Exe 1

Xin-Xins-MacBook-Air:~ xin$ cd temp

Xin-Xins-MacBook-Air:temp xin$ less ex1.py

Xin-Xins-MacBook-Air:temp xin$ cat ex1.py

Xin-Xins-MacBook-Air:temp xin$ hello world

-bash: hello: command not found

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py

Xin-Xins-MacBook-Air:temp xin$ touch ex2.py

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py ex2.py

Xin-Xins-MacBook-Air:temp xin$ rm ex2.py

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py

Xin-Xins-MacBook-Air:temp xin$ cat ex1.py Hello world

cat: Hello: No such file or directory

cat: world: No such file or directory

Xin-Xins-MacBook-Air:temp xin$ cat ex1.py

Xin-Xins-MacBook-Air:temp xin$ Hello World

-bash: Hello: command not found

Xin-Xins-MacBook-Air:temp xin$ hello world!

-bash: hello: command not found

Xin-Xins-MacBook-Air:temp xin$ print("hello World!")

-bash: !": event not found

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py

Xin-Xins-MacBook-Air:temp xin$ python3.6 ex1.py

-bash: python3.6: command not found

Xin-Xins-MacBook-Air:temp xin$ python3.6 ex1.py

-bash: python3.6: command not found

Xin-Xins-MacBook-Air:temp xin$ less ex1.py

Xin-Xins-MacBook-Air:temp xin$ cat ex1.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 py

python3.7: can't open file 'py': [Errno 2] No such file or directory

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

hello world!

hello again

I like typing this.

This is fun.

Yay! Printing.

I'd much rather you 'not'.

I "said" do not touch this

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

hello world!

hello again

I like typing this.

This is fun.

Yay! Printing.

I'd much rather you 'not'.

I "said" do not touch this

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

File "ex1.py", line 8

print("error example.)

^

SyntaxError: EOL while scanning string literal

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

hello world!

hello again

I like typing this.

This is fun.

Yay! Printing.

I'd much rather you 'not'.

I "said" do not touch this

error example.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py

hello world!

hello again

I like typing this.

This is fun.

Yay! Printing.

I'd much rather you 'not'.

I "said" do not touch this

error example.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py line 7

hello world!

hello again

I like typing this.

This is fun.

Yay! Printing.

I'd much rather you 'not'.

I "said" do not touch this

error example.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py, line 5

python3.7: can't open file 'ex1.py,': [Errno 2] No such file or directory

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py/line 5

python3.7: can't open file 'ex1.py/line': [Errno 20] Not a directory

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex1.py.line 5

python3.7: can't open file 'ex1.py.line': [Errno 2] No such file or directory

exe 2

Xin-Xins-MacBook-Air:temp xin$ touch ex2.py

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py ex2.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex2.py

this will run

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex2.py

this will run

hi # there

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex2.py

this will run

hi # there

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex2.py

this will run

hi # there

print

Xin-Xins-MacBook-Air:temp xin$

Exe 3

Xin-Xins-MacBook-Air:temp xin$ touch ex3.py

Xin-Xins-MacBook-Air:temp xin$ ls

ex1.py ex2.py ex3.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex3.py

I will count my cows:

hens 30.0

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex3.py

I will count my different kinds of chickens:

hens 30.0

rooster 97

now i will count the eggs:

6.75

Is it true that 3 + 2 < 5 - 7?

False

What is 3 + 2? 5

What is 5 - 7? -2

Oh, that's why it's False.

How about some more

Is it greater? True

Is it great or equal? True

Is it less or equal? False

Xin-Xins-MacBook-Air:temp xin$ touch caculate.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex1.py ex2.py ex3.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 caculate.py

How many characters in the movie Harry Potter?

Gryffindor 70

Slytherin 50.0

Professor Snap Knew the number of the Hufflepuff

59

How many people in the Ravenclaw

49.0

what is 4+5? 9

what is 8/2? 4.0

Is it greater? False

Xin-Xins-MacBook-Air:temp xin$

Exe 4

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex4.py

There are 100 cars available.

there are only 30 drivers available.

There will be 70 empty cars today.

We can tranport 120 people today.

We have 90 to carpool today.

We need to put about 3.0 in each car.

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex1.py ex2.py ex3.py ex4.py

Xin-Xins-MacBook-Air:temp xin$ touch caculate1.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex1.py ex3.py

caculate1.py ex2.py ex4.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 caculate1.py

There are 20 tables aviable

There are 120 people can sit in the tables

We have 100 people today

We have 20 seats left

Xin-Xins-MacBook-Air:temp xin$

Exe 5

Xin-Xins-MacBook-Air:temp xin$ python ex5.py

File "ex5.py", line 8

print(f "let's talk about {my\_name}.")

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python ex5.py

File "ex5.py", line 8

print(f "Let's talk about {my\_name}.")

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python 3.7 ex5.py

python: can't open file '3.7': [Errno 2] No such file or directory

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5.py

File "ex5.py", line 8

print(f "Let's talk about {my\_name}")

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5.py

File "ex5.py", line 8

print(f "Let's talk about {my\_name}. ")

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5.py

File "ex5.py", line 9

print(f "I am {my\_age} years old.")

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5.py

Let's talk about Xin Xin.

I am 25 years old.

I am 166 cm tall.

I am 50 kg heavy.

Actually I have really great body shape.

I have brown eyes.

They look really good.

My teeth are usually white, and I really like that white color.

If I add 25, 166 and 50, I get 241.

Exe5 study drills

Xin-Xins-MacBook-Air:temp xin$ touch ex5studydrills.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex2.py ex5.py

caculate1.py ex3.py ex5studydrills.py

ex1.py ex4.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5studydrills.py

I am a graduate student from Washington state university.

I am in the molecular plant sciences program.

I started from 2019.

I really enjoyed what I have learnt.

There are 12 in our lab.

They are really nice.

We have 5 phd students, and 4 undergraduates.

we have 3 technicians also.

Xin-Xins-MacBook-Air:temp xin$

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex5.py

Let's talk about Xin Xin.

I am 25 years old.

I am 166 cm tall.

I am 50 kg heavy.

Actually I have really great body shape.

I have brown eyes.

They look really good.

My teeth are usually white, and I really like that white color.

If I add 25, 166 and 50, I get 241.

So I am 111.11111111111111 lbs heavy, and I am 5.4461942257217855 ft tall.

Xin-Xins-MacBook-Air:temp xin$

Exe 6

Terminal

Xin-Xins-MacBook-Air:temp xin$ touch ex6.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex2.py ex5.py

caculate1.py ex3.py ex5studydrills.py

ex1.py ex4.py ex6.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex6.py

There are 10 types of people.

Those who know binary and those who don't.

I said: There are 10 types of people.

I also said: 'Those who know binary and those who don't.'

Isn't that joke so funny?! False

This is the left side of ...a string with a right side.

Xin-Xins-MacBook-Air:temp xin$

Exe6 study drills

types\_of\_people = 10

#name the types\_of\_people equals 10

x = f"There are {types\_of\_people} types of people."

#string is put inside of a string

binary = "binary"

#name binary equals binary

do\_not = "don't"

#name do not equals don't

y = f"Those who know {binary} and those who {do\_not}."

#there are two strings putting inside into one string

print(x)

#export results

print(y)

#export results

print(f"I said: {x}")

#embed x in a string

print(f"I also said: '{y}'")

#embed y in a string

hilarious = False

#name hilarious equals false

joke\_evaluation = "Isn't that joke so funny?! {}"

#name joke\_evaluation equals Isn't that joke so funny?! {}

print(joke\_evaluation.format(hilarious))

#embed variable hilarious inside string

w = "This is the left side of ..."

#name w equals This is the left side of

e = "a string with a right side."

#name e equals a string with a right side

print(w + e)

#export results

3. I think there are 5 places where is a string is putting inside a string.

4.we could separate one string into 2 strings and use + to put them into together.

Exe7

Xin-Xins-MacBook-Air:temp xin$ touch ex7.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex3.py ex6.py

caculate1.py ex4.py ex7.py

ex1.py ex5.py

ex2.py ex5studydrills.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex7.py

Mary has a little lamb.

Its fleece was white as snow.

And everywhere that Mary went.

..........

Cheese Burger

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex7.py

Mary has a little lamb.

Its fleece was white as snow.

And everywhere that Mary went.

..........

Cheese

Burger

Exe7 study drills

print("Mary has a little lamb.")

#export text "Mary has a little lamb."

print ("Its fleece was white as {}.".format('snow'))

#embed snow as a string inside another string and export

print("And everywhere that Mary went.")

#export text "And everywhere that Mary went."

print("." \* 10)

#export 10 .

end1 = "C"

#name end1 equals C

end2 = "h"

#name end2 equals h

end3 = "e"

#name end3 equals e

end4 = "e"

#name end4 equals e

end5 = "s"

#name end5 equals s

end6 = "e"

#name end6 equals e

end7 = "B"

#name end7 equals B

end8 = "u"

#name end8 equals u

end9 = "r"

#name end9 equals r

end10 = "g"

#name end10 equals g

end11 = "e"

#name end11 equals e

end12 = "r"

#name end12 equals r

print(end1 + end2 + end3 + end4 + end5 + end6, end= ' ')

#exports from end1 to end6, and use end= ' ' let the next exported results in the same line

print(end7 + end8 + end9 + end10 + end11 + end12)

#exports from end7 to end12

Exe 8

Xin-Xins-MacBook-Air:temp xin$ touch ex8.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex3.py ex6.py

caculate1.py ex4.py ex7.py

ex1.py ex5.py ex8.py

ex2.py ex5studydrills.py

in-Xins-MacBook-Air:temp xin$ python3.7 ex8.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex8.py

1 2 3 4

one two three four

True False False True

Traceback (most recent call last):

File "ex8.py", line 6, in <module>

print(fomatter.format(formatter, formatter, formatter, formatter))

NameError: name 'fomatter' is not defined

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex8.py

1 2 3 4

one two three four

True False False True

{} {} {} {} {} {} {} {} {} {} {} {} {} {} {} {}

Traceback (most recent call last):

File "ex8.py", line 8, in <module>

"Try your"

IndexError: tuple index out of range

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex8.py

1 2 3 4

one two three four

True False False True

{} {} {} {} {} {} {} {} {} {} {} {} {} {} {} {}

Try your Own text here Maybe a poem Or a song about fear

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex8.py

1 2 3 4

one two three four

True False False True

{} {} {} {} {} {} {} {} {} {} {} {} {} {} {} {}

Try your Own text here Maybe a poem Or a song about fear

Exe 8 study drills

formatter = "{} {} {} {}"

#name formatter equals quadruple {}

print(formatter . format (1,2,3,4))

#name every single {} in formatter seperately as 1, 2, 3, 4, and export results

print(formatter.format("one","two", "three", "four"))

#name every single {} in formatter seperately as one, two, three, four, and export results

print(formatter.format(True, False, False, True))

#name every single {} in formatter seperately as True, False, False, True

print(formatter.format(formatter, formatter, formatter, formatter))

#name every single {} in formatter seperately as quadruple formatter, each formatter equals quadruple {}

print(formatter.format(

"Try your",

"Own text here",

"Maybe a poem",

"Or a song about fear"))

#name every single {} in formatter seperately as a poem, respectively

Exe 9

Xin-Xins-MacBook-Air:temp xin$ touch ex9.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex3.py ex6.py

caculate1.py ex4.py ex7.py

ex1.py ex5.py ex8.py

ex2.py ex5studydrills.py ex9.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex9.py

File "ex9.py", line 7

print("here are the months:" months)

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex9.py

here are the days: Monday Tuesday Wednesday Thursday Friday Saturday Sunday

here are the months: Jan

Feb

Mar

Apr

May

Jun

Jul

Aug

There's something going on here.

With the three double-quotes.

We'll be able to type as much as we like.

Even 4 lines if we want, or 5, or 6.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex9.py

here are the days: Monday Tuesday Wednesday Thursday Friday Saturday Sunday

here are the months:

Jan

Feb

Mar

Apr

May

Jun

Jul

Aug

There's something going on here.

With the three double-quotes.

We'll be able to type as much as we like.

Even 4 lines if we want, or 5, or 6.

Exe 9 study drills

# Here are some new stange stuff, remember tyoe it exactly.

days = "Monday Tuesday Wednesday Thursday Friday Saturday Sunday"

#name days as Monday Tuesday Wednesday Thursday Friday Saturday Sunday

months = "\nJan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"

#name days as Jan Feb Mar Apr May Jun Jul Aug

#\n means going to next line

print("here are the days:", days)

#export text

print("here are the months:", months)

#export text

print("""

There's something going on here.

With the three double-quotes.

We'll be able to type as much as we like.

Even 4 lines if we want, or 5, or 6.

""")

#export text

#"""means all the contents in that text are just exported text, no other means.

Exe 10

Xin-Xins-MacBook-Air:temp xin$ touch ex10.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex3.py ex7.py

caculate1.py ex4.py ex8.py

ex1.py ex5.py ex9.py

ex10.py ex5studydrills.py

ex2.py ex6.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex10.py

I'm tabbed in.

I'm split

on a line.

I'm \ a \ cat.

I'll do a list:

\* Cat food

\* fishies

\* catnip

\* Grass

tabby\_cat = "\tI'm tabbed in."

persian\_cat = "I'm split\non a line."

backslash\_cat = "I'm \\ a \\ cat."

fat\_cat = """

I'll do a list:

\t\* Cat food

\t\* fishies

\t\* catnip\n\t\* Grass

"""

print(tabby\_cat)

print(persian\_cat)

print(backslash\_cat)

print(fat\_cat)

exe 10 study drills

2. tabby\_cat = "\tI'm tabbed in."

persian\_cat = "I'm split\non a line."

backslash\_cat = "I'm \\ a \\ cat."

fat\_cat = '''

I'll do a list:

\t\* Cat food

\t\* fishies

\t\* catnip\n\t\* Grass

'''

print(tabby\_cat)

print(persian\_cat)

print(backslash\_cat)

print(fat\_cat)

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex10.py

I'm tabbed in.

I'm split

on a line.

I'm \ a \ cat.

I'll do a list:

\* Cat food

\* fishies

\* catnip

\* Grass

In many conditions, ‘’’ has the same function with “””.

3.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex10studydrill.py

1 3 5 6

red is not 'yellow

blue is blue

white eaquls everything

and black means black.

flag = "{} {} {} {}"

print(flag.format (1,3,5,6))

print(flag. format(

"\n\tred is not \'yellow",

"\n\tblue is blue",

"\n white eaquls everything",

"\n\tand black means black."

))

Exe 11

Xin-Xins-MacBook-Air:temp xin$ touch ex11.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex11.py ex5studydrills.py

caculate1.py ex2.py ex6.py

ex1.py ex3.py ex7.py

ex10.py ex4.py ex8.py

ex10studydrill.py ex5.py ex9.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex11.py

How old are you? 24

how tall are you? 166

how much do you weight? 101

So,you are 24 old, 166 tall and 101 heavy.

Xin-Xins-MacBook-Air:temp xin$

print("How old are you?", end=' ')

age = input()

print("how tall are you?", end=' ')

height=input()

print("how much do you weight?", end=' ')

weight=input()

print(f"So,you are {age} old, {height} tall and {weight} heavy.")

exe 11 study drills

Xin-Xins-MacBook-Air:temp xin$ touch exe11studydrills.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex2.py ex7.py

caculate1.py ex3.py ex8.py

ex1.py ex4.py ex9.py

ex10.py ex5.py exe11studydrills.py

ex10studydrill.py ex5studydrills.py

ex11.py ex6.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 exe11studydrills.py

I like Harry potter.

Do you like Harry Potter?

yes

Do you like the Game of Throne? yes

Then if you say yes and yes, then we can be friends.

Xin-Xins-MacBook-Air:temp xin$

print("I like Harry potter.")

print("Do you like Harry Potter?")

answer\_1 = input ()

print("Do you like the Game of Throne?", end=' ')

answer\_2 = input()

print(f"Then if you say {answer\_1} and {answer\_2}, then we can be friends.")

ex12

Xin-Xins-MacBook-Air:temp xin$ touch ex12.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex12.py ex6.py

caculate1.py ex2.py ex7.py

ex1.py ex3.py ex8.py

ex10.py ex4.py ex9.py

ex10studydrill.py ex5.py exe11studydrills.py

ex11.py ex5studydrills.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex12.py

How old are you? 24

How tall are you?266

How much do you weight?101

So, you are 24 old, 266 tall and 101 heavy.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex12.py

How old are you? 24

How tall are you?166

How much do you weight?101

So, you are 24 old,

166 tall and 101 heavy.

age = input ("How old are you? ")

height = input ("How tall are you?")

weight = input ("How much do you weight?")

print(f"So, you are {age} old, \n\t{height} tall and {weight} heavy.")

exe 12 study drills

Xin-Xins-MacBook-Air:temp xin$ less ex12.py

Xin-Xins-MacBook-Air:temp xin$

age = input ("How old are you? ")

height = input ("How tall are you?")

weight = input ("How much do you weight?")

print(f"So, you are {age} old, \n\t{height} tall and {weight} heavy.")

exe 13

Xin-Xins-MacBook-Air:temp xin$ touch ex13.py

Xin-Xins-MacBook-Air:temp xin$ ls

caculate.py ex12.py ex5studydrills.py

caculate1.py ex13.py ex6.py

ex1.py ex2.py ex7.py

ex10.py ex3.py ex8.py

ex10studydrill.py ex4.py ex9.py

ex11.py ex5.py exe11studydrills.py

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py first 2nd 3rd

The script is called: ex13.py

Traceback (most recent call last):

File "ex13.py", line 5, in <module>

print("your first variable is:", frist)

NameError: name 'frist' is not defined

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py first 2nd 3rd

The script is called: ex13.py

your first variable is: first

your second variable is: 2nd

your third variable is: 3rd

Xin-Xins-MacBook-Air:temp xin$ python3.7 apple pear orange

Exe 13 study drills

Xin-Xins-MacBook-Air:temp xin$ python3.7 apple pear orange

python3.7: can't open file 'apple': [Errno 2] No such file or directory

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear orange

The script is called: ex13.py

your first variable is: apple

your second variable is: pear

your third variable is: orange

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear orange

The script is called: ex13.py

Traceback (most recent call last):

File "ex13.py", line 5, in <module>

print("your first variable is:", first)

NameError: name 'first' is not defined

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py first pear orange

The script is called: ex13.py

Traceback (most recent call last):

File "ex13.py", line 5, in <module>

print("your first variable is:", apple)

NameError: name 'apple' is not defined

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple orange pear

The script is called: ex13.py

your first variable is: apple

your second variable is: orange

your third variable is: pear

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear banana orange

The script is called: ex13.py

your first variable is: apple

your second variable is: pear

your third variable is: banana

/n orange

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear orange banana

The script is called: ex13.py

your first variable is: apple

your second variable is: pear

your third variable is: orange

/n banana

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple banana orange

Traceback (most recent call last):

File "ex13.py", line 2, in <module>

script, apple, pear, orange, banana = argv

ValueError: not enough values to unpack (expected 5, got 4)

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear orange banana

The script is called: ex13.py

your first variable is: apple

your second variable is: pear

your third variable is: orange

/n banana

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple banana orange

Traceback (most recent call last):

File "ex13.py", line 2, in <module>

script, apple, pear, orange, banana = argv

ValueError: not enough values to unpack (expected 5, got 4)

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple pear orange banana

The script is called: ex13.py

your first variable is: apple

your second variable is: banana

your third variable is: orange

/n banana

from sys import argv

script, apple, pear, orange, banana = argv

print ("The script is called:", script)

print("your first variable is:", apple)

print("your second variable is:", banana)

print("your third variable is:", orange)

print("/n", banana)

3.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex13.py apple orange pear banana

The script is called: ex13.py

your first variable is: apple

your second variable is: banana

your third variable is: pear

/n banana

my favorite fruit is: apple

I like apple very much!

Xin-Xins-MacBook-Air:temp xin$

from sys import argv

script, apple, pear, orange, banana = argv

print ("The script is called:", script)

print("your first variable is:", apple)

print("your second variable is:", banana)

print("your third variable is:", orange)

print("/n", banana)

print("my favorite fruit is:", end=" ")

favorite\_fruit=input ()

print (f"I like {favorite\_fruit} very much!")

exe14

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex14.py Xin

Hi Xin, I'm the ex14.py script.

I'd like to ask you few qustions.

Do you like me Xin?

>yes

where do you live Xin?

>Pullman

what kind of computer do you have?

>macbook air

Alright, so you said yes about liking me.

You live in Pullman. Not sure where that is. And you have a macbook air computer. Nice

from sys import argv

script, user\_name = argv

prompt = '>'

print(f"Hi {user\_name}, I'm the {script} script.")

print("I'd like to ask you few qustions.")

print(f"Do you like me {user\_name}?")

likes = input (prompt)

print(f"where do you live {user\_name}?")

lives = input(prompt)

print ("what kind of computer do you have?" )

computer = input (prompt)

print (f'''

Alright, so you said {likes} about liking me.

You live in {lives}. Not sure where that is. And you have a {computer} computer. Nice ''')

Exe14 study drills

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex14.py Xin

Hi Xin, I'm the ex14.py script.

I'd like to ask you few qustions.

Do you like me Xin?

:p no

where do you live Xin?

:p WA

what kind of computer do you have?

:p apple

Alright, so you said no about liking me.

You live in WA. Not sure where that is. And you have a apple computer. Nice

Xin-Xins-MacBook-Air:temp xin$

from sys import argv

script, user\_name = argv

prompt = ':p '

print(f"Hi {user\_name}, I'm the {script} script.")

print("I'd like to ask you few qustions.")

print(f"Do you like me {user\_name}?")

likes = input (prompt)

print(f"where do you live {user\_name}?")

lives = input(prompt)

print ("what kind of computer do you have?" )

computer = input (prompt)

print (f'''

Alright, so you said {likes} about liking me.

You live in {lives}. Not sure where that is. And you have a {computer} computer. Nice ''')

3.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex14studydrills.py Xin second first

Hi, Xin!

Have you ever learned python before? is that your second time?

>first

Is this your first lesson.

>second

Oh, you learn python first and this is your second lesson!

from sys import argv

script, user\_name, first, second = argv

prompt='>'

print(f"Hi, {user\_name}!")

print(f"Have you ever learned python before? is that your {first} time?")

time\_1 = input (prompt)

print(f"Is this your {second} lesson.")

time\_2 = input (prompt)

print(f"""Oh, you learn python {time\_1} and this is your {time\_2} lesson!""")

exe15

Xin-Xins-MacBook-Air:temp xin$ touch ex15.py

Xin-Xins-MacBook-Air:temp xin$ touch ex15\_example.txt

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex15.py ex15\_example.txt

File "ex15.py", line 13

txt\_again = open(file again)

^

SyntaxError: invalid syntax

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex15.py ex15\_example.txt

here's your file ex15\_example.txt:

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

type the filename again:

>ex15\_example.txt

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

from sys import argv

script, filename = argv

txt = open(filename)

print(f"here's your file {filename}:")

print(txt.read())

print("type the filename again:")

file\_again = input(">" )

txt\_again = open(file\_again)

print(txt\_again.read())

exe15 study drill

4.

from sys import argv

#add features to the script

script, filename = argv

#assign 2 variables can work with

txt = open(filename)

#assign this file's text is the text

print(f"here's your file {filename}:")

#export

print(txt.read())

#export the text of the ex15\_example.txt

print("type the filename again:")

#export

file\_again = input(">" )

#input the name of the ex15\_example.txt

txt\_again = open(file\_again)

#assign txt\_again

print(txt\_again.read())

#export the text of the ex15\_example.txt

5.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex15.py ex15\_example.txt

here's your file ex15\_example.txt:

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

6.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex15.py

type the filename again:

>ex15\_example.txt

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

7.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex15.py ex15\_example.txt

here's your file ex15\_example.txt:

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

type the filename again:

>ex15\_example.txt

This is stuff I typed into a file.

It is really cool stuff .

Lots and lots of fun to have in here.

None

None

from sys import argv

#add features to the script

script, filename = argv

#assign 2 variables can work with

txt = open(filename)

#assign this file's text is the text

print(f"here's your file {filename}:")

#export

print(txt.read())

#export the text of the ex15\_example.txt

print("type the filename again:")

#export

file\_again = input(">" )

#input the name of the ex15\_example.txt

txt\_again = open(file\_again)

#assign txt\_again

print(txt\_again.read())

#export the text of the ex15\_example.txt

print(txt.close())

print(txt\_again.close())

exe16

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex16.py ex16\_test.txt

We are going to erase ex16\_test.txt.

if you do not want that, hit CTRL+C (^C).

if you do want that, hit RETURN

?

Opening the file...

Truncating the file. Goodbye!

now I'm going to ask you for three lines.

line 1: I have a pen

line 2: I have an apple

line 3: apple pen

I'm going to write these to the file.

And finally, we close it.

from sys import argv

script, filename = argv

print (f"We are going to erase {filename}.")

print ("if you do not want that, hit CTRL+C (^C).")

print ("if you do want that, hit RETURN")

#export text

input("?")

#keyboard hit

print ("Opening the file...") #export text

target = open(filename, "w") #assign target as ex16\_test.txt

print("Truncating the file. Goodbye!") #wxport text

target.truncate() # truncate the ex16\_test.txt

print("now I'm going to ask you for three lines.")

#export text

line1 = input("line 1: ")

line2 = input("line 2: ")

line3 = input("line 3: ")

#input three lines

print("I'm going to write these to the file.")

target.write(line1) #export line1

target.write("\n") #next line

target.write(line2) #export line2

target.write("\n")

target.write(line3) # export line3

target.write("\n")

print("And finally, we close it.")

target.close() #close ex16\_test.txt

ex12.py (END)

ex16 study Drill

2. Xin-Xins-MacBook-Air:temp xin$ python ex16\_studydrill.py ex16\_test.txt

I just typed:

I have a pen

I have an apple

apple pen

from sys import argv

script, filename = argv

txt=open (filename)

print (f"I just typed:")

print (txt. read())

ex17

Xin-Xins-MacBook-Air:temp xin$ touch ex17\_test.txt

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex17.py ex16\_test.txt ex17\_test.txt

Copying from ex16\_test.txtto ex17\_test.txt

the input file is 11 bytes long

Does the output file exist? True

ready, hit RETURN to continue, CTRL-C to abort.

Alright, all done.

Xin-Xins-MacBook-Air:temp xin$

from sys import argv

from os.path import exists

script, from\_file, to\_file = argv

print(f"Copying from {from\_file}to {to\_file}")

in\_file = open(from\_file)

indata = in\_file.read()

print(f"the input file is {len(indata)} bytes long")

print(f"Does the output file exist? {exists(to\_file)}")

print("ready, hit RETURN to continue, CTRL-C to abort.")

input()

out\_file = open(to\_file, 'w')

out\_file.write(indata)

print("Alright, all done.")

out\_file.close()

in\_file.close()

ex17 study drill

2.

Xin-Xins-MacBook-Air:temp xin$ python3.7 ex17.py ex16\_test.txt ex17\_test.txt

Copying from ex16\_test.txtto ex17\_test.txt

the input file is 11 bytes long

Does the output file exist? True

ready, hit RETURN to continue, CTRL-C to abort.

Alright, all done.

from sys import argv

from os.path import exists

script, from\_file, to\_file = argv

print(f"Copying from {from\_file}to {to\_file}")

#in\_file = open(from\_file)

#indata = in\_file.read()

indata=open(from\_file).read()

print(f"the input file is {len(indata)} bytes long")

print(f"Does the output file exist? {exists(to\_file)}")

print("ready, hit RETURN to continue, CTRL-C to abort.")

input()

out\_file = open(to\_file, 'w')

out\_file.write(indata)

print("Alright, all done.")

out\_file.close()

#in\_file.close()

3.

Xin-Xins-MacBook-Air:temp xin$ cat ex17\_test.txt

eee

222

Ff

4.

The method close() closes the opened file. A closed file cannot be read or written