



## EXPLORER

## OPEN EDITORS 1 UNSAVED

{ } User Settings C:\Users\pcurtis7\AppData...

percipio42\_reading\_binary\_data.py ...

## PYTHON

percipio13\_tuple\_type.py

percipio14\_slice\_type.py

percipio14a\_list\_copy\_boolean\_chec...

▸ 03\_Collections-Mapping-Looping

▸ 04\_Modules-Functions

▸ 05\_Classes

▸ 06\_Working-with-Files

≡ games - Shortcut.Ink

≡ games.txt

≡ loremipsum - Shortcut.Ink

percipio36\_docstrings.py

percipio37\_code\_comments.py

percipio38\_documentation\_best\_pra..

percipio39\_reading\_text\_files.py

percipio40\_writing\_data.py

percipio41\_writing\_large\_files.py

percipio42\_reading\_binary\_data.py

percipio43\_writing\_binary\_data.py

percipio44\_exercise\_create\_custom\_...

≡ reading\_text\_files - Shortcut.Ink

≡ reading\_text\_files.txt

● sample.avi

≡ writing\_data - Shortcut.Ink

≡ writing\_data.txt

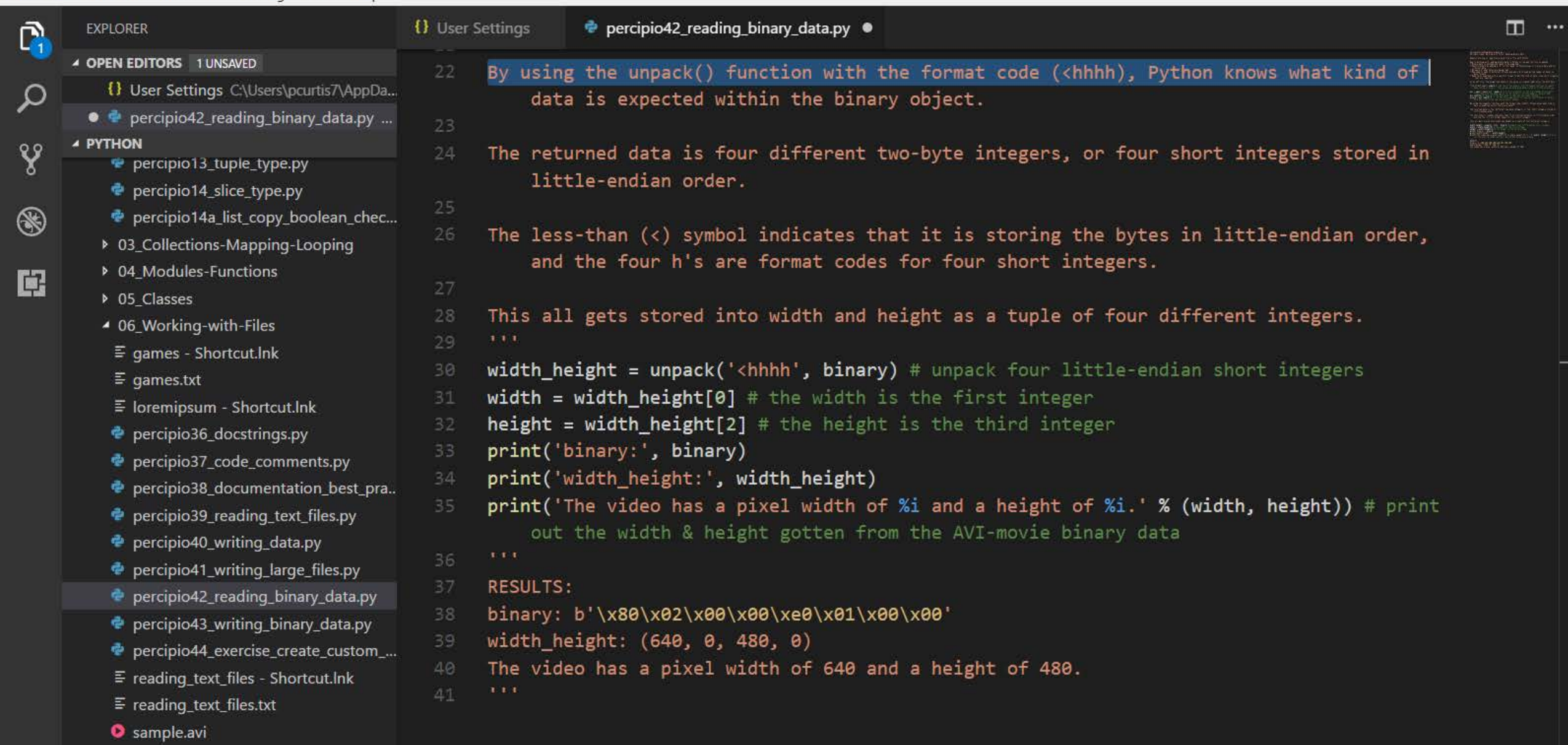
▸ 07\_Comprehensions

{ } User Settings

percipio42\_reading\_binary\_data.py •

```
1  '''
2  percipio42_reading_binary_data.py
3  Percipio video: Working with Files; Reading Binary Data
4
5  Demonstrate how to read binary data from a file with Python
6
7  Key to working with reading binary data in Python is the way the file is opened
8  * Binary data is encoded and needs to be unpacked.
9  * Specify binary by adding a 'b' to the 'mode' so Python knows it's binary data and not
   textual data.
10 * Be aware of what bytes are being read
11 * May need to seek to different offsets and ability to seek by the number of bytes to
   move forward
12 * Need to be unpacked with a specific format to get the kind of data (like short integers)
   that are required
13
14 In an AVI file, the height and width of the movie is located right after the 64th byte
15 '''
16 from struct import unpack # from the Struct module, use the unpack function to unpack
   data stored in binary. This allows working with data stored in C-like structures.
17
18 avi = open('sample.avi', mode='br') # This is similar to the r/w/a modes of previous
   files, but with the addition of 'b' for binary. So this mode is 'binary read'
19 avi.seek(64) # seeking to the 64th byte to then use the read() method,...
20 binary = avi.read(8) # ... of the file-like object to read the next 8 bytes of binary
   data which is stored the the variable 'binary'
21 '''
22 By using the unpack() function with the format code (<hhhh), Python knows what kind of
```





EXPLORER

1 UNSAVED

User Settings C:\Users\pcurtis7\AppData\Local\Microsoft\VisualStudio\16.0\Settings\pcurtis7\...

percipio42\_reading\_binary\_data.py ...

PYTHON

- percipio13\_tuple\_type.py
- percipio14\_slice\_type.py
- percipio14a\_list\_copy\_boolean\_chec...
- 03\_Collections-Mapping-Looping
- 04\_Modules-Functions
- 05\_Classes
- 06\_Working-with-Files
  - games - Shortcut.Ink
  - games.txt
  - loremipsum - Shortcut.Ink
  - percipio36\_docstrings.py
  - percipio37\_code\_comments.py
  - percipio38\_documentation\_best\_pra...
  - percipio39\_reading\_text\_files.py
  - percipio40\_writing\_data.py
  - percipio41\_writing\_large\_files.py
  - percipio42\_reading\_binary\_data.py
  - percipio43\_writing\_binary\_data.py
  - percipio44\_exercise\_create\_custom\_...
  - reading\_text\_files - Shortcut.Ink
  - reading\_text\_files.txt
  - sample.avi

```
22 By using the unpack() function with the format code (<hhhh), Python knows what kind of
23 data is expected within the binary object.
24 The returned data is four different two-byte integers, or four short integers stored in
25 little-endian order.
26 The less-than (<) symbol indicates that it is storing the bytes in little-endian order,
27 and the four h's are format codes for four short integers.
28 This all gets stored into width and height as a tuple of four different integers.
29 '''
30 width_height = unpack('<hhhh', binary) # unpack four little-endian short integers
31 width = width_height[0] # the width is the first integer
32 height = width_height[2] # the height is the third integer
33 print('binary:', binary)
34 print('width_height:', width_height)
35 print('The video has a pixel width of %i and a height of %i.' % (width, height)) # print
36 out the width & height gotten from the AVI-movie binary data
37 '''
38 RESULTS:
39 binary: b'\x80\x02\x00\x00\xe0\x01\x00\x00'
40 width_height: (640, 0, 480, 0)
41 The video has a pixel width of 640 and a height of 480.
42 '''
```