Programming with Python

File I/O

File I/O

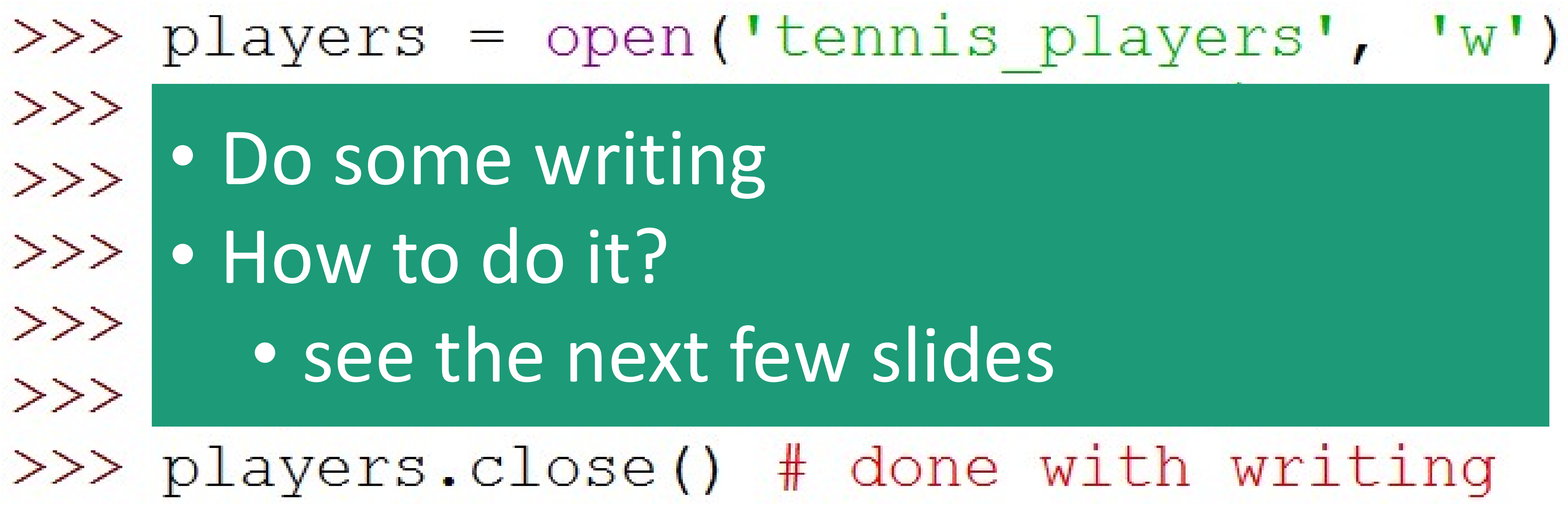
- Files are persistent storage
- Allow data to be stored beyond program lifetime
- The basic operations on files are
  - open, close, read, write
- Python treat files as sequence of lines
  - sequence operations work for the data read from files

File I/O: open and close

open(filename, mode)

- While opening a file, you need to supply
  - The name of the file, including the path
  - The mode in which you want to open a file
  - Common modes are r (read), w (write), a (append)
- Mode is optional, defaults to r
- open(..) returns a file object
- close() on the file object closes the file
  - finishes any buffered operations

File I/O: Example



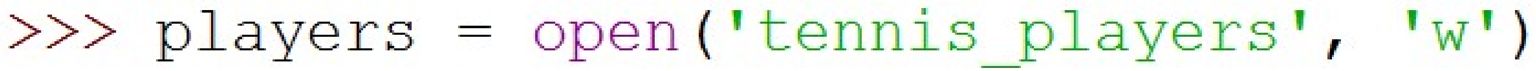
```
>>> players = open('tennis_players', 'w')
>>>
>>>
>>>
>>>
>>>
>>> players.close() # done with writing
```

- Do some writing
- How to do it?
  - see the next few slides

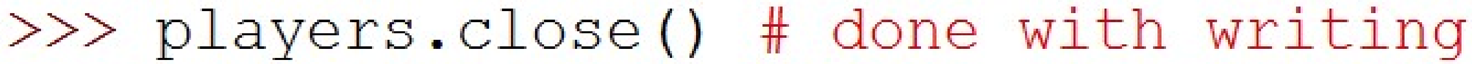
File I/O: read, write and append

- Reading from an open file returns the contents of the file
    - as **sequence** of lines in the program
- Writing to a file
    - IMPORTANT: If opened with mode 'w', **clears** the existing contents of the file
    - Use append mode ('a') to preserve the contents
    - Writing happens at the end

File I/O: Examples

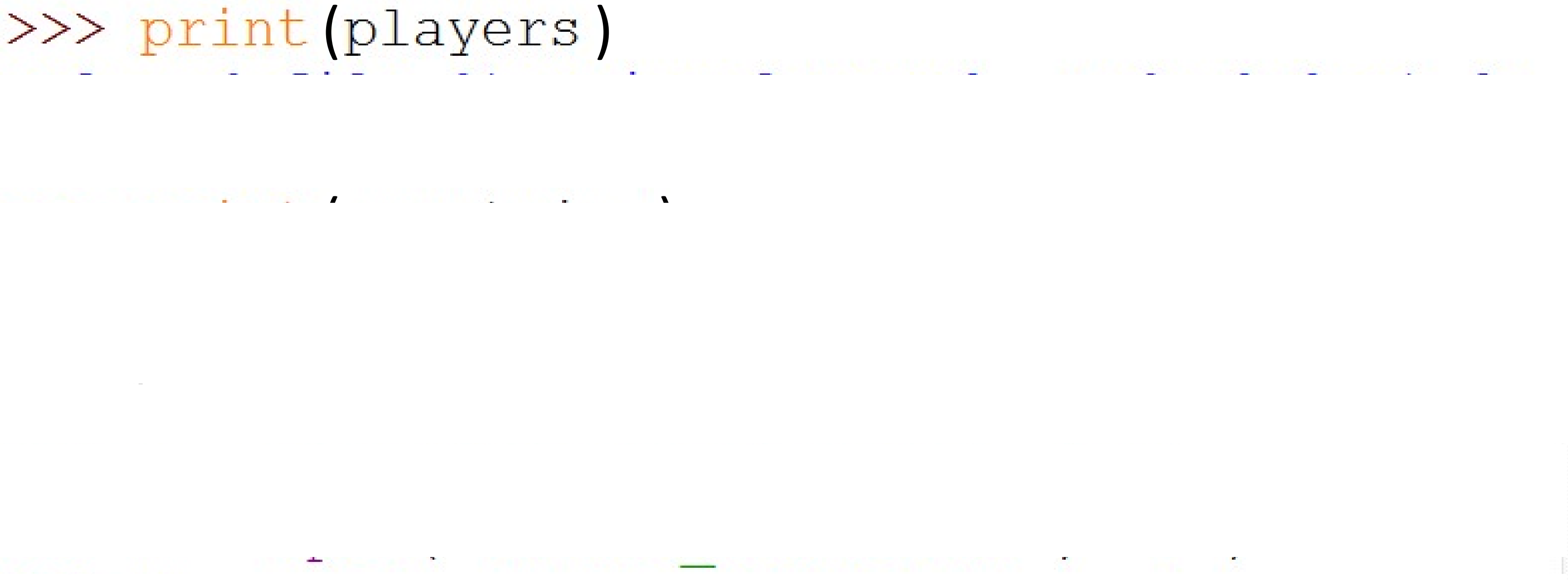


```
>>> players = open('tennis_players', 'w')
```

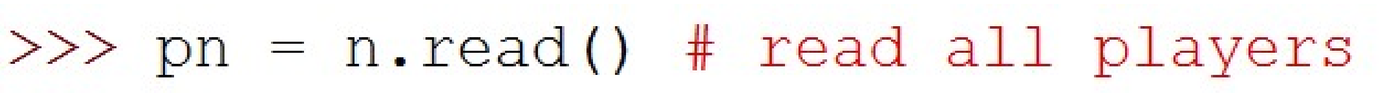


```
>>> players.close() # done with writing
```

File I/O: Examples

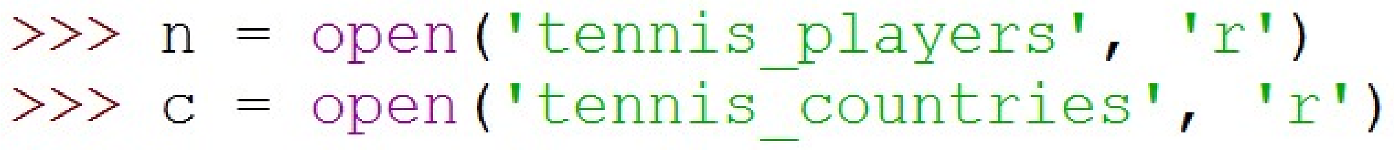


```
>>> print(players)
```



```
>>> pn = n.read() # read all players
```

File I/O: Examples



```
>>> n = open('tennis_players', 'r')
>>> c = open('tennis_countries', 'r')
```

f for ... in

File I/O: Examples