

Reading from Files (with time traveling)

MONTH

OCT

DAY

28

YEAR

2009

11

HOUR

04

DESTINATION TIME

MONTH

OCT

DAY

28

YEAR

2009

PM

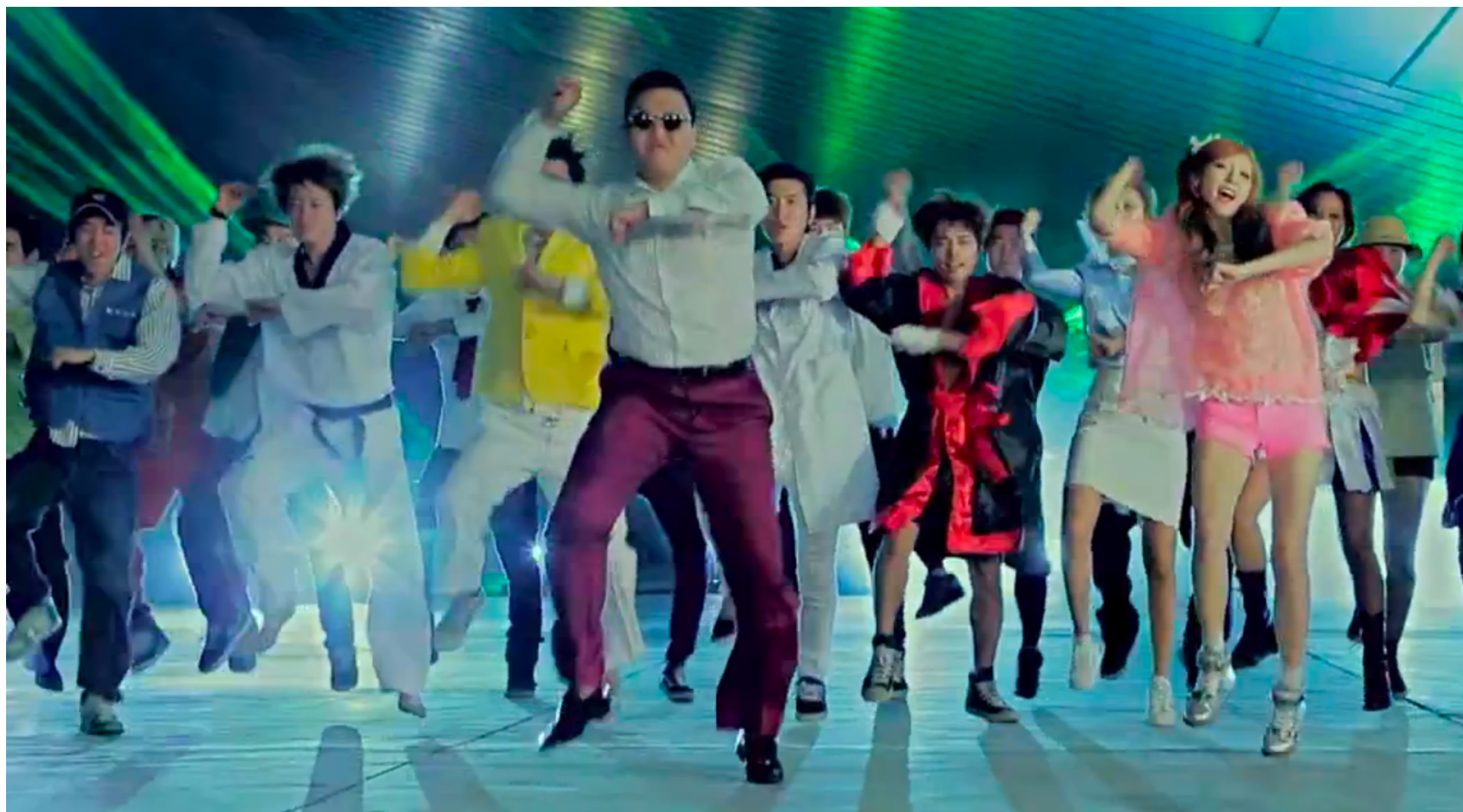
HOUR

04

PRESENT TIME









Chacha-chacha-chacha-chow!





One: Open
the file

Step 1: Open the file

- Uses the `open()` function.
- Always done the same way no matter how the file is organized.

```
file = open("filename.txt", "r")
```

`open()` returns a "file object," which is a data type like `int`, `float`, or `string`.

Replace this string with the real name of your file (don't forget the quotes!)

The "r" means open the file for reading.



Two: loop
over the file.

Step 2: Loop over the file

```
file = open("filename.txt", "r")  
for line in file:
```

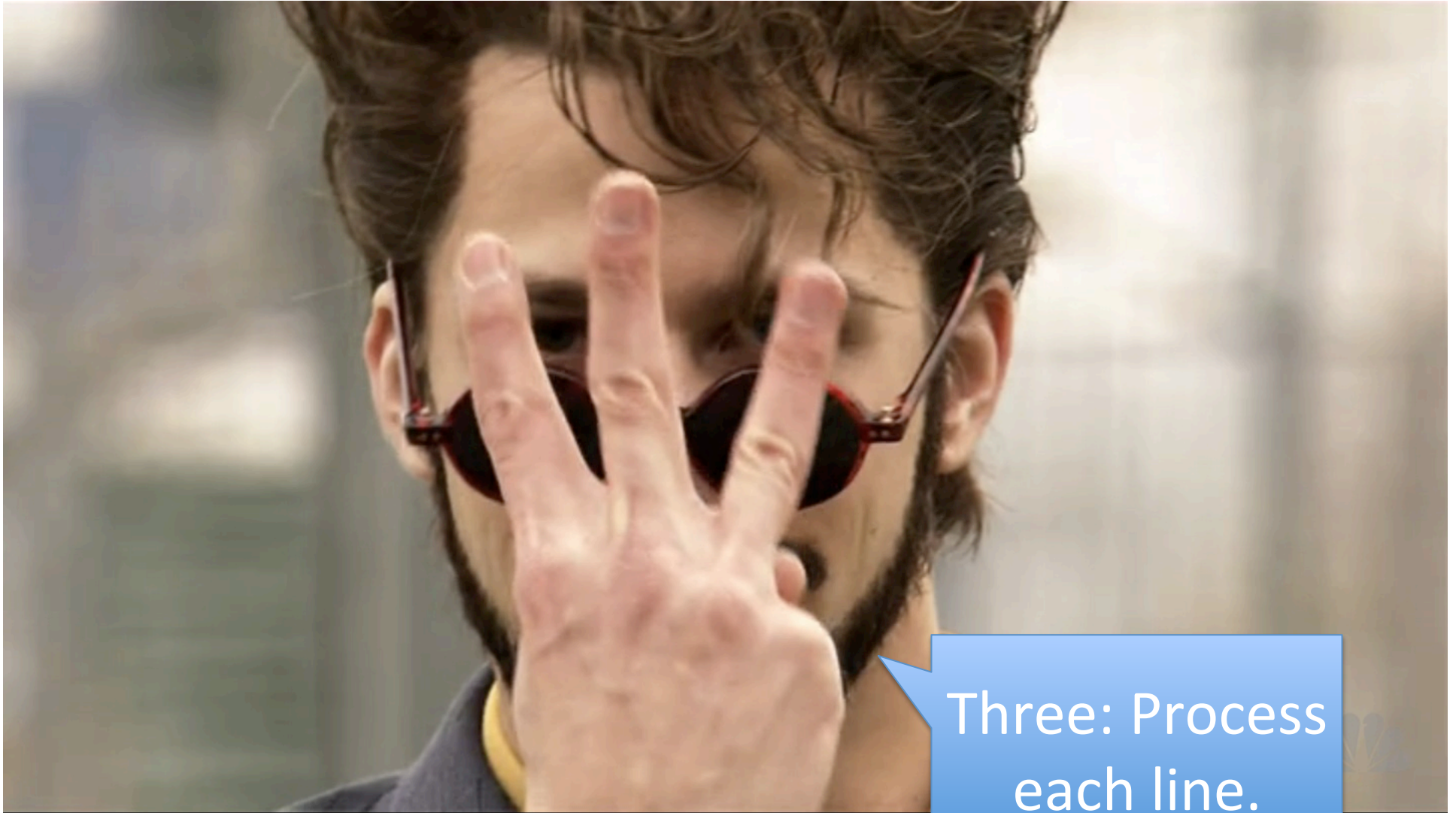
line can be any string variable you want. This variable will store each line of the file as it is read.



Step 2: Loop over the file

```
file = open("filename.txt", "r")  
while [there are more lines in the  
file that we haven't read]:  
    line = [read the next line  
from the file]
```

You only have access to one line of the file at a time.



Three: Process
each line.

Step 3: Process each line

- Do whatever you need to do with the string variable (usually called line).

```
file = open("filename.txt", "r")  
for line in file:  
    print(line)
```

Step 3: Process each line

- Usually a good idea to "strip" the newline character from the line before processing:

```
file = open("filename.txt", "r")
for line in file:
    line = line.rstrip()
    print(line)
```

Step 4: Close the file

- After you are done reading from the file, you should close the file:

```
file = open("filename.txt", "r")
for line in file:
    line = line.rstrip()
    print(line)
file.close()
```


Complete file-reading loop

- Use this as a template for file reading:

```
file = open("filename.txt", "r")
for line in file:
    line = line.rstrip()
    [process a line]
file.close()
```