

# Computer Science Programming Fundamentals

## Files



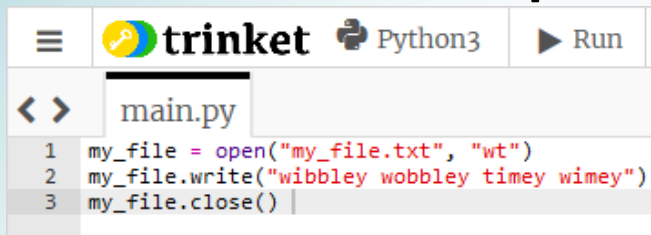
# Files Intro

Most video games save your progress somehow. Otherwise, you would have to start over every time you play!

Similarly, there are a lot of benefits to being able to save some information on the computer so that Python can read it later. Whether you want to save a user's settings or make a word list for a charades game, learning how to work with files in Python makes it easy!

# Files 1 pt 1

- Run the example:



```
1 my_file = open("my_file.txt", "wt")
2 my_file.write("wibbly wobbly timey wimey")
3 my_file.close()
```

Nothing should appear on the right side where we usually see the output, but you will see a new tab open up next to main called `my_file.txt`.

- Open that tab.

What does it say? The same thing we see on line two.

But how does this code work?

# Files pt 2

```
my_file = open("my_file.txt", "wt")
```

The above line tells Python to open a file called "my\_file.txt" and get ready to write text in it (hence the "wt").

```
my_file.write("wibbly wobbly timey  
wimey")
```

The above line tells Python to write a string into the file.

```
my_file.close()
```

When you open the fridge to take something or put something in, do you leave it open when you're done and walk away? Of course not!

- ! In Python, when we are done using a file, we need to close it so we don't waste memory.

# Files pt 3

- Add this to your code and run:

```
my_file = open("my_file.txt", "wt")  
my_file.write("wibbly wobbly timey  
wimey")  
my_file.close()
```

```
my_file = open("my_file.txt", "rt")  
message = my_file.read()  
print(message)  
my_file.close()
```

The first line in our new code tells Python to open the file and get ready to read text. That's what "rt" means.

The line after that tells Python to read the file and save the contents in a variable called message.

Then, we print the message so we can see what was in the file.

Finally, we close it. Don't waste memory!

# Files pt 4

- Add this to your code and run:

```
my_file = open("my_file.txt", "wt")
my_file.write("wibbly wobbly timey wimey\n")
my_file.write("don't \n")
my_file.write("blink \n")
my_file.close()
```

```
my_file = open("my_file.txt", "rt")
message = my_file.read()
print(message)
my_file.close()
```

**Q:** Based on the output, what does `\n` do?

*Hint: Try removing it from one of the lines and run again.*

**A:** `\n` tells Python to go to a new line. Unlike `print()`, `write()` does not automatically add a new line for us.



# Files pt 5

- Make this change and run:

```
my_file = open("my_file.txt", "wt")
```

```
my_file.write("wibbly wobbly timey wimey  
\n")
```

```
my_file.write("don't \n")
```

```
my_file.write("blink \n")
```

```
my_file.close()
```

```
my_file = open("my_file.txt", "rt")
```

```
for line in my_file:
```

```
    if "w" in line:
```

```
        continue
```

```
    print(line)
```

```
my_file.close()
```

**Q: Based on the output, what does  
continue do?**

# Files pt 6

**A:** `continue` tells Python to stop this run of the loop and go to the next one. Any code before `continue` will be run, but any code after it will be skipped until the next time the loop runs.



# Files pt 7



## Activity:

- Let's make an encoder and a decoder to send and receive secret messages!
- This assignment works best if you work with a partner. One student will make the encoder, the other can make the decoder.
- The encoder will take a message and turn it into numbers.
- The decoder will take those numbers and turn them back into letters.

# Files pt 8

## Encoder example:

```
def encode(message):  
    encoded_message = []  
  
    for char in message:  
        code_char = ord(char)  
        code_char_str = str(code_char)  
        encoded_message.append(code_char_str)  
    )  
  
    return encoded_message  
  
my_file = open("code.txt", "wt")  
msg = input("Write the message to encode:  
")  
secret_msg = encode(msg)  
  
for line in secret_msg:  
    my_file.write(line + "\n")  
  
my_file.close()
```

# Files pt 9



## Explanation:

First, we define a function that takes one parameter: `message`.

Inside the function, we make an empty list that we will fill in the next step.

We make a for loop to go through the message, character by character. The `ord()` function turns that character into an int.

On the next line, we turn that int into a string so we don't break the program.

After that, we add our new string to the end of the list we made.

When we are done, we return, or give, the list that we made.

Outside the function, we open a file, let the user write some input, then we encode that input.

In the last for loop, we go through every character in the secret message and write it on a new line in our file.

Finally, we close the file.

# Files pt 10

## Decoder example:

```
def decode(code_file):  
    my_file = open(code_file, "rt")  
  
    message_list = []  
    for line in my_file:  
        line_int = int(line)  
        character = chr(line_int)  
        message_list.append(character)  
  
    message = "".join(message_list)  
  
    print(message)  
  
    my_file.close()  
  
decode("code.txt")
```

# Files pt 11



## Explanation:

We define a function that takes a file name as a parameter.

Once we have the file name, we open it in read mode.

Then, we make an empty list, loop through the lines in our file, and turn each line into an int.

Once we turn each line into an int, we turn each line into a string using the `chr()` function.

Then, we append that string to our new list.

When we are finished, we turn the list into a string using:

```
message = "".join(message_list)
```

This line tells Python to make a variable called `message`, then take every item in `message_list` and join them together. The `""` part tells Python that we don't want anything in between the letters.

To end the function, we print the message and close the file.

On the last line, outside the function, we call the function, passing the file name as a parameter, so that our decoder will run.

**Note:** You will need to make a new file, name it `code.txt`, and paste the encoded message in it for the decoder to work.