

Python

Input and Output



Copyright © Software Carpentry 2010

This work is licensed under the Creative Commons Attribution License See http://software-carpentry.org/license.html for more information.





Been using print to see what programs are doing How to save data to files?



How to save data to files?

And read data from them?



How to save data to files?

And read data from them?

Python's solution looks very much like C's

Python



How to save data to files?

And read data from them?

Python's solution looks very much like C's

A file is a sequence of bytes

How to save data to files?

And read data from them?

Python's solution looks very much like C's

- A file is a sequence of bytes
- But it's often more useful to treat it as a sequence of lines



Sample data file: "haiku.txt"

Three things are certain: Death, taxes, and lost data. Guess which has occurred.

Errors have occurred. We won't tell you where or why. Lazy programmers.

With searching comes loss and the presence of absence: "My Thesis" not found.

A crash reduces your expensive computer to a simple stone.







bytes - Assume 1-to-1 for now



bytes - Assume 1-to-1 for now

Revisit later



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```



```
reader = [open]' haiku. txt', 'r')
data = reader. read()
reader. close()
print len(data)
```

Create a file object



```
reader = open( haiku. txt', 'r')
data = reader. read()
reader. close()
print len(data)
```

File to connect to



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)

To read
```



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Now holds file object



```
reader = open('haiku. txt', 'r')
data = reader. read()
reader. close()
print len(data)
```

Read entire content of file into a string



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Now has a copy of all the bytes that were in the file



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Disconnect from the file



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Disconnect from the file
Not strictly necessary
in small programs, but
good practice



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Report how many characters were read



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
```

Report how many

characters were read

bytes



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
print len(data)
293
```





```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader.read(64)
print len(data)
reader.close()
```



```
reader = open('haiku. txt', 'r')
data = reader. read(64)
while data != '':

print len(data)
data = reader. read(64)
print len(data)
reader. close()
```



```
reader = open('haiku.txt', 'r')
data = reader. read(64)
while data != '':
    print len(data)
    data = reader. read(64)
print len(data)
                                  if there is no more data
reader.close()
```

Read (at most) 64 bytes Or the empty string



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != ''
    print len(data)
    data = reader.read(64)
    the last reader.close()
```

Keep looping as long as the last read returned



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader.read(64)

print len(data)
reader.close()
Do something with
the data
```



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader.read(64)
print len(data)
reader.close()
(Try to) reload
```



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader.read(64)

print len(data)
reader.close()
```

Should be 0 (or the loop would still be running)



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader. read(64)
print len(data)
reader.close()
64
64
64
64
37
```



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader. read(64)
print len(data)
reader.close()
64
                       Don't do this unless
64
64
64
37
```



```
reader = open('haiku.txt', 'r')
data = reader.read(64)
while data != '':
    print len(data)
    data = reader. read(64)
print len(data)
reader.close()
64
                       Don't do this unless the file really
64
                       might be very large (or infinite)
64
64
37
```



More common to read one line at a time



```
reader = open('haiku.txt', 'r')
line = reader.readline()
total = 0
count = 0
while line != '':
    count += 1
    total += len(line)
    line = reader.readline()
reader.close()
print 'average', float(total) / float(count)
```



```
reader = open('haiku.txt', 'r')
line = reader.readline()
total = 0
count = 0
while line != '':
    count += 1
    total += len(line)
    line = reader.readline()
reader.close()
print 'average', float(total) / float(count)
```





```
reader = open('haiku.txt', 'r')
line = reader.readline()
total = 0
count = 0
while line != '':
    count += 1
    total += len(line)
    line = reader.readline()
    reader.close()
print 'average', float(total) / float(count)
```



```
reader = open('haiku.txt', 'r')
line = reader.readline()
total = 0
count = 0
while line != '':
    count += 1
    total += len(line)
    line = reader.readline()
reader.close()
print 'Average', float(total) / float(count)
Average 19. 53333333
```





```
reader = open('haiku.txt', 'r')
contents = reader.readlines()
reader.close()
total = 0
count = 0
for line in contents:
    count += 1
    total += len(line)
print 'average', float(total) / float(count)
```



```
reader = open('haiku.txt', 'r')
contents = reader.readlines()
reader.close()

total = 0
count = 0

for line in contents:
    count += 1
    total += len(line)
print 'average', float(total) / float(count)
All lines in file
as list of strings
for line in contents:
    count += 1
    total += len(line)
print 'average', float(total) / float(count)
```



```
reader = open('haiku.txt', 'r')
contents = reader.readlines()
reader.close()
total = 0
count = 0

for line in contents:
    count += 1
    total += len(line)
print 'average', float(total) / float(count)
```



```
reader = open('haiku.txt', 'r')
contents = reader.readlines()
reader.close()
total = 0
count = 0
for line in contents:
    count += 1
    total += len(line)
print 'Average', float(total) / float(count)
Average 19.533333333
```



"Read lines as list" + "loop over list" is common idiom





```
reader = open('haiku.txt', 'r')
total = 0
count = 0
for line in reader:
    count += 1
    total += len(line)
reader.close()
print 'average', float(total) / float(count)
```





```
reader = open('haiku.txt', 'r')
total = 0
count = 0
for line in reader:
    count += 1
    total += len(line)
reader.close()
print 'average', float(total) / float(count)
19.533333333
```

Python





```
writer = open('temp.txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```



```
writer = open('temp. txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

Same function





```
writer = open('temp. txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

File to write to

Created if it doesn't exist



```
writer = open('temp.txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

For writing instead of reading



```
writer = open('temp.txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

Write a single string



```
writer = open('temp. txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

Write each string in a list



```
writer = open('temp.txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

elementsHeNeArKr



```
writer = open('temp.txt', 'w')
writer.write('elements')
writer.writelines(['He', 'Ne', 'Ar', 'Kr'])
writer.close()
```

elementsHeNeArKr

Python only writes what you tell it to



```
writer = open('temp.txt', 'w')
writer.write('elements\n')
writer.writelines(['He\n', 'Ne\n', 'Ar\n', 'Kr\n'])
writer.close()
```

Have to provide end-of-line characters yourself



```
writer = open('temp.txt', 'w')
writer.write('elements\n')
writer.writelines(['He\n', 'Ne\n', 'Ar\n', 'Kr\n'])
writer.close()
```

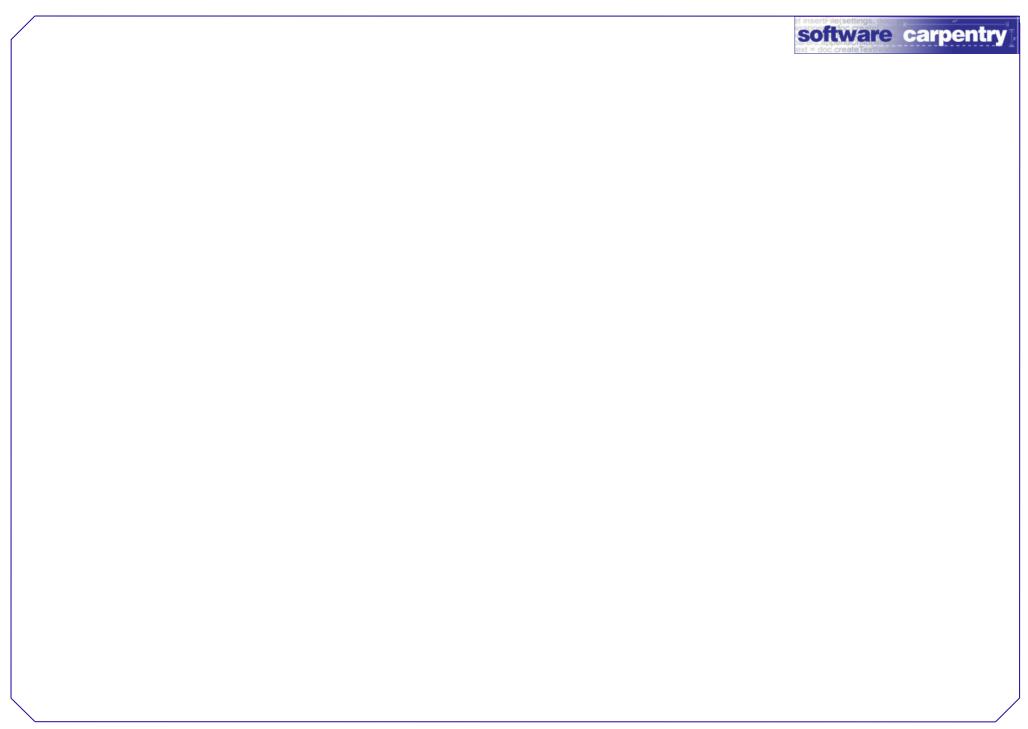
```
elements
```

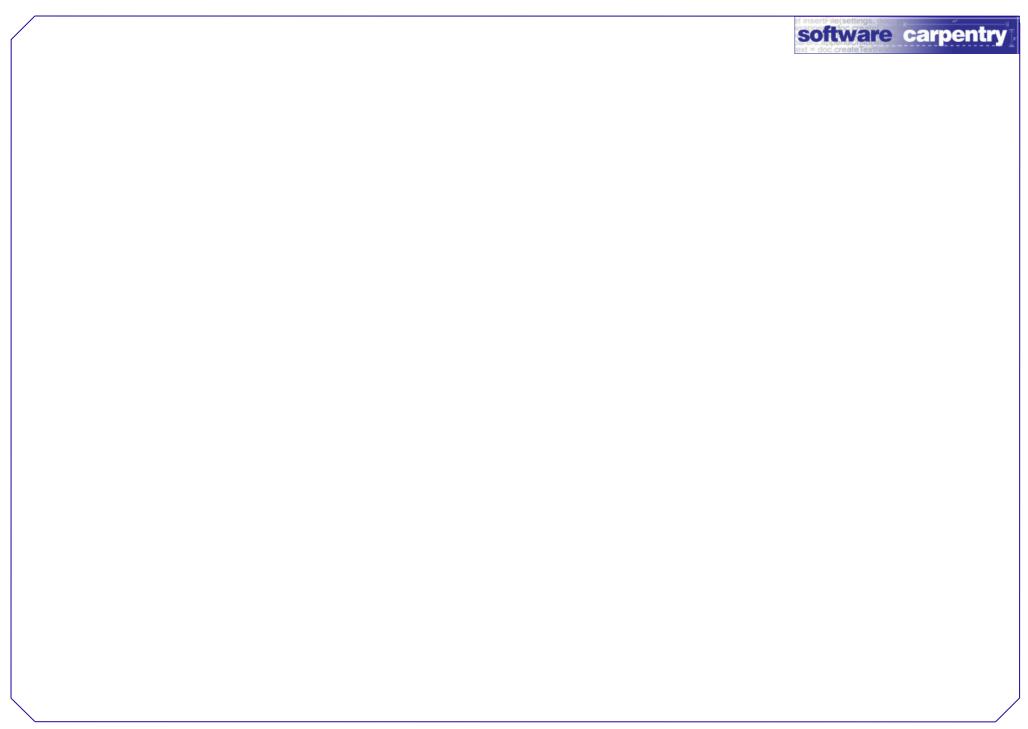
Не

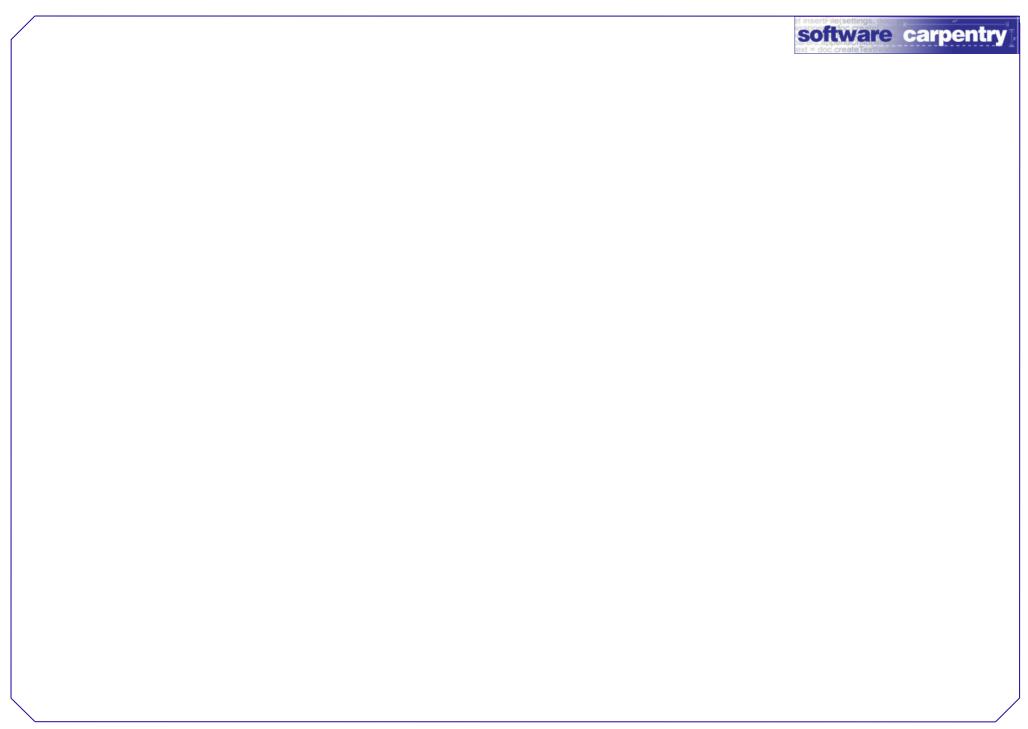
Ne

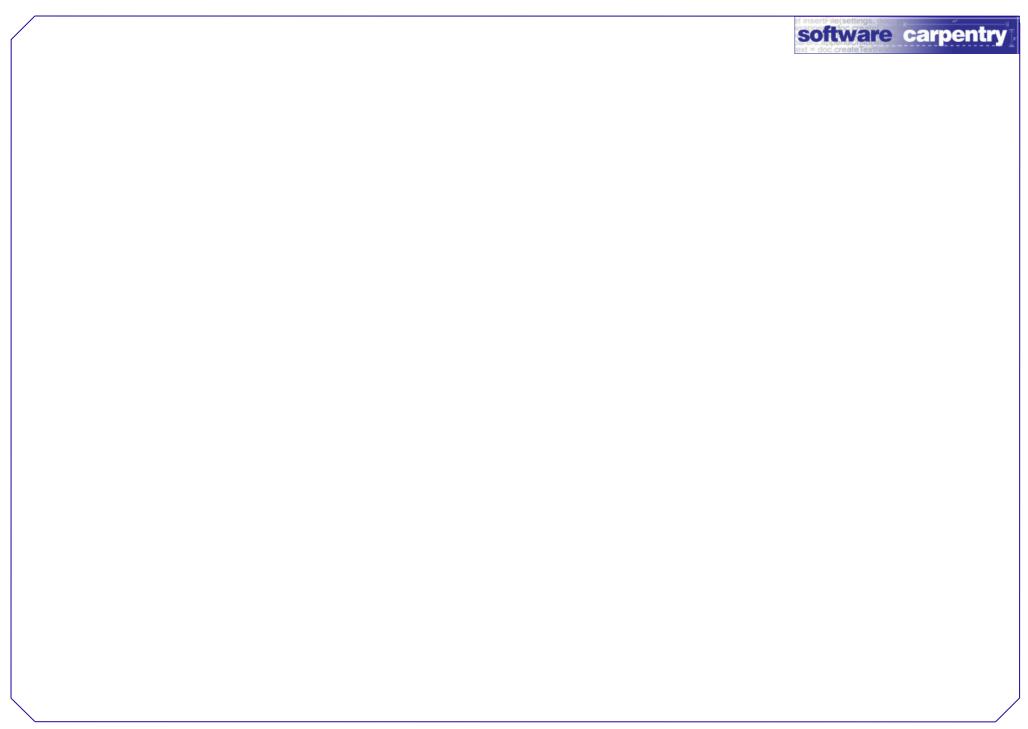
Ar

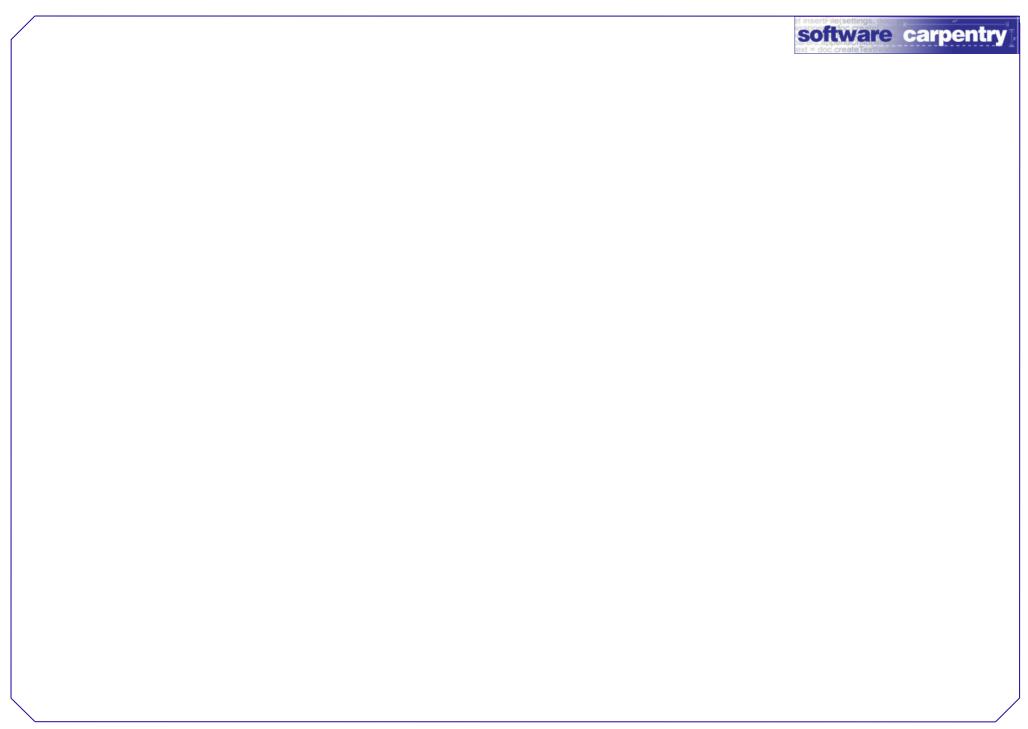
Kr















```
writer = open('temp. txt', 'w')
print >> writer, 'elements'
for gas in ['He', 'Ne', 'Ar', 'Kr']:
    print >> writer, gas
writer.close()
```



```
writer = open('temp.txt', 'w')
print >> writer, 'elements'
for gas in ['He', 'Ne', 'Ar', 'Kr']:
    print >> writer, gas
writer.close()
```

Specify open file after >>



```
writer = open('temp. txt', 'w')
print >> writer, 'elements'
for gas in ['He', 'Ne', 'Ar', 'Kr']:
    print >> writer, gas
writer.close()
```

print automatically adds the newline

Python





```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
writer = open('temp.txt', 'w')
write.write(data)
writer.close()
```



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
writer = open('temp.txt', 'w')
write.write(data)
writer.close()
```



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
writer = open('temp.txt', 'w')
write.write(data)
writer.close()
Write all
```



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
writer = open('temp.txt', 'w')
write.write(data)
writer.close()
```

Probably won't work with a terabyte...



```
reader = open('haiku.txt', 'r')
data = reader.read()
reader.close()
writer = open('temp.txt', 'w')
write.write(data)
writer.close()
```

Probably won't work with a terabyte...

...but we probably don't care





```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    writer.write(line)
reader.close()
writer.close()
```



```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    writer.write(line)
reader.close()
writer.close()
```

Assumes the file is text



```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    writer.write(line)
reader.close()
writer.close()
```

Assumes the file is text

Or at least that the end-of-line character appears frequently





```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    print >> writer, line
reader.close()
writer.close()
```



```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    print >> writer, line
reader.close()
writer.close()
```

Python keeps the newline when reading



```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    print >> writer, line
reader.close()
writer.close()
```

Python keeps the newline when reading print automatically adds a newline



```
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
for line in reader:
    print >> writer, line
reader.close()
writer.close()
```

Python keeps the newline when reading print automatically adds a newline Result is double-spaced output





```
BLOCKSIZE = 1024
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
data = reader.read(BLOCKSIZE)
while len(data) > 0:
    writer.write(data)
    data = reader.read(BLOCKSIZE)
reader.close()
writer.close()
```



```
BLOCKSIZE = 1024
reader = open('haiku.txt', 'r')
writer = open('temp.txt', 'w')
data = reader.read(BLOCKSIZE)
while len(data) > 0:
    writer.write(data)
    data = reader.read(BLOCKSIZE)
reader.close()
writer.close()
```

(Needlessly?) harder to understand



created by

Greg Wilson

October 2010



Copyright © Software Carpentry 2010

This work is licensed under the Creative Commons Attribution License See http://software-carpentry.org/license.html for more information.