

```

# This file scrapes data from a file and saves it to a CSV
# Author: [REDACTED]
# Version: 1.0
# Date: [REDACTED]

import os
import sys
import re
import time
from selenium import webdriver
from bs4 import BeautifulSoup
import pandas as pd

# Check if file exists
if len(sys.argv) < 2:
    print("Usage: python script.py <file>")
    sys.exit(1)

file = sys.argv[1]
if not os.path.exists(file):
    print(f"File '{file}' does not exist!")
    sys.exit(1)

line = int(input("From which line you want to scrape the data? "))

# Read file
with open(file, 'r') as f:
    lines = f.readlines()

# Print header
print("\n".join(lines[:line]))

# Print data
for i in range(line, len(lines)):
    print("\n".join(lines[i:i+10]))

```



# Python basic:

## ▼ Python Comments:

### 1. Single Line Comments

```
#This is a comment
print("Hello, World!")
```

### 2. Multiline Comments

```
"""
This is a comment
written in
more than just one line
"""

print("Hello, World!")
```

## ▼ Python Variables:

## Variables:

### 1. Creating Variables

```
x = 5 # x is of type int  
y = "John" # y is of type str  
print(x)  
print(y)
```

### 2. Casting

```
x = str(3) # x will be '3'  
y = int(3) # y will be 3  
z = float(3) # z will be 3.0
```

### 3. Get The Type

```
x = 5  
y = "John"  
print(type(x))  
print(type(y))
```

### 4. Single and Double Quotes

```
x = "John"  
# is the same as  
x = 'John'
```

### 5. Case-Sensitive

```
a = 4  
A = "Sally"  
#A will not overwrite a
```

### 6. Variable Names

```
# Legal variable names:
```

```
myvar = "John"  
my_var = "John"  
_my_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"
```

```
# Illegal variable names:
```

```
2myvar = "John"  
my-var = "John"  
my var = "John"
```

## 7. Multi Words Variable Names

```
# Camel Case
```

```
# Each word, except the first, starts with a capital letter:  
myVariableName = "John"
```

```
# Pascal Case
```

```
# Each word starts with a capital letter:  
MyVariableName = "John"
```

```
# Snake Case
```

```
# Each word is separated by an underscore character:  
my_variable_name = "John"
```

## 8. Assign Multiple Values

```
# Many Values to Multiple Variables
```

```
x, y, z = "Orange", "Banana", "Cherry"  
print(x)  
print(y)  
print(z)
```

```
# One Value to Multiple Variables
x = y = z = "Orange"
print(x)
print(y)
print(z)

# Unpack a Collection
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

## 9. Output Variables

```
x = "Python"
y = "is"
z = "awesome"
print(x, y, z) # you output multiple variables, separated by a comma
print(x + y + z) # also use the + operator to output multiple variables
```

## 10. Global Variables

```
# Without using global variable
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```

```

# Using Global Variable
x = "awesome"

def myfunc():
    global x
    x = "fantastic"

myfunc()

print("Python is " + x)

```

## ▼ Python data types:

### 1. Built-in Data Types

Text Type:	<code>str</code>
Numeric Types:	<code>int</code> , <code>float</code> , <code>complex</code>
Sequence Types:	<code>list</code> , <code>tuple</code> , <code>range</code>
Mapping Type:	<code>dict</code>
Set Types:	<code>set</code> , <code>frozenset</code>
Boolean Type:	<code>bool</code>
Binary Types:	<code>bytes</code> , <code>bytearray</code> , <code>memoryview</code>
None Type:	<code>NoneType</code>

### 2. Getting the Data Type

```

x = 5
print(type(x))

```

### 3. Setting the Specific Data Type

<code>x = str("Hello World")</code>	<code>str</code>
<code>x = int(20)</code>	<code>int</code>
<code>x = float(20.5)</code>	<code>float</code>
<code>x = complex(1j)</code>	<code>complex</code>

## ▼ Python Numbers:

### 1. Int

```
x = 1  
y = 35656222554887711  
z = -3255522
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

### 2. Float

```
x = 1.10  
y = 1.0  
z = -35.59
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

### 3. Complex

```
x = 3+5j  
y = 5j  
z = -5j
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

### 4. Type Conversion

```
x = 1 # int  
y = 2.8 # float
```

```
z = 1j # complex

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)

#convert from int to complex:
c = complex(x)

print(a)
print(b)
print(c)

print(type(a))
print(type(b))
print(type(c))
# Note: You cannot convert complex numbers into another number type.
```

## 5. Random Number

```
import random

print(random.randrange(1, 10))
# Import the random module, and display a random number from 1 to 9:
```

# ▼ Python Casting:

## 1. Specify a Variable Type

```
# Integers:
x = int(1) # x will be 1
y = int(2.8) # y will be 2
z = int("3") # z will be 3
```

```
# Floats:  
x = float(1) # x will be 1.0  
y = float(2.8) # y will be 2.8  
z = float("3") # z will be 3.0  
w = float("4.2") # w will be 4.2
```

```
# Strings:  
x = str("s1") # x will be 's1'  
y = str(2) # y will be '2'  
z = str(3.0) # z will be '3.0'
```

## ▼ Python Strings:

### String

```
print("Hello")  
print('Hello')
```

#### 1. Quotes Inside Quotes

```
print("It's alright")  
print("He is called 'Johnny'")  
print('He is called "Johnny"')
```

#### 2. Assign String to a Variable

```
a = "Hello"  
print(a)
```

#### 3. Multiline Strings

```
a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,
```

```
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.""  
print(a)
```

```
a = '''Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.''  
print(a)
```

#### 4. Strings are Arrays

```
a = "Hello, World!"  
print(a[1])
```

#### 5. Looping Through a String

```
for x in "banana":  
    print(x)
```

#### 6. String Length

```
a = "Hello, World!"  
print(len(a))
```

#### 7. Check String

```
txt = "The best things in life are free!"  
print("free" in txt)
```

```
txt1 = "The best things in life are free!"  
if "free" in txt1:  
    print("Yes, 'free' is present.")
```

#### 8. Check if NOT

```
txt = "The best things in life are free!"  
print("expensive" not in txt)  
  
txt1 = "The best things in life are free!"  
if "expensive" not in txt1:  
    print("No, 'expensive' is NOT present.")
```

## Slicing Strings

```
# Get the characters from position 2 to position 5 (not included):  
  
b = "Hello, World!"  
print(b[2:5])
```

### 1. Slice From the Start

```
# Get the characters from the start to position 5 (not included):  
  
b = "Hello, World!"  
print(b[:5])
```

### 2. Slice To the End

```
# Get the characters from position 2, and all the way to the end:  
  
b = "Hello, World!"  
print(b[2:])
```

### 3. Negative Indexing

```
# From: "o" in "World!" (position -5)  
  
# To, but not included: "d" in "World!" (position -2):
```

```
b = "Hello, World!"  
print(b[-5:-2])
```