algorithmics

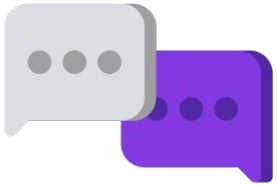
Module 3. Lesson 4.

The Smart Notes App P. 3

Link to the methodological guidelines

Discussion:

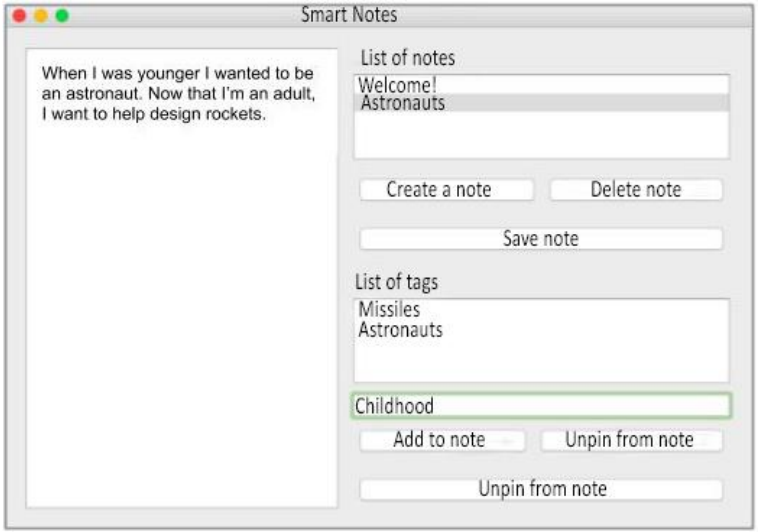
The optimal solution



Let’s keep working on the order!

The Scientific Institute of Theoretical Physics placed an order for the creation of a Smart Notes App.

Last time we:

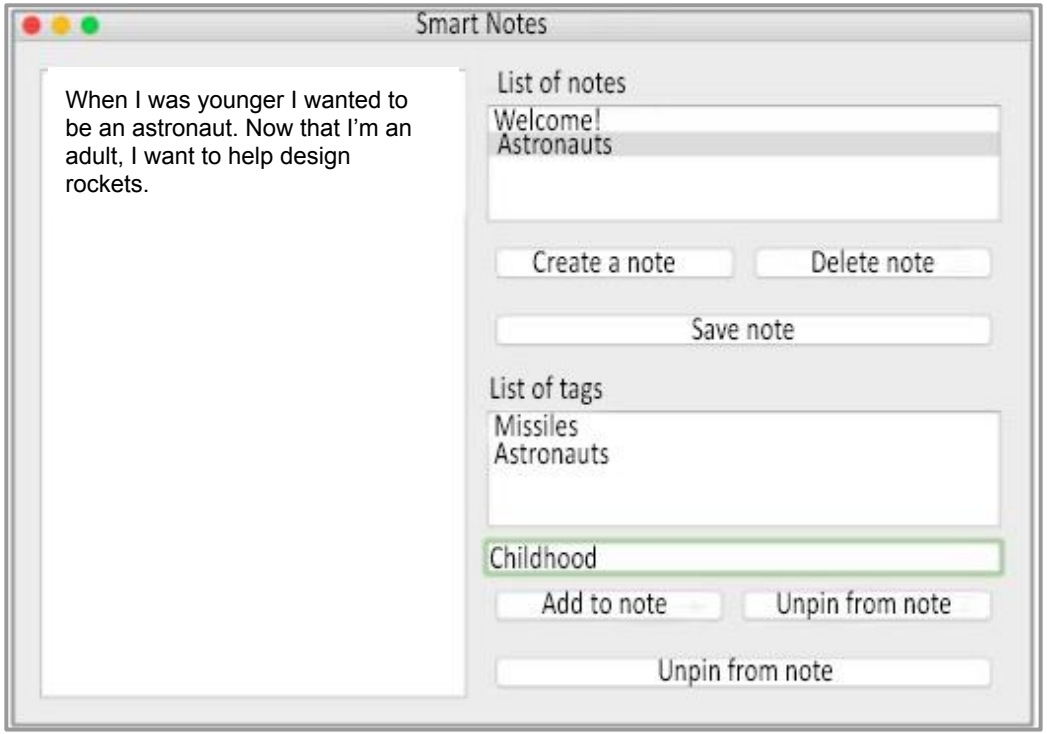


- Have **completed** the Smart Notes project.
- Have made the app’s interface fully active.
- Have **implemented** searching by tag.

Cole,
senior software developer

Discussion of tasks

Storing data for notes



Data for notes is stored in a **json file.**

After reading the data, the program works with the **dictionary of dictionaries.**

Discussion of tasks

Is this the optimal solution?

The customers have reviewed the preliminary solution.

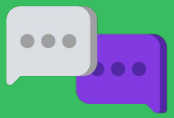
This letter of response has arrived:

> *Is this really the **most optimal solution** ?*
>
> *Sometimes I work on devices that don't have our app installed. Is it still possible to use these devices for editing the files with notes?*

Discussion of tasks

We can work with text files and json files.

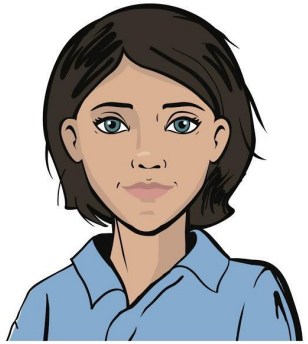
Now we can see if we've really chosen the optimal solution for everyday work.

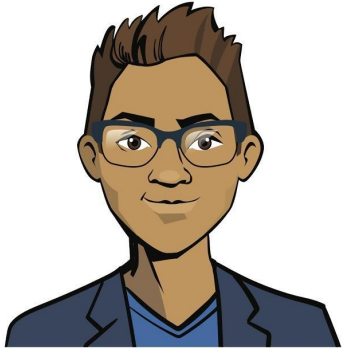


Discussion of tasks

To do this, today we'll need to discuss:

- Can it be optimal to store notes in text files?
- Is it **convenient** to work with notes in files? Is it possible to edit them without the app?
- **Is it better to store notes** in one text file or in several?

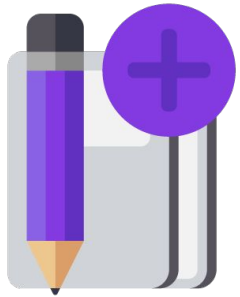




Discussion of tasks

Brainstorming:

Json files or text files?

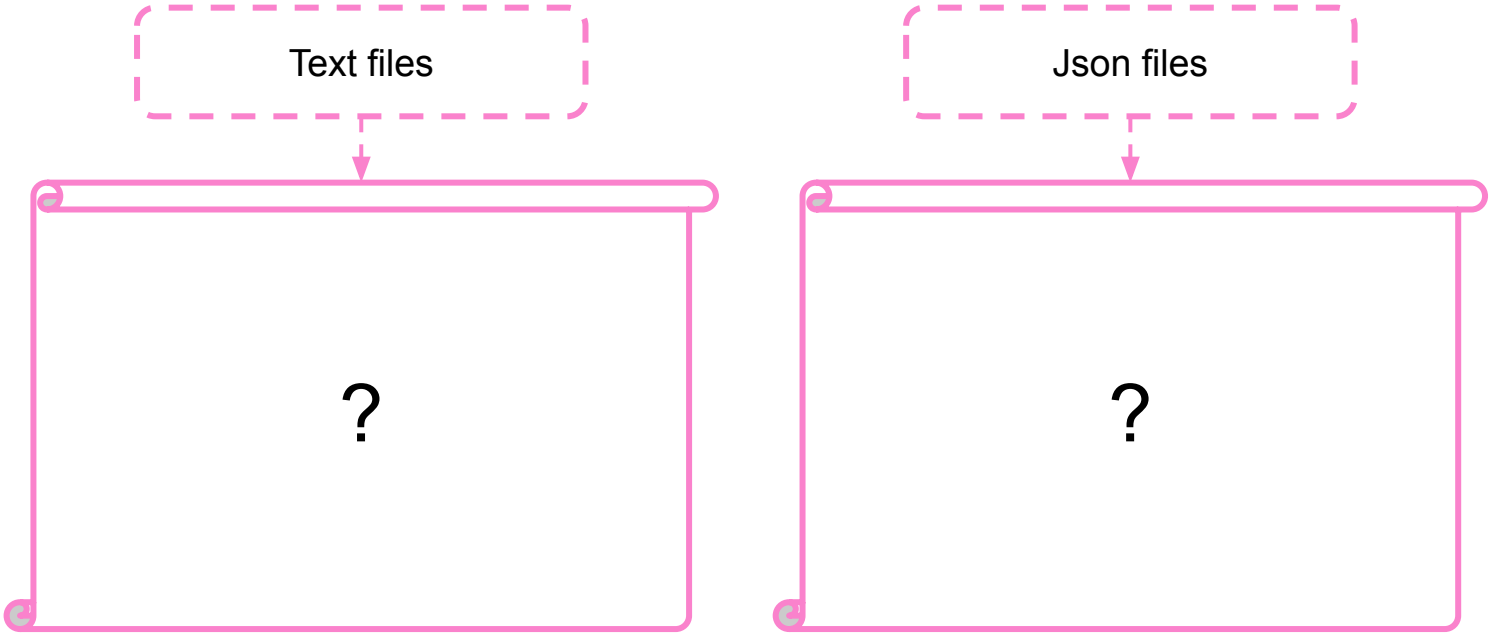


Is it possible to optimally store notes in text files?

Brainstorming

To answer this question, compare json files and text files

Let’s compare the structures inside different types of files



How is data presented in different types of files?

Brainstorming

Let’s compare the structures inside different types of files

Text files

Format: any

```
About the moon / Why is the moon a satellite and not a planet? / #moon #planet
```

data.txt

Json files

Format: Dictionaries or lists

```
{
"About the moon" : {
"text": "Why is the moon a satellite and not a planet?"
"tags": [ "moon",  "planet" ]}
}
```

data.json

Brainstorming

How to correctly read data from a text file?

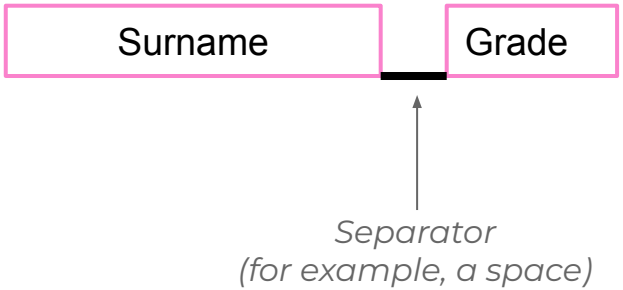
Brainstorming

Let's say that the file has the surnames of all of the students, as well as their grades.
When reading the file, how can we understand where the student's last name ends and their grade begins?

Reading data from a text file:

Different types should be separated from each other.

Let's say that the file has the surnames of all of the students, as well as their grades.
When reading the file, how can we understand where the student's last name ends and their grade begins?



Brainstorming

Separating data within a file:

Text files

The data is separated **according to the algorithm written by the user**.

Json files

The data is separated **automatically** according to the arrangement of the dictionary.

Brainstorming

Separating data within a file:

Text files

```
01.07_Children’s Day\n
04.11_Birthday\n
01.09_Day of Knowledge\n
```

*The date is separated from the holiday with an **underscore.***

*The units of data are located on **separate lines.***

Json files

```
{ "Ocean" :
{"tags": ["ocean", "sea"],
"text": "Can a large sea
become an ocean?"}}
```

Brainstorming

Problem:

The more data you have in a text file, the harder it will be to write a parser

Brainstorming

What if there are newline transitions in the text of the note?

What separator should we choose to make sure the user doesn't use it in a set of text or tags?

A parser is a program that picks out pieces of information in the data.

Brainstorming

You actually used a parser when you were working with **text files**.

- You selected lines in the text (1 line = 1 student)
- You split the strings into substrings (surname, name, grade).

Parsing happens automatically **in json files**.

Storing notes in text files:

Storing all of the notes
in one text file

Storage type
“One note, one file”

Which approach is more convenient?

Brainstorming

Storing notes in text files:

Brainstorming

Storing all **of the** notes
in one text file.

- The program only works with one text file.

↓

Simple for data recording, but difficult for parsing.

Storage type
“One note, one file”

- Notes can be created on a phone/laptop without the Smart Notes App, then simply copied to a file in the project folder.
- These notes are easy to send to others and print out.

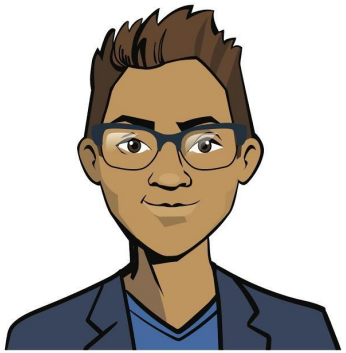
↓

Easy to collaborate with other devices and easy to parse.

Plan for finishing work on Smart Notes:

- Let’s see how well we know the basics of working with files *(first half of the workday)*.

- Let’s find a solution, program it, and compare it with the one we already have *(second half of the workday)*.



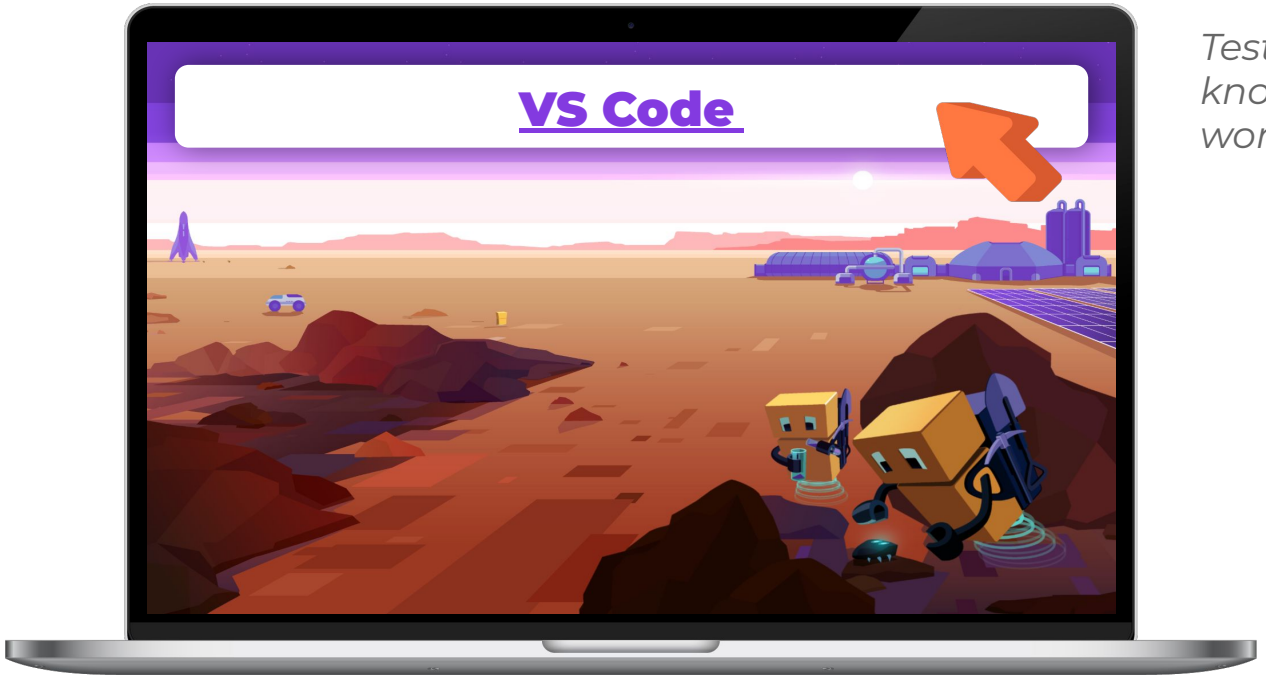
Brainstorming

Visual Studio Code:

The Smart Notes App

Qualification Test

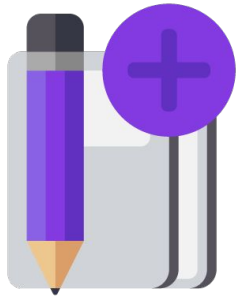
Test: Working with Files



Test your knowledge of working with files!

Brainstorming:

Smart Notes in Text Files.



Let’s see how the app changes when we switch to text files.

Review
of the new topic

Necessary changes:

Data storage	App functionality	Interface
How is data storage organized using the “one note, one file” method?	First and foremost, what part of the app needs to be changed?	Does anything need to be changed in the interface?

Review of the new topic

Necessary changes:

Data storage

For the starting note, create a file named 0.txt. The rest of the files will be named 1.txt, 2.txt, and so on.

Every data item (name, text, tags) will be written on a new line.

App functionality

What needs to be changed is the creation of new notes and the storage of notes in a file.

The rest of the functions either work with the ready-made structures or work in a similar way.

Interface

Remains the same.

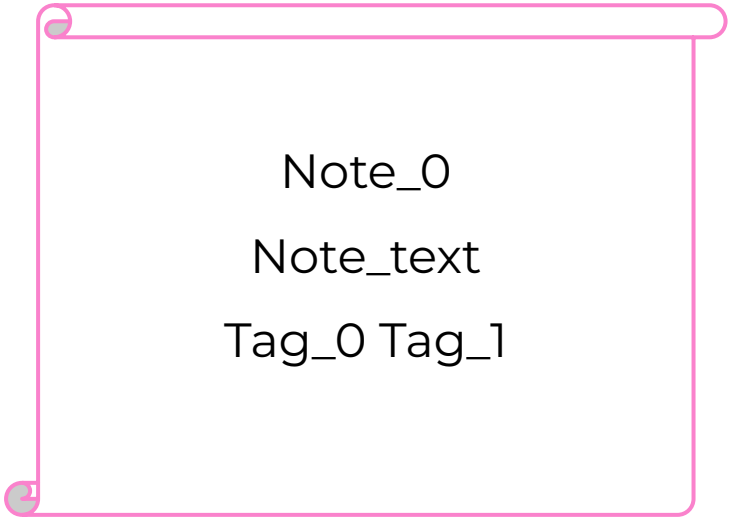
1.

2.

Review of the new topic

1. Data Storage

To change the data storage, create a starting note, and read it.



0.txt

How do I read the data?

Review of the new topic

1. Data Storage

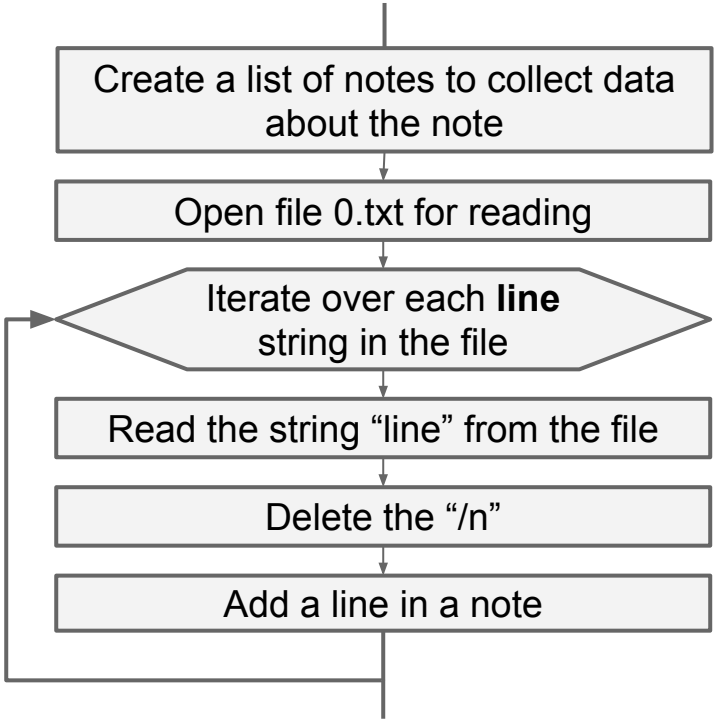
To change the data storage, create a starting note, and read it.

Note_0

Note_text

Tag_0 Tag_1

0.txt



Review of the new topic

1. Data Storage

To change the data storage, create a starting note, and read it.

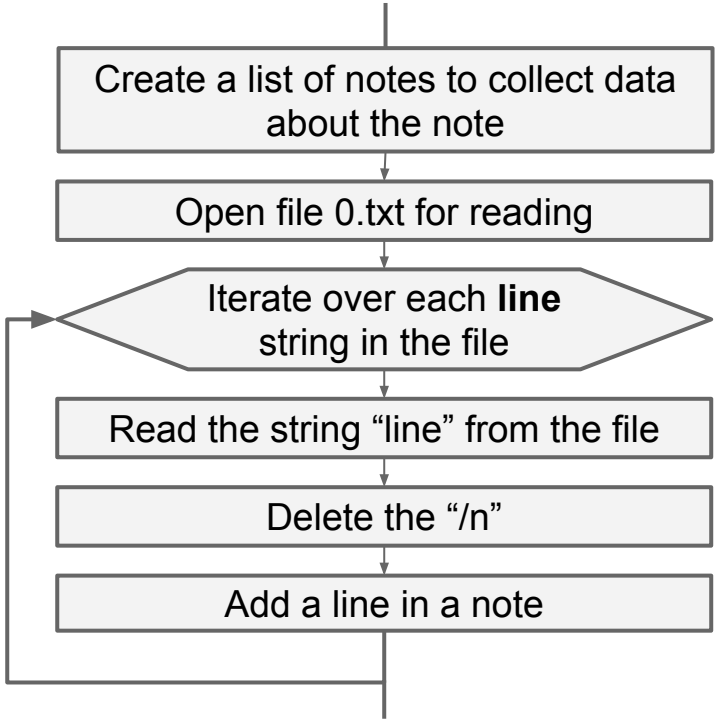
note =

[Note_0 ,

Note_text ,

Tag_0 Tag_1]

The tags need to be separated. How?



Review of the new topic

1. Data Storage

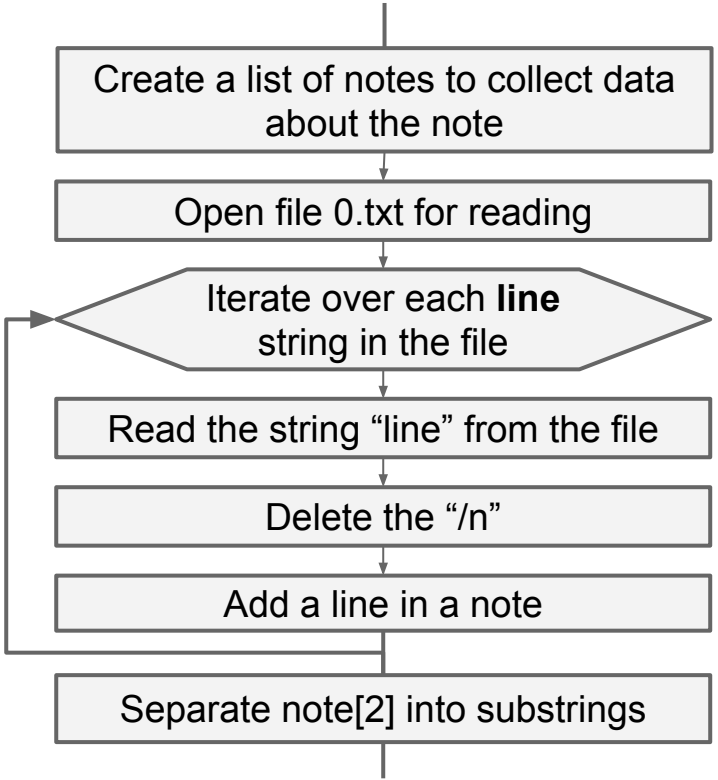
To change the data storage, create a starting note, and read it.

note =

[Note_0 ,

Note_text ,

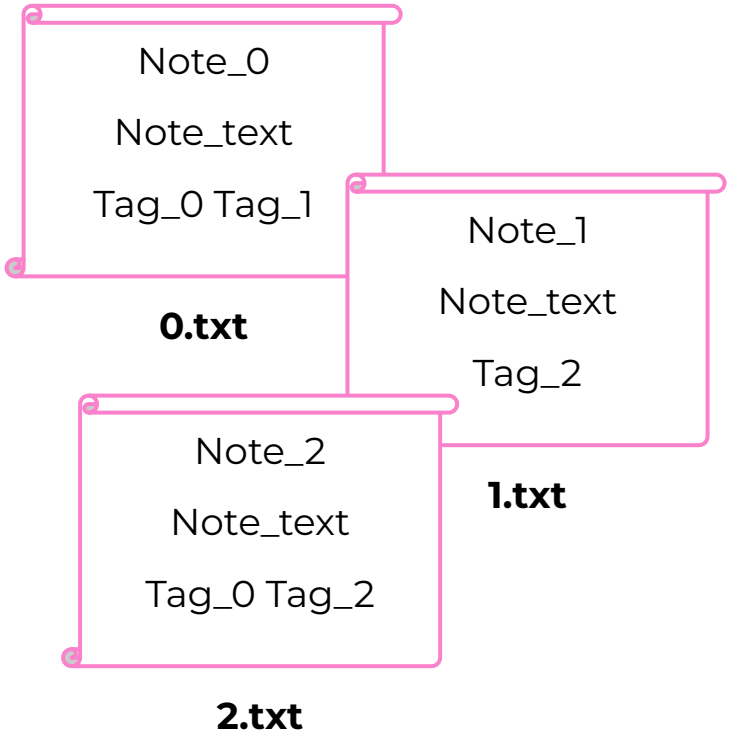
Tag_0,Tag_1]



Review of the new topic

1. Data Storage

How do I read data from several different note files?



How do I read the data?

Review of the new topic

1. Data Storage

The current name of the file will be **`str(num_name)+".txt"`**

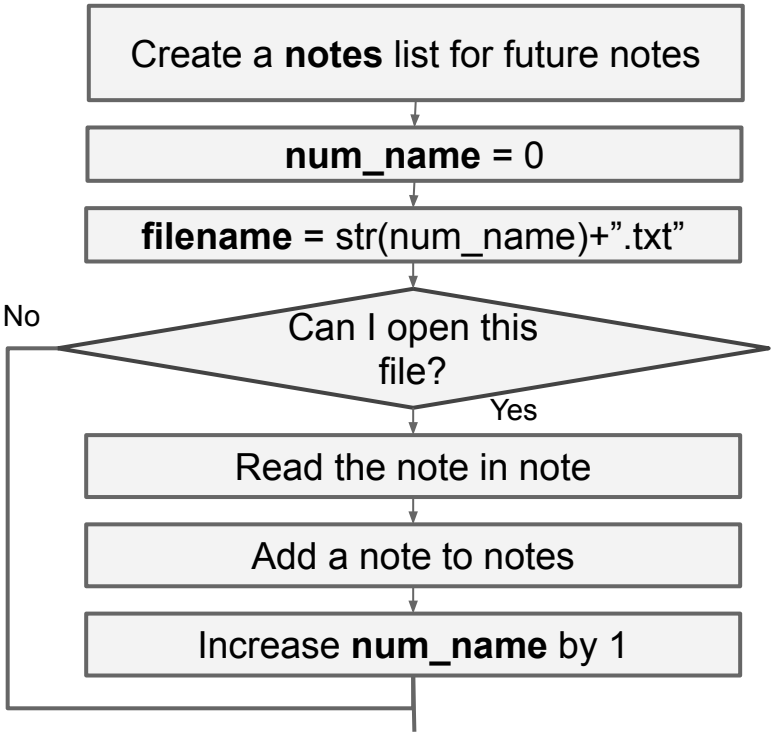
Note_0

Note_text

Tag_0 Tag_1

0.txt

`num_name = 0`



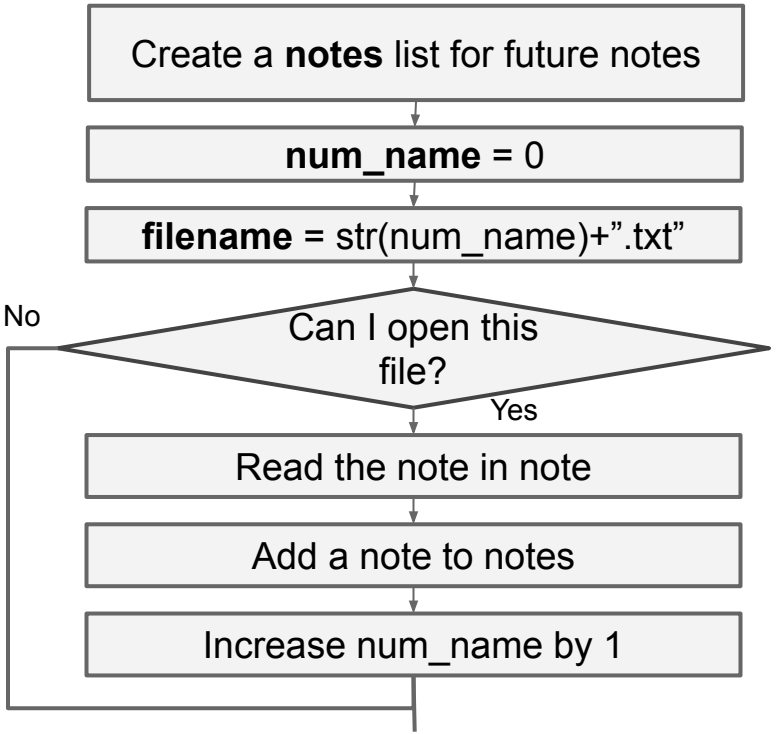
Review of the new topic

1. Data Storage

The current name of the file will be **str(num_name)+“.txt”**

```
while True:
    filename = str(name)+".txt"
    try:
        with open(filename, "r") as file:
            #reading and adding
            #notes in notes
    except IOError:
        break
```

Error reading the file (for example, the indicated file does not exist)



Review of the new topic

2. App functionality

Let’s just figure out how to create and save notes.

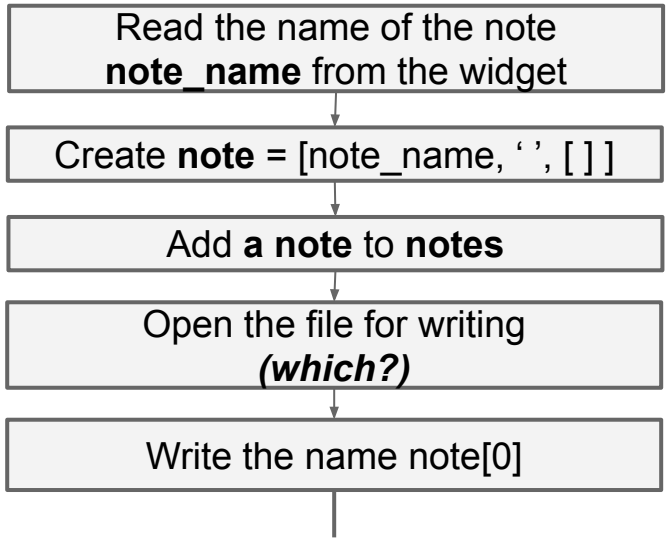
How do I create a new note
and write it to the file?

Review
of the new topic

2. App functionality

Let’s just figure out how to **create** and save notes.

How do I create a new note and write it to the file?

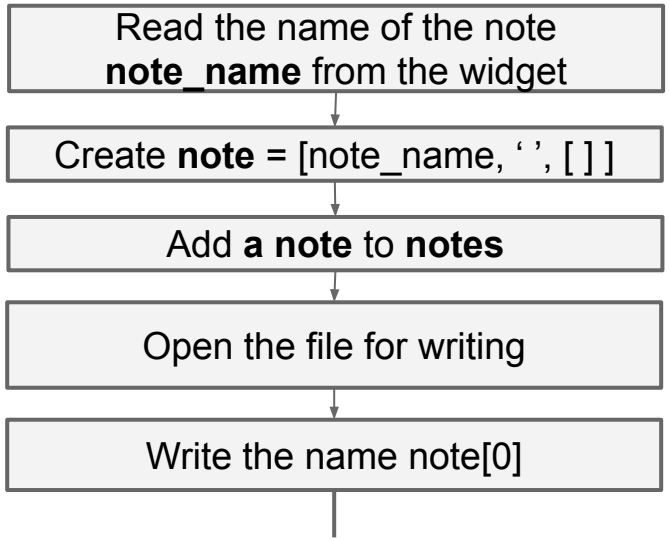


Review of the new topic

2. App functionality

Let's just figure out how to **create** and save notes.

```
def add_note():
#reading the note name note_name
if ok and note_name != "":
    note = list()
    note = [note_name, '', []]
    notes.append(note)
    list_notes.addItem(note[0])
    filename = str(len(notes)-1)+".txt"
    with open(filename, "w") as file:
        file.write(note[0]+'\n')
```



Review of the new topic

2. App functionality

Let’s just figure out how to create and **save** notes.

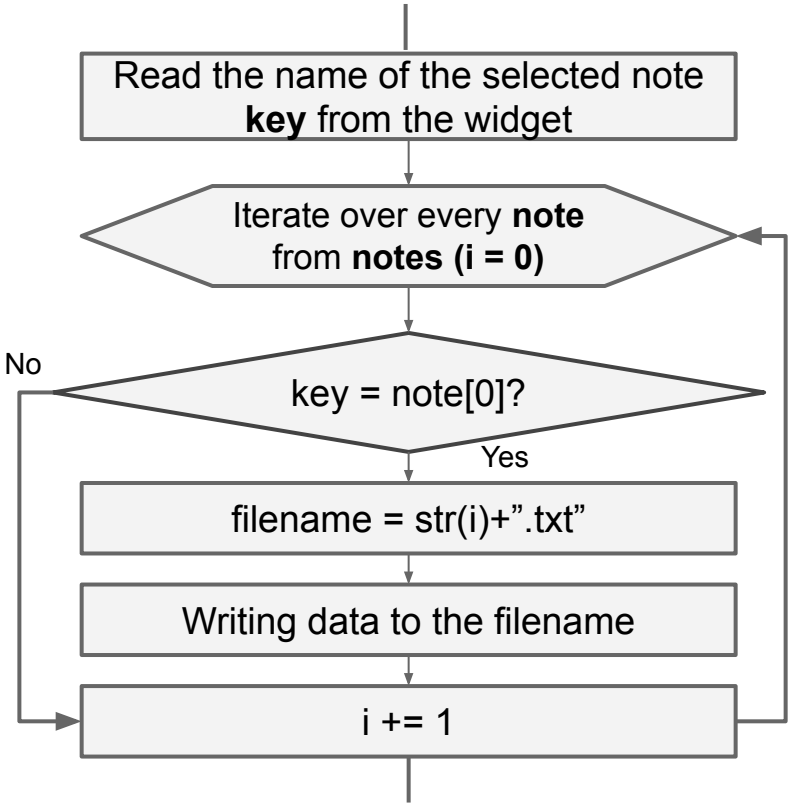
How do I store data for a note in a file?

Review of the new topic

2. App functionality

Let’s just figure out how to create and **save** notes.

How do I store data for a note in a file?

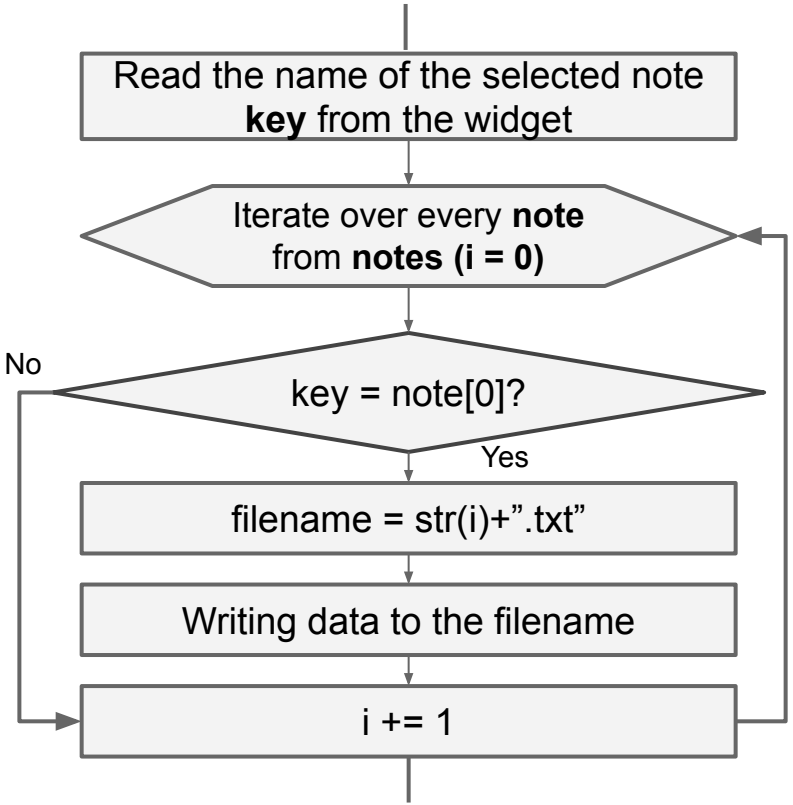


Review of the new topic

2. App functionality

Let’s just figure out how to create and **save** notes.

```
key = list_notes.selectedItems()[0].text()
i = 0
for note in notes:
    if note[0] == key:
        note[1] = field_text.toPlainText()
        filename = str(i)+".txt"
        with open(filename, "w") as file:
            file.write(note[0]+'\n')
            file.write(note[1]+'\n')
            for tag in note[2]:
                file.write(tag+' ')
            file.write('\n')
    i += 1
```



Review of the new topic

Tasks:

- Create an alternative demo version of Smart Notes with data stored in text files.
- Analyze which data storage is optimal for this task.

Attention! Use the following command to make sure that the txt-file data is correctly displayed on computers with different operating systems:

```
with open(filename, "r", encoding='utf-8') as file:
```

Review of the new topic

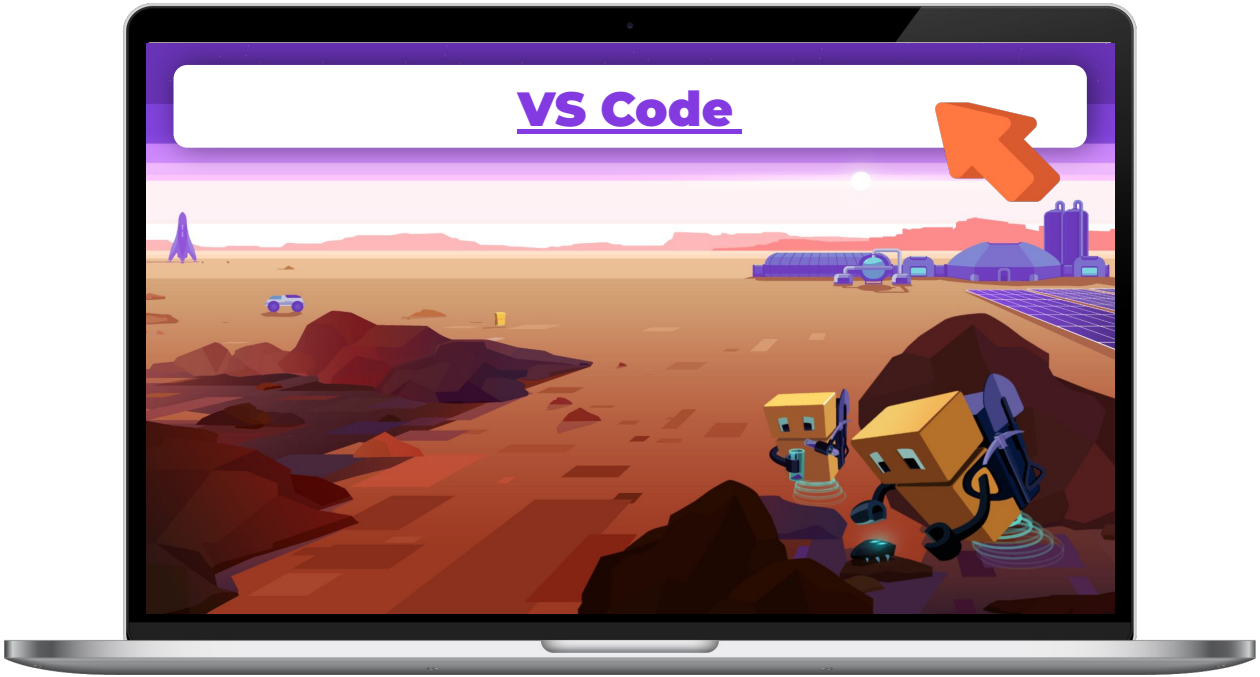
Break

Visual Studio Code:

The Smart Notes App

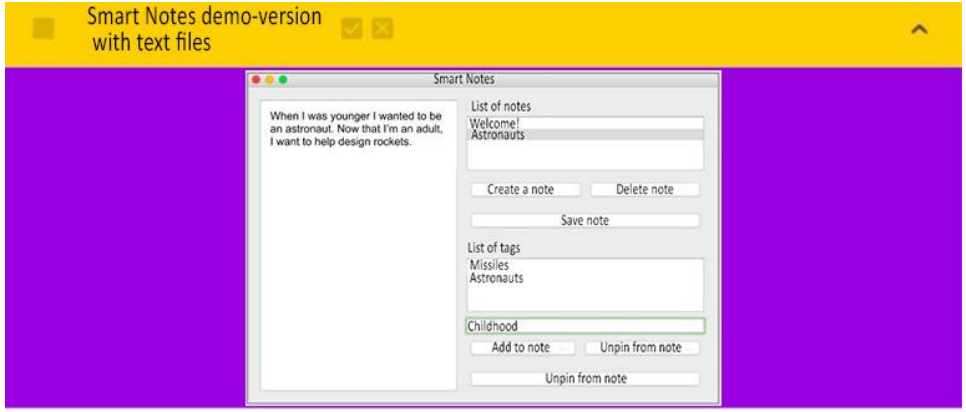
Complete tasks in VS Code

➡ “VSC: Smart Notes Analysis”



Complete tasks in VS Code

“VSC: Smart Notes Analysis”



Smart Notes demo-version with text files

Program a demo-version of the app using text files. You need to:

1) Create a starting note named 0.txt with data (File-> New...). Read the note and display the data in a widget
2) Program note creation and storage. If necessary, use the hint.

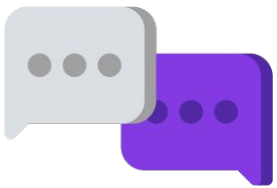
Which note storage do you think is the most optimal? Why?

Program a demo version of the app using text files.

Which storage is the most optimal?

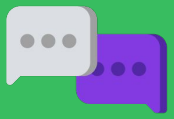
Discussion:

Optimal data storage



Which way of storing notes is the most convenient?

What kind of data is suitable for storing in text files?



Discussion of tasks

Note structures within different kinds of files

Text files

Format:
One string = One piece of data

```
About the moon
Why is the moon a satellite?
#moon #planet
```

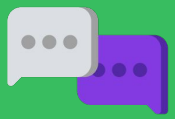
0.txt

Json files

Format: a dictionary of dictionaries

```
{
"About the moon" : {
"text": "Why is the moon a
satellite and not a planet?"
"tags": [ "moon",  "planet" ]}
}
```

data.json



Discussion of tasks

Conclusions:

Text files

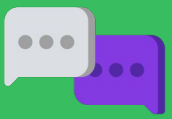
Convenient for working with simple or non-standard data.

Examples of suitable data: ?

Json files

Convenient for working with structured data.

Examples of suitable data: ?



Discussion of tasks

Conclusions:

Text files

Convenient for working with simple data. Files are easy to open on other devices.

Examples:

- **List of surnames** (simple data that can be read with one command)
- **Book text** for analysis (unstructured data where every word needs to be analyzed)

Json files

Convenient for working with structured data.

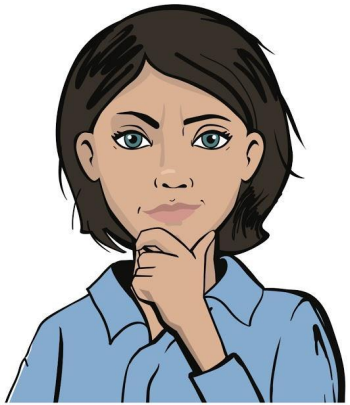
Examples:

- **Data from a questionnaire** (all of the data is distributed into fields).
- **Database for a grocery stockroom** (all of the products have the same properties: quantity, expiration date, etc.).

Discussion of tasks

Wrapping up the work day

List all of the files that you now know how to work with .



Wrapping up the work day

We can use:

- Files with **.py** programs and modules
- **.txt** Text files
- Text files with a **.json** structure

\+ We can use .jpg/.png

We can use:

- Files with **.py** programs and modules
- **.txt** Text files
- Text files with a **.json** structure

\+ We can use .jpg/.png

*We can **use and change** data stored in these files.*

*We can **use** a ready-made file without changing it.*

Wrapping up the work day

We can use:

- Files with **.py** programs and modules
- **.txt** Text files
- Text files with a **.json** structure
+ We can use .jpg/.png

*We can **use and change** data stored in these files.*

*We can **use** a ready-made file without changing it.*

Actually, we can also analyze and change graphic files using Python.
We will cover this in the next module.

Wrapping up the work day