10) Recap lesson + Storing data into a file

Agenda

- Q&A about the the first two lessons
- Storing data into a file

Q&A

- Any question regarding the first two Python lessons?
  - Variable types
  - If/else statements
  - Loops
  - etc.

Storing data into a TXT file

> *Do this after the Q&A (but at least 30 minutes before the end of the lesson)*

Every good game needs to **store** the best score, so that you can try to best it.

In the case of the "Guess the secret number" game, we'd like to store the **number of attempts** we needed to guess the number. Did we need 6 attempts? Maybe 4? Or even only 2? The lower the number, the better!

Attempts counter

First let's add the `attempts` counter in our program:

```
import random

secret = random.randint(1, 30)
attempts = 0

while True:
    guess = int(input("Guess the secret number (between 1 and 30): "))
    attempts += 1

    if guess == secret:
        print("You've guessed it - congratulations! It's number " + str(secret))
        print("Attempts needed: " + str(attempts))
        break
    elif guess > secret:
        print("Your guess is not correct... try something smaller")
    elif guess < secret:
        print("Your guess is not correct... try something bigger")
```

Try out the game. As you can see, the program tells you how many attempts you needed to win the game.

We will store our `attempts` score in a simple TXT file in our project folder.

Create a text file

So let's first **create** this TXT file. Create a new file and name it `score.txt`. Enter a number `100` inside, so it's not empty.

Reading from a file

Add the following "with-open" code just above the while loop:

```
with open("score.txt", "r") as score_file:
    best_score = int(score_file.read())
    print("Top score (attempts): " + str(best_score))

while True:
    # the rest of the code...
```

This will:

- **open** the `score.txt` file,
- **read** its contents,
- **save** it into the variable named `best_score`,
- and **print** it in the Terminal.

Run the program. It should say "Top score: 100" in the Terminal.

Writing into the file

Now let's store the number of attempts into the file. Find this `if` statement in the `while` loop (`if guess == secret`). Add the line that saves attempts into the file:

```
if guess == secret:
    with open("score.txt", "w") as score_file:
        score_file.write(str(attempts))
```

As you can see, we used the **"w"** when opening the file. This means that we are opening it for **writing** purposes (and we **overwrite** its contents). If we used "r", we wouldn't be able to write into it, but only read it.

Also note that we have to **convert** the number to string **before** writing it into the text file.

Let's play the game a few times. Does everything work okay?

It's only saving the last result, not the best one!

Oops! Our program **writes the result** into the file **everytime** we finish our game. That's why it saves **every** last score in the text file. But we want it to **only** save the **best** one.

What do we do?

We can compare our `best_score` to the number of `attempts` and see which one is a lower number (remember: lower is better). If `attempts` is lower than `best_score`, we write the `attempts` number into the file.

Let's adapt our code like this:

```
if guess == secret:
    if attempts < best_score:
        with open("score.txt", "w") as score_file:
            score_file.write(str(attempts))
```

So if the player's `guess` is the same as the `secret` and if the number of player's attempts are lower than the best score, open the `score.txt` for writing and enter the number of attempts into it.

> *See the complete example here (https://github.com/smartninja/wd1-py3-exercises/tree/master/lesson-10/guess-secret-number).*

Q&A

Any question?

If there's enough time after Q&A, students can start working on the homework from previous lessons that they haven't finished yet.

Working on exercises from the previous two lessons

Work on the homework you haven't finished yet. Upload all the exercises to GitHub (each exercise is its own repository).

> ***Instructor:*** *Make sure that everyone successfully uploads exercises to Git and that* `.gitignore` *is properly set up.*

Useful links

- Handling files (https://www.w3schools.com/python/python_file_handling.asp)