xmpp-client        5222/udp        jabber-client
xmpp-server        5269/tcp        jabber-server        # Jabber Server Connection
xmpp-server        5269/udp        jabber-server
cfengine        5308/tcp
cfengine        5308/udp
mdns                5353/tcp                        # Multicast DNS
mdns                5353/udp
postgresql        5432/tcp        postgres        # PostgreSQL Database
postgresql        5432/udp        postgres
```

```
freeciv                 5556/tcp        rptp                # Freeciv gameplay
freeciv                 5556/udp
amqps                   5671/tcp                            # AMQP protocol over TLS/SSL
amqp                    5672/tcp
amqp                    5672/udp
amqp                    5672/sctp
ggz                     5688/tcp                            # GGZ Gaming Zone
ggz                     5688/udp
x11                     6000/tcp        x11-0               # X Window System
x11                     6000/udp        x11-0
x11-1                   6001/tcp
x11-1                   6001/udp
x11-2                   6002/tcp
x11-2                   6002/udp
x11-3                   6003/tcp
x11-3                   6003/udp
x11-4                   6004/tcp
x11-4                   6004/udp
x11-5                   6005/tcp
x11-5                   6005/udp
x11-6                   6006/tcp
x11-6                   6006/udp
x11-7                   6007/tcp
x11-7                   6007/udp
gnutella-svc            6346/tcp                            # gnutella
gnutella-svc            6346/udp
gnutella-rtr            6347/tcp                            # gnutella
gnutella-rtr            6347/udp
sge-qmaster             6444/tcp        sge_qmaster         # Grid Engine Qmaster Service
sge-qmaster             6444/udp        sge_qmaster
sge-execd               6445/tcp        sge_execd           # Grid Engine Execution Service
sge-execd               6445/udp        sge_execd
mysql-proxy             6446/tcp                            # MySQL Proxy
mysql-proxy             6446/udp
babel                   6696/udp                            # Babel Routing Protocol
ircs-u                  6697/tcp                            # Internet Relay Chat via TLS/SSL
afs3-fileserver 7000/tcp        bbs                 # file server itself
afs3-fileserver 7000/udp        bbs
afs3-callback           7001/tcp                            # callbacks to cache managers
afs3-callback           7001/udp
afs3-prserver           7002/tcp                            # users & groups database
afs3-prserver           7002/udp
afs3-vlserver           7003/tcp                            # volume location database
afs3-vlserver           7003/udp
afs3-kaserver           7004/tcp                            # AFS/Kerberos authentication
afs3-kaserver           7004/udp
afs3-volser             7005/tcp                            # volume managment server
afs3-volser             7005/udp
```

```
afs3-errors        7006/tcp                        # error interpretation service
afs3-errors        7006/udp
afs3-bos        7007/tcp                        # basic overseer process
afs3-bos        7007/udp
afs3-update        7008/tcp                        # server-to-server updater
afs3-update        7008/udp
afs3-rmtsys        7009/tcp                        # remote cache manager service
afs3-rmtsys        7009/udp
font-service        7100/tcp        xfs                # X Font Service
font-service        7100/udp        xfs
http-alt        8080/tcp        webcache        # WWW caching service
http-alt        8080/udp
puppet                8140/tcp                        # The Puppet master service
bacula-dir        9101/tcp                        # Bacula Director
bacula-dir        9101/udp
bacula-fd        9102/tcp                        # Bacula File Daemon
bacula-fd        9102/udp
bacula-sd        9103/tcp                        # Bacula Storage Daemon
bacula-sd        9103/udp
xmms2                9667/tcp        # Cross-platform Music Multiplexing System
xmms2                9667/udp
nbd                10809/tcp                        # Linux Network Block Device
zabbix-agent        10050/tcp                        # Zabbix Agent
zabbix-agent        10050/udp
zabbix-trapper        10051/tcp                        # Zabbix Trapper
zabbix-trapper        10051/udp
amanda                10080/tcp                        # amanda backup services
amanda                10080/udp
dicom                11112/tcp
hkp                11371/tcp                        # OpenPGP HTTP Keyserver
hkp                11371/udp
bprd                13720/tcp                        # VERITAS NetBackup
bprd                13720/udp
bpdbm                13721/tcp                        # VERITAS NetBackup
bpdbm                13721/udp
bpjava-msvc        13722/tcp                        # BP Java MSVC Protocol
bpjava-msvc        13722/udp
vnetd                13724/tcp                        # Veritas Network Utility
vnetd                13724/udp
bpcd                13782/tcp                        # VERITAS NetBackup
bpcd                13782/udp
vopied                13783/tcp                        # VERITAS NetBackup
vopied                13783/udp
db-lsp                17500/tcp                        # Dropbox LanSync Protocol
dcap                22125/tcp                        # dCache Access Protocol
gsidcap                22128/tcp                        # GSI dCache Access Protocol
wnn6                22273/tcp                        # wnn6
wnn6                22273/udp
```

```
#
# Datagram Delivery Protocol services
#
rtmp             1/ddp                        # Routing Table Maintenance Protocol
nbp             2/ddp                        # Name Binding Protocol
echo             4/ddp                        # AppleTalk Echo Protocol
zip             6/ddp                        # Zone Information Protocol

#=========================================================================
# The remaining port numbers are not as allocated by IANA.
#=========================================================================

# Kerberos (Project Athena/MIT) services
# Note that these are for Kerberos v4, and are unofficial.  Sites running
# v4 should uncomment these and comment out the v5 entries above.
#
kerberos4       750/udp                kerberos-iv kdc        # Kerberos (server)
kerberos4       750/tcp                kerberos-iv kdc
kerberos-master       751/udp                kerberos_master        # Kerberos authentication
kerberos-master       751/tcp
passwd-server       752/udp                passwd_server        # Kerberos passwd server
krb-prop       754/tcp                krb_prop krb5_prop hprop # Kerberos slave propagation
krbupdate       760/tcp                kreg                # Kerberos registration
swat             901/tcp                            # swat
kpop             1109/tcp                        # Pop with Kerberos
knetd             2053/tcp                        # Kerberos de-multiplexor
zephyr-srv       2102/udp                        # Zephyr server
zephyr-clt       2103/udp                        # Zephyr serv-hm connection
zephyr-hm       2104/udp                        # Zephyr hostmanager
eklogin             2105/tcp                        # Kerberos encrypted rlogin
# Hmmm. Are we using Kv4 or Kv5 now? Worrying.
# The following is probably Kerberos v5  --- ajt@debian.org (11/02/2000)
kx             2111/tcp                        # X over Kerberos
iprop             2121/tcp                        # incremental propagation
#
# Unofficial but necessary (for NetBSD) services
#
supfilesrv       871/tcp                                # SUP server
supfiledbg       1127/tcp                        # SUP debugging

#
# Services added for the Debian GNU/Linux distribution
#
linuxconf       98/tcp                                # LinuxConf
poppassd       106/tcp                                # Eudora
poppassd       106/udp
moira-db       775/tcp                moira_db        # Moira database
```

```
moira-update        777/tcp                moira_update        # Moira update protocol
moira-ureg        779/udp                moira_ureg        # Moira user registration
spamd                783/tcp                                   # spamassassin daemon
omirr                808/tcp               omirrd                # online mirror
omirr                808/udp               omirrd
customs                1001/tcp                                # pmake customs server
customs                1001/udp
skkserv                1178/tcp                                # skk jisho server port
predict                1210/udp                                # predict -- satellite tracking
rmtcfg                1236/tcp                                # Gracilis Packeten remote config server
wipld                1300/tcp                                # Wipl network monitor
xtel                1313/tcp                                # french minitel
xtelw                1314/tcp                                # french minitel
support                1529/tcp                                # GNATS
cfinger                2003/tcp                                # GNU Finger
frox                2121/tcp                                # frox: caching ftp proxy
ninstall        2150/tcp                        # ninstall service
ninstall        2150/udp
zebrasrv        2600/tcp                        # zebra service
zebra                2601/tcp                                # zebra vty
ripd                2602/tcp                                # ripd vty (zebra)
ripngd                2603/tcp                                # ripngd vty (zebra)
ospfd                2604/tcp                                # ospfd vty (zebra)
bgpd                2605/tcp                                # bgpd vty (zebra)
ospf6d                2606/tcp                                # ospf6d vty (zebra)
ospfapi                2607/tcp                                # OSPF-API
isisd                2608/tcp                                # ISISd vty (zebra)
afbackup        2988/tcp                        # Afbackup system
afbackup        2988/udp
afmbackup        2989/tcp                        # Afmbackup system
afmbackup        2989/udp
xtell                4224/tcp                                # xtell server
fax                4557/tcp                                # FAX transmission service (old)
hylafax                4559/tcp                                # HylaFAX client-server protocol (new)
distmp3                4600/tcp                                # distmp3host daemon
munin                4949/tcp        lrrd                    # Munin
enbd-cstatd        5051/tcp                                # ENBD client statd
enbd-sstatd        5052/tcp                                # ENBD server statd
pcrd                5151/tcp                                # PCR-1000 Daemon
noclog                5354/tcp                                # noclogd with TCP (nocol)
noclog                5354/udp                                # noclogd with UDP (nocol)
hostmon                5355/tcp                                # hostmon uses TCP (nocol)
hostmon                5355/udp                                # hostmon uses UDP (nocol)
rplay                5555/udp                                # RPlay audio service
nrpe                5666/tcp                                # Nagios Remote Plugin Executor
nsca                5667/tcp                                # Nagios Agent - NSCA
mrtd                5674/tcp                                # MRT Routing Daemon
bgpsim                5675/tcp                                # MRT Routing Simulator
```

```
canna                   5680/tcp                        # cannaserver
syslog-tls          6514/tcp                        # Syslog over TLS [RFC5425]
sane-port           6566/tcp        sane saned          # SANE network scanner daemon
ircd                    6667/tcp                        # Internet Relay Chat
zope-ftp            8021/tcp                        # zope management by ftp
tproxy                  8081/tcp                        # Transparent Proxy
omniorb                 8088/tcp                        # OmniORB
omniorb                 8088/udp
clc-build-daemon 8990/tcp                       # Common lisp build daemon
xinetd                  9098/tcp
mandelspawn         9359/udp        mandelbrot          # network mandelbrot
git                 9418/tcp                        # Git Version Control System
zope                    9673/tcp                        # zope server
webmin                  10000/tcp
kamanda                 10081/tcp                       # amanda backup services (Kerberos)
kamanda                 10081/udp
amandaidx           10082/tcp                       # amanda backup services
amidxtape           10083/tcp                       # amanda backup services
smsqp                   11201/tcp                       # Alamin SMS gateway
smsqp                   11201/udp
xpilot                  15345/tcp                       # XPilot Contact Port
xpilot                  15345/udp
sgi-cmsd            17001/udp               # Cluster membership services daemon
sgi-crsd            17002/udp
sgi-gcd                 17003/udp                       # SGI Group membership daemon
sgi-cad                 17004/tcp                       # Cluster Admin daemon
isdnlog                 20011/tcp                       # isdn logging system
isdnlog                 20011/udp
vboxd                   20012/tcp                       # voice box system
vboxd                   20012/udp
binkp                   24554/tcp                       # binkp fidonet protocol
asp                 27374/tcp                       # Address Search Protocol
asp                 27374/udp
csync2                  30865/tcp                       # cluster synchronization tool
dircproxy           57000/tcp                       # Detachable IRC Proxy
tfido                   60177/tcp                       # fidonet EMSI over telnet
fido                    60179/tcp                       # fidonet EMSI over TCP

# Local services
```

```
Out[9]: ''
```

```
In [14]: a=fp.read(100)
         print(a)
         type(a)
```

```
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for bot
```

```
Out[14]: str

In [15]: #fp.readlines()

In [16]: wholefile=fp.read()   # read the whole rest of the file

In [17]: fp.seek(10)
         fp.read(10)

Out[17]: 'services, '

In [18]: fp.close()
```

10 OS and environment

- `sys.argv` list of command line arguments.
- `sys.path` search path for modules
- `sys.exit(0)` exit program
- `os.stat("path")` get system information about file (size, protection, owner, type, timestamps)
- `os.scandir("dirpath")` read content of a file, returns an iterator
- `os.fwalk("dirpath")` recursive content of a file, returns iterator to 4-tuples (name, dirlist, filelist, id)
- `os.system("command line")` execute a program

```
In [1]: import sys
        print(sys.argv)

['/usr/lib/python3/dist-packages/ipykernel/__main__.py', '--debug', '-f', '/home/onur/.local/sha


In [2]: print(sys.path)

['', '/usr/lib/python35.zip', '/usr/lib/python3.5', '/usr/lib/python3.5/plat-x86_64-linux-gnu',


In [3]: sys.path.append('/home/mymodules/mylibrary')
        print(sys.path)

['', '/usr/lib/python35.zip', '/usr/lib/python3.5', '/usr/lib/python3.5/plat-x86_64-linux-gnu',


In [4]: print(sys.copyright)
        print(sys.version, sys.version_info)
```

```
Copyright (c) 2001-2017 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170118] sys.version_info(major=3, minor=5, micro=3, releaselevel='final', serial=0)
```

```
In [5]: #sys.exit(1)
```

```
In [6]: import os
        import time

        os.chdir("/tmp")
        fst=os.stat("/etc/services")
        print(fst)
        print(fst.st_uid, time.asctime(time.localtime(fst.st_mtime)))
```

```
os.stat_result(st_mode=33188, st_ino=5983366, st_dev=2081, st_nlink=1, st_uid=0, st_gid=0, st_si
0 Mon Dec 26 04:56:39 2016
```

```
In [66]: dircontent=os.scandir("/etc/apt")
         for i in dircontent:
             print(i.name, i.is_dir(), i.stat(), i.path)
```

```
listchanges.conf False os.stat_result(st_mode=33188, st_ino=5981767, st_dev=2081, st_nlink=1, st
preferences.d True os.stat_result(st_mode=16877, st_ino=5980418, st_dev=2081, st_nlink=2, st_uid
sources.list~ False os.stat_result(st_mode=33188, st_ino=5980603, st_dev=2081, st_nlink=1, st_ui
sources.list False os.stat_result(st_mode=33188, st_ino=5981266, st_dev=2081, st_nlink=1, st_uid
apt.conf.d True os.stat_result(st_mode=16877, st_ino=5980197, st_dev=2081, st_nlink=2, st_uid=0,
trusted.gpg.d True os.stat_result(st_mode=16877, st_ino=5980412, st_dev=2081, st_nlink=2, st_uid
trusted.gpg False os.stat_result(st_mode=33188, st_ino=5983168, st_dev=2081, st_nlink=1, st_uid=
trusted.gpg~ False os.stat_result(st_mode=33188, st_ino=5983154, st_dev=2081, st_nlink=1, st_uid
sources.list.d True os.stat_result(st_mode=16877, st_ino=5980419, st_dev=2081, st_nlink=2, st_ui
```

```
In [67]: for v in os.fwalk("/etc/apt"):
             print(v)
```

```
('/etc/apt', ['preferences.d', 'apt.conf.d', 'trusted.gpg.d', 'sources.list.d'], ['listchanges.c
('/etc/apt/preferences.d', [], [], 43)
```

```
('/etc/apt/apt.conf.d', [], ['70debconf', '00CDMountPoint', '01autoremove-kernels', '50appstream
('/etc/apt/trusted.gpg.d', [], ['debian-archive-wheezy-stable.gpg~', 'debian-archive-wheezy-auto
('/etc/apt/sources.list.d', [], ['google-talkplugin.list', 'google-chrome.list', 'deb-multimedia
```

```
In [ ]: os.system("ls -l /") # don't use. use subprocess instead
```

10.1 Time utilities

- `time.time()` Unix time as a floating point value (seconds since 1st Jan 1970)
- `time.localtime()` Time in structured form
- `time.asctime()` Time as a string in current system config.
- `time.strftime(format,timestruct)` Time in user defined format string
- `time.strptime(inputstring, format)` String to structured time

```
In [8]: import time
        now=time.time()
        print(now)
        now2=time.localtime(now)
        print("year {}, month {}, day {}, hour: {} minutes: {}".format(
        now2.tm_year, now2.tm_mon, now2.tm_mday, now2.tm_hour, now2.tm_min))
        time.strftime("%Y/%m/%d %H:%M:%S %s",now2)
```

```
1540200417.5185628
year 2018, month 10, day 22, hour: 12 minutes: 26
```

```
Out[8]: '2018/10/22 12:26:57 1540200417'
```

```
In [15]: bt=time.strptime("1999/12/30","%Y/%m/%d")
```

```
In [11]: time.asctime()
```

```
Out[11]: 'Mon Oct 22 12:29:16 2018'
```

11 Regular expressions

A regular expression matches

- `.` matches any character
- `[c1-c2]` matches all characters in range c1-c2
- `[aeuio]` matches all characters enclosed `[a-z.:-]`
- `[^a-z]` matches anything but a to z
- `exp?` matches 0 or 1 occurences of regex exp `[a-z]?`
- `exp*` matches 0 or more occurences of regex exp
- `exp+` matches 1 or more occurences of regex exp
- non greedy `??` `*?` `+?` Instead of longest, they match smallest
- `exp{m,n}` `exp{m,}` `exp{,n}` m to n occurences of regex
- ^ matches start of the string as position

- $ matches end of the string

```
In [2]: import re
        #print(re.search("[a-z]+","Onur Sehitoglu"))
        #print(re.search("[a-z]+[a-z]+$","Onur Sehitoglu"))
        print(re.match("[a-z]+","Onur Sehitoglu"))
        print(re.match("([a-z]+)([a-z]+$)","onursehitoglu").groups())
        print(re.match("([a-z]+?)([a-z]+)$","onursehitoglu").groups())
        print(re.match("([a-z]{1,5}?)([a-z]{1,8}?)$","onursehitoglu").groups())
```

```
None
('onursehitogl', 'u')
('o', 'nursehitoglu')
('onurs', 'ehitoglu')
```

```
In [21]: print(re.search("^[0-9.]+$", "123.123123123.123412"))
         print(re.search("^[0[0-9]*$", ".231"))
```

```
<_sre.SRE_Match object; span=(0, 20), match='123.123123123.123412'>
None
```

11.1 Grouping

- `(exp)` Group the expressions.
- `(exp)? (exp)* (exp)+ (exp){m,n}`
- `e1|e2` either e1 or e2

```
In [8]: print(re.search("^[0-9]+(\.[0-9]*)?$", "21312.123"))
        print(re.search("^([A-Z]+|[a-z]+)$", "oNur"))
        sname=re.search("^(([A-Z]+|[a-z]+) ?)+$", "onur tolga SEHITOGLU ")
        print(sname)
```

```
<_sre.SRE_Match object; span=(0, 9), match='21312.123'>
None
<_sre.SRE_Match object; span=(0, 21), match='onur tolga SEHITOGLU '>
```

```
In [9]: print(sname.start(), sname.end(), sname.span(), sname.group(), sname.groups())
```

```
0 21 (0, 21) onur tolga SEHITOGLU  ('SEHITOGLU ', 'SEHITOGLU')
```

```
In [20]: match=re.search("([a-z]+) ([a-z]+) ([a-z]+)", "+|+onur tolga sehitoglu><bla bla")

         print(match.group())
         print(match.groups())
```

```
onur tolga sehitoglu
('onur', 'tolga', 'sehitoglu')
```

```
In [21]: match=re.search("(?P<name>[a-z]+) ([a-z]+) (?P<sname>[a-z]+)", "onur tolga sehitoglu")
```

```
In [32]: match.groupdict()
```

```
Out[32]: {'name': 'onur', 'sname': 'sehitoglu'}
```

```
In [3]: match=re.search("(?P<word>[a-z]+) (?P=word)","onur tolga")
        print(match)
        match=re.search("(?P<word>[a-z]+) (?P=word)","onur onur")
        print(match)
```

```
None
<_sre.SRE_Match object; span=(0, 9), match='onur onur'>
```

```
In [33]: print(re.search("^([a-z]+|[A-Z]+) \\1$", "onur onur"))
         print(re.search("^([a-z]+|[A-Z]+) ([a-z]+) \\1$", "onur tolga onur"))
         print(re.search("^([a-z]+|[A-Z]+) ([a-z]+) \\2$", "onur tolga tolga"))
```

```
<_sre.SRE_Match object; span=(0, 9), match='onur onur'>
<_sre.SRE_Match object; span=(0, 15), match='onur tolga onur'>
<_sre.SRE_Match object; span=(0, 16), match='onur tolga tolga'>
```