algorithmics

Module 3. Lesson 2.

The Smart Notes application

Link to guidelines

Discussion:

The Smart Notes application

Let's get back to the request!

The theoretical research institute has turned to us with a request for a Smart Notes application.

The scientists should be able to:

- Create and delete notes.
- Edit notes.
- Add tags to notes.
- Search the notes using tags.

Ready to get started?

Cole,
senior developer

Discussing work tasks

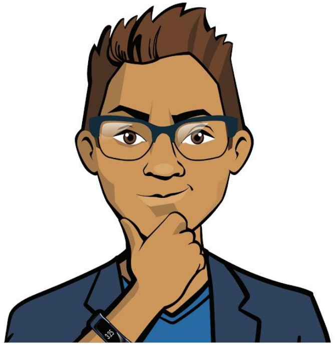
Last time we resolved two important issues

- *How do we organize the storage of these notes?*

  We need to program **long-term storage** of information! For example, we can use text files.

- *How do we program the appearance of the program?*

  Of course, using **PyQT**!

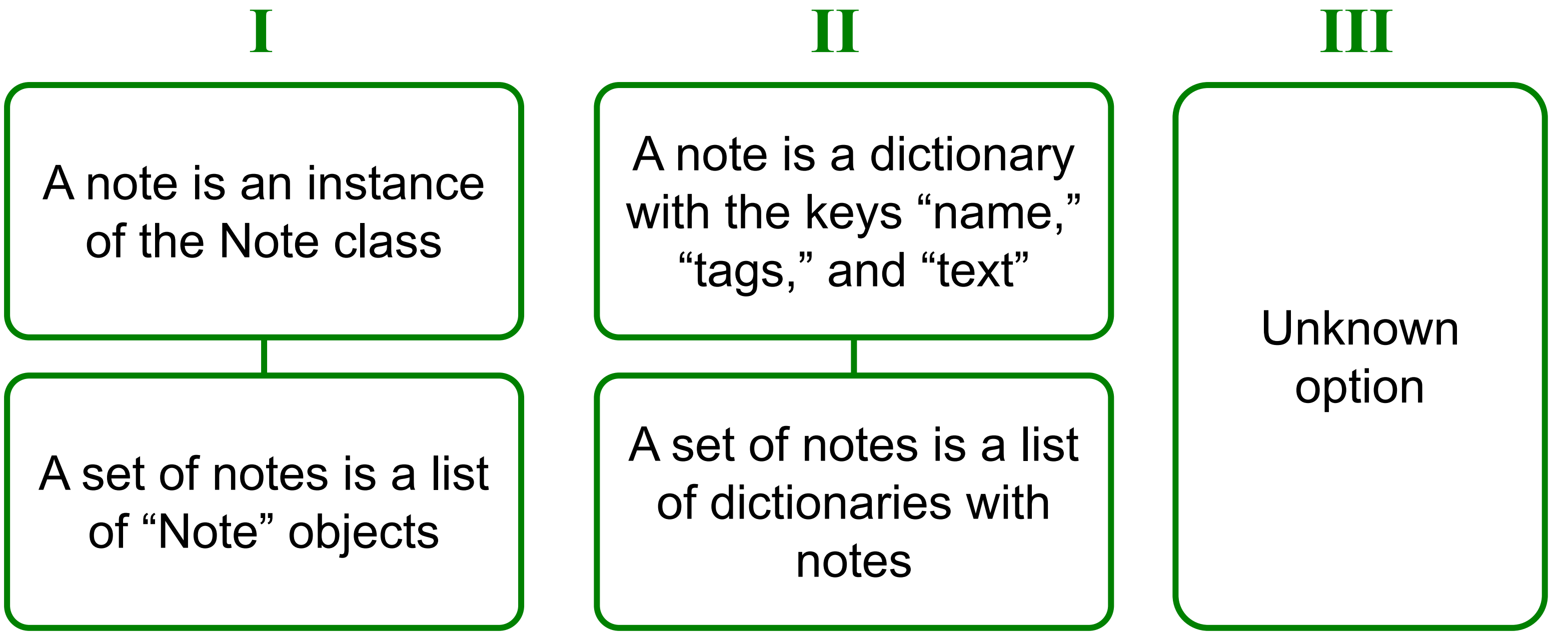


Discussing work tasks

How do we read notes from a file and use them in a program?

Discussion:
Smart Notes

How do we read notes from a file and use them in a program?



I

A note is an instance of the Note class

A set of notes is a list of "Note" objects

II

A note is a dictionary with the keys "name," "tags," and "text"

A set of notes is a list of dictionaries with notes

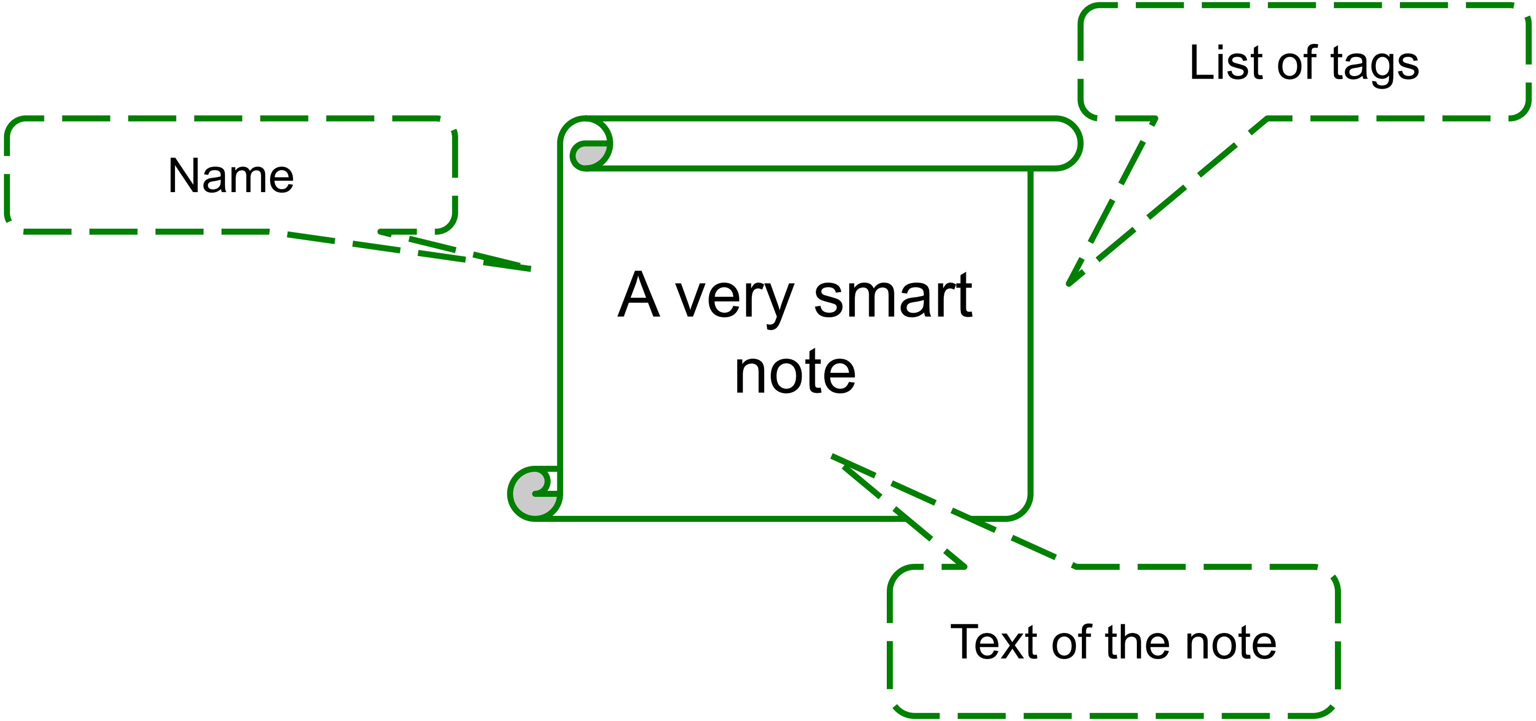
III

Unknown option

Possible options for presenting a set of notes

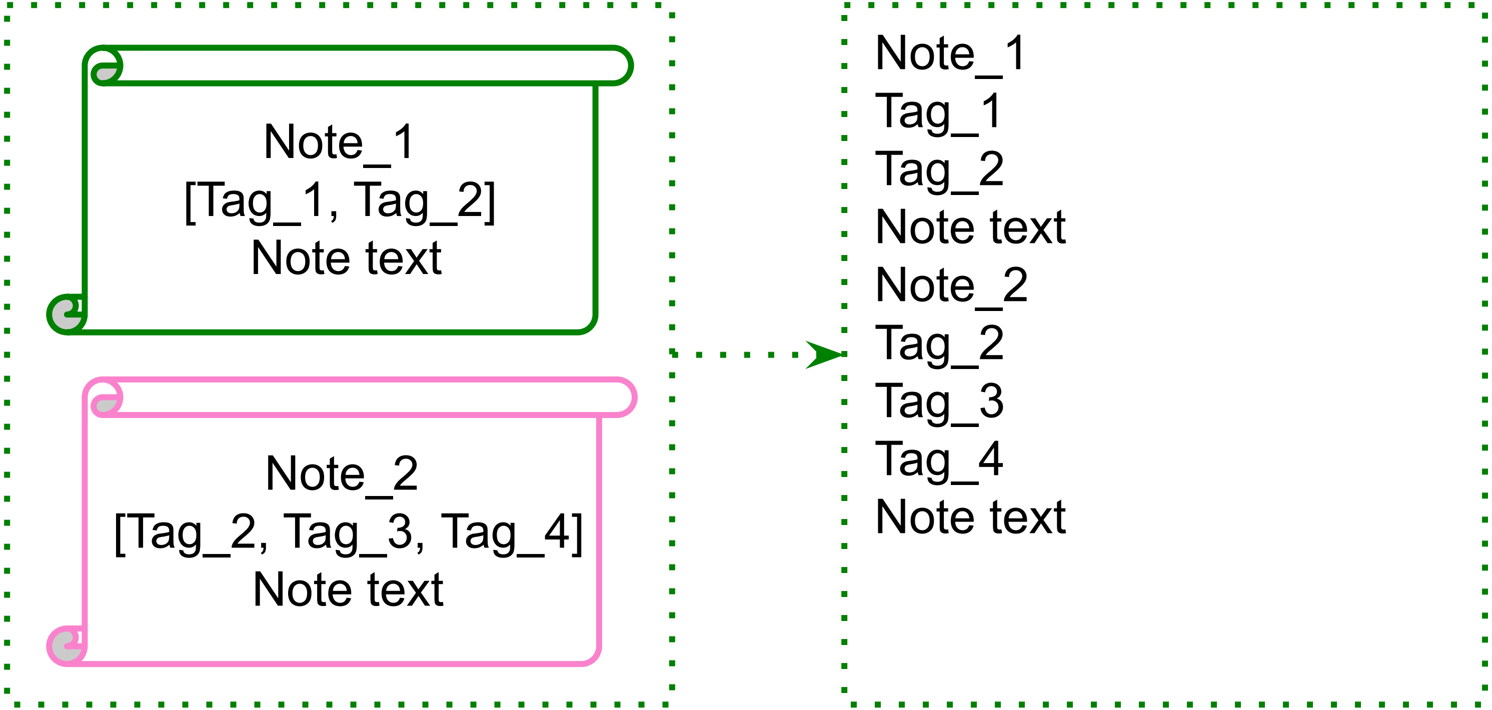
Discussion:
Smart Notes

If it is arranged as a **single note** :



Discussion: Smart Notes

Then the arrangement of a **file with notes** :



Note_1
[Tag_1, Tag_2]
Note text

Note_2
[Tag_2, Tag_3, Tag_4]
Note text

Note_1
Tag_1
Tag_2
Note text
Note_2
Tag_2
Tag_3
Tag_4
Note text

Possible arrangement for a file with notes.

Then the arrangement of a **file with notes** :

And the number of tags may be different.

Any note can be deleted by the user.

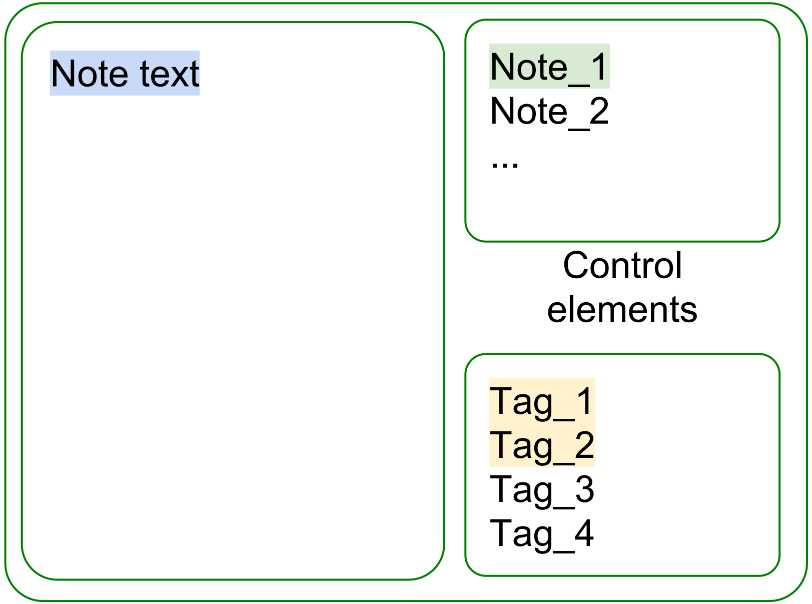
Tags for a specific note can also be added and deleted.

```
Note_1
Tag_1
Tag_2
Note text
Note_2
Tag_2
Tag_3
Tag_4
Note text
```

Possible arrangement for a file with notes.

Discussion:
Smart Notes

The file data should also be displayed in application widgets:

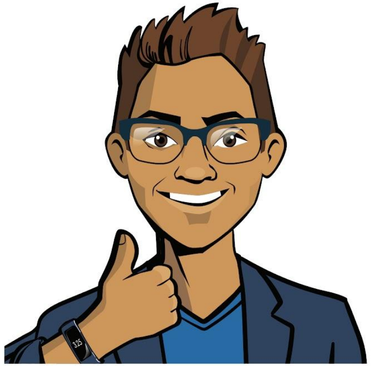


Note_1
Tag_1
Tag_2
Note text
Note_2
Tag_2
Tag_3
Tag_4
Note text

The structure is quite complicated to work with!

Professional developer recommendations:

Optimize your productivity by using special files with predefined data structures!



III

Unknown option

—

Json files

Discussion: Smart Notes

The goal of the work day is

to program the application interface and arrange the storage of notes in a json file.

Today you will :

- Learn how a json file works — a file with a predefined data structure.
- Program the application interface.
- Upload your first smart note.

Discussing work tasks

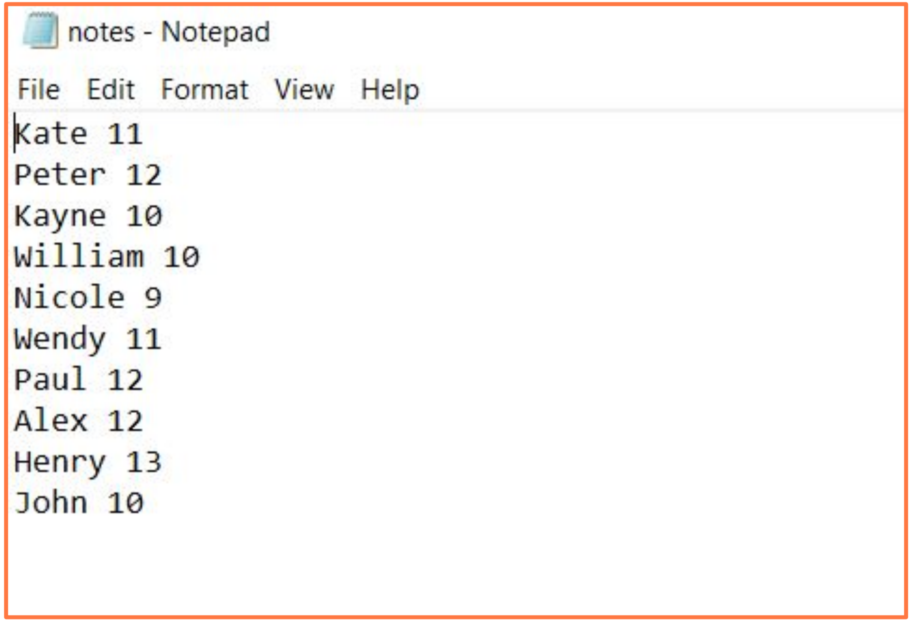
Qualifications

Demonstrate your knowledge of the PyQt library and working with text files

Qualifications

Where and how can I arrange long-term data storage ?

For example, in a text file:



```
Kate 11
Peter 12
Kayne 10
William 10
Nicole 9
Wendy 11
Paul 12
Alex 12
Henry 13
John 10
```

Which access attributes do you know for a file with data?

Qualifications

File access attributes :

Purpose of the function	*Function in Python*
Open file for reading	`open(“notes.txt”, “r”)`
Open file for writing	`open(“notes.txt”, “w”)`
Open file for appending	`open(“notes.txt”, “a”)`

Qualifications

What construction will open a notes.txt file for reading data?

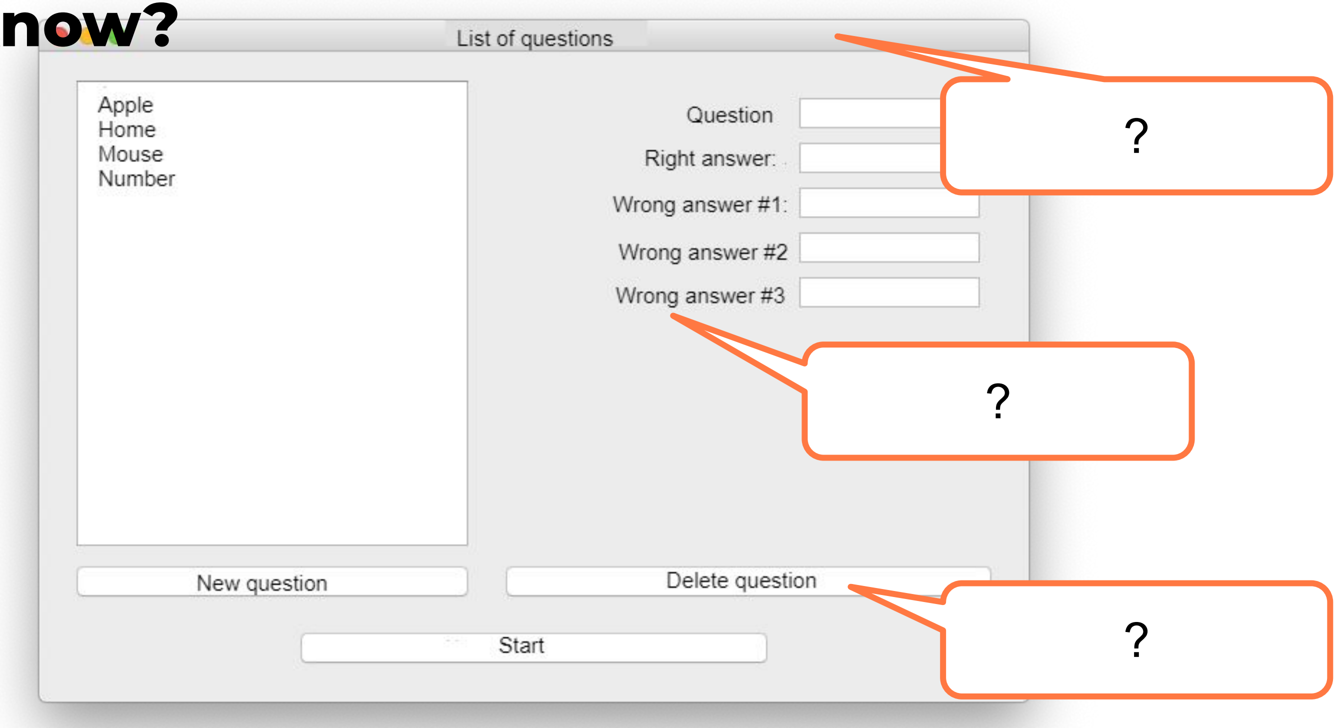
Qualifications

The construction will open a notes.txt file for reading data:

Purpose of the function	*Function in Python*
Open file for reading	`with open(“notes.txt”, “r”) as file:`
Reading file data	`data = file.read()`
Close file when finished	The file will automatically close after the end of the with operator block

Qualifications

How do we create an application window in PyQT?
What widgets for this window do you know?



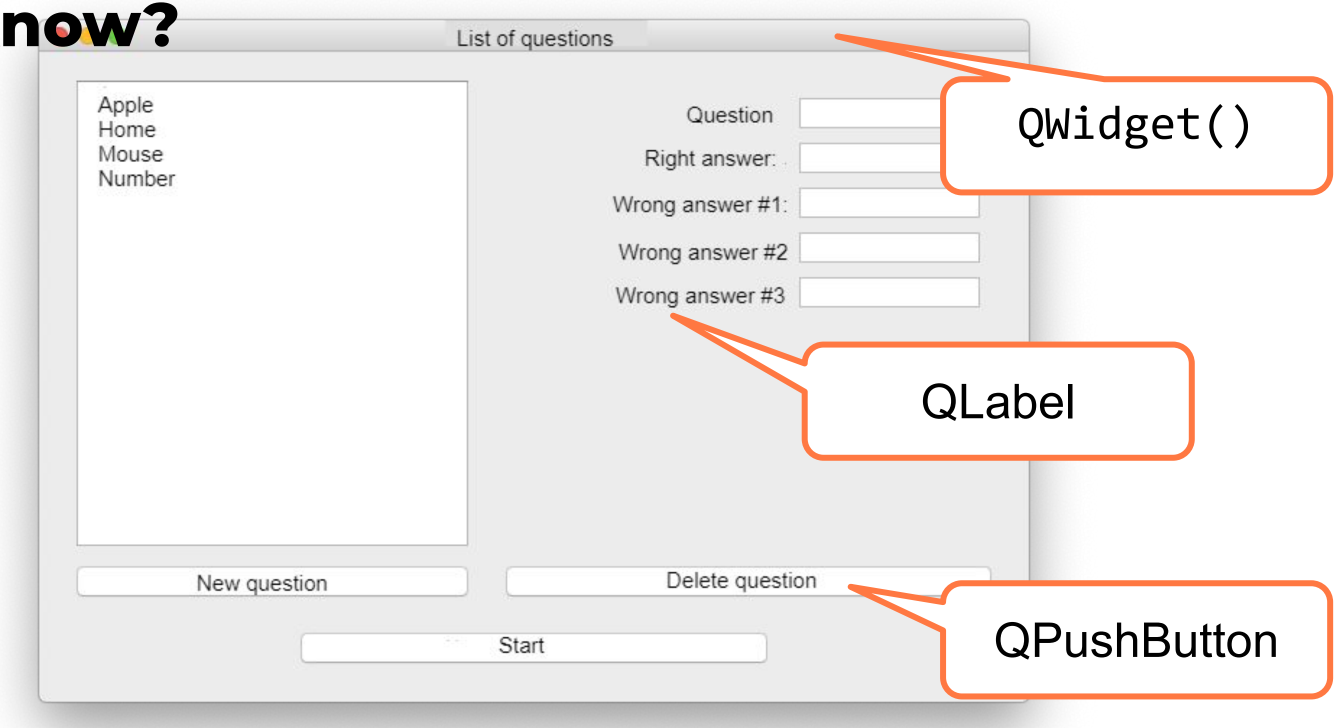
List of questions

Apple
Home
Mouse
Number

Question
Right answer:
Wrong answer #1:
Wrong answer #2
Wrong answer #3

Qualifications

How do we create an application window in PyQT?
What widgets for this window do you know?



List of questions

Apple
Home
Mouse
Number

Question
Right answer:
Wrong answer #1:
Wrong answer #2
Wrong answer #3

New question
Delete question
Start

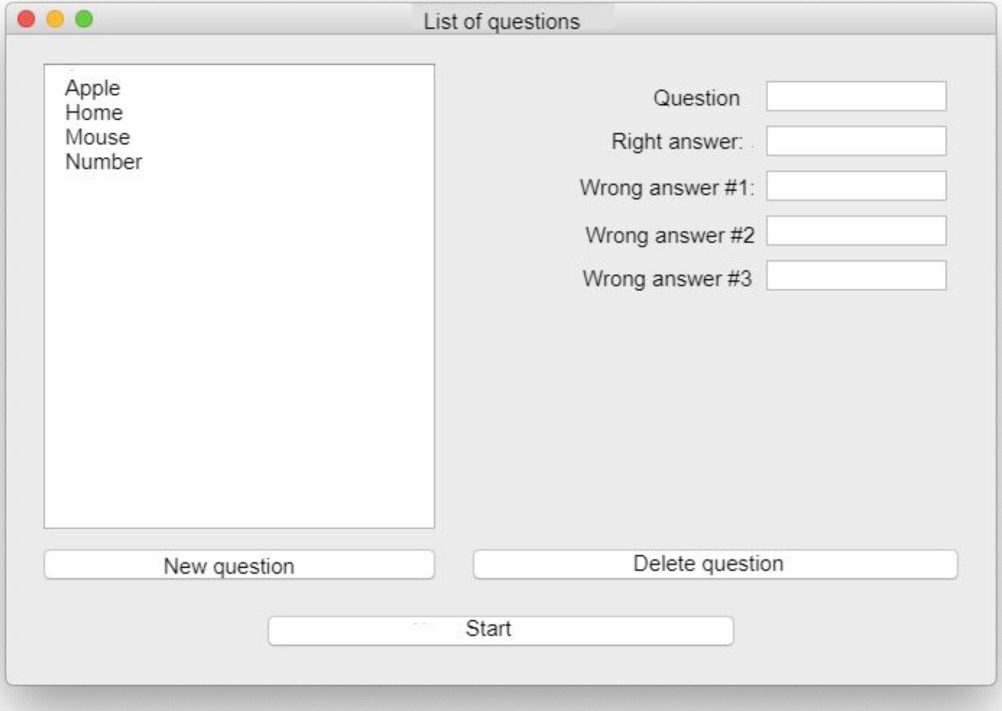
QWidget()

QLabel

QPushButton

Qualifications

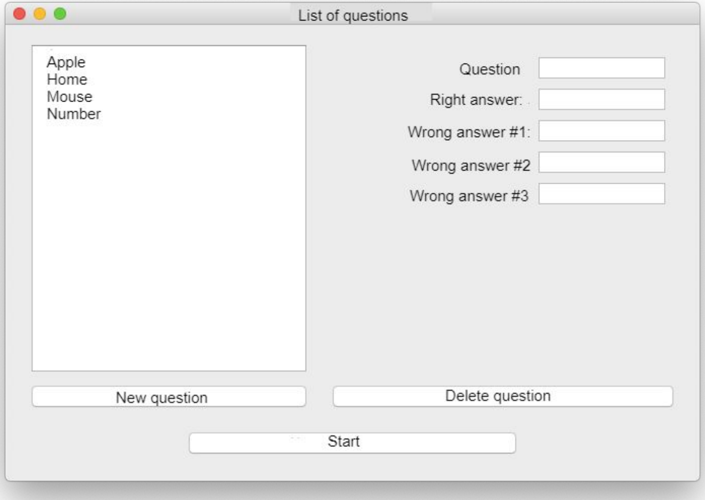
What is a layout? What does it consist of?



Apple
Home
Mouse
Number

Qualifications

What is a layout? What does it consist of?

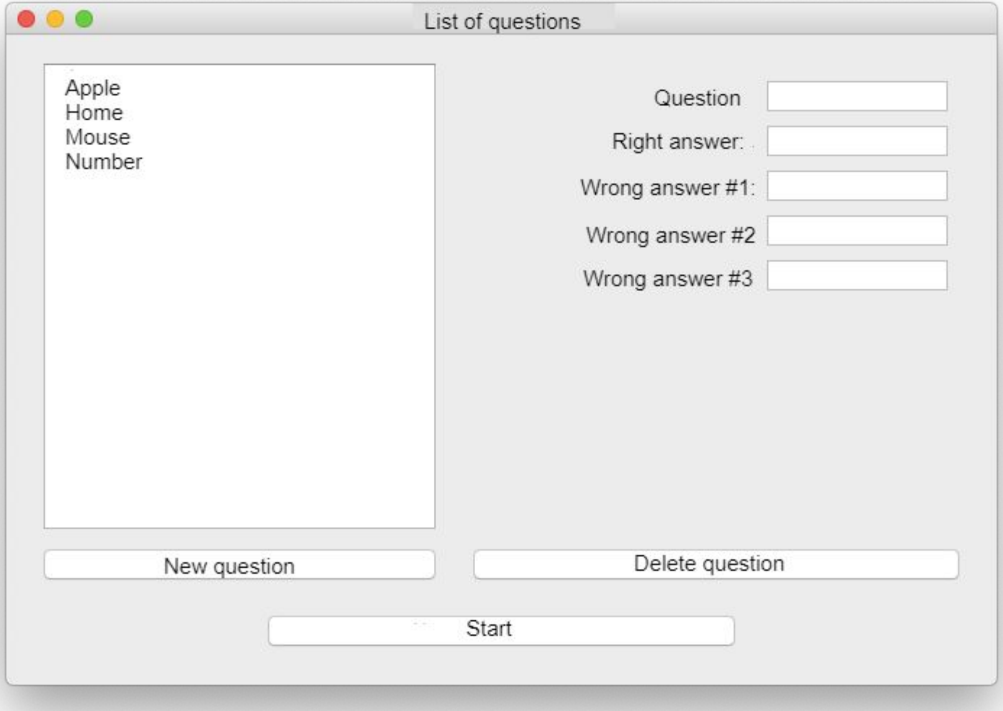


List of questions

Apple
Home
Mouse
Number

A layout is an interface element with which widgets can be arranged along lines.

Name and show the guide lines in this window:



List of questions

Apple
Home
Mouse
Number

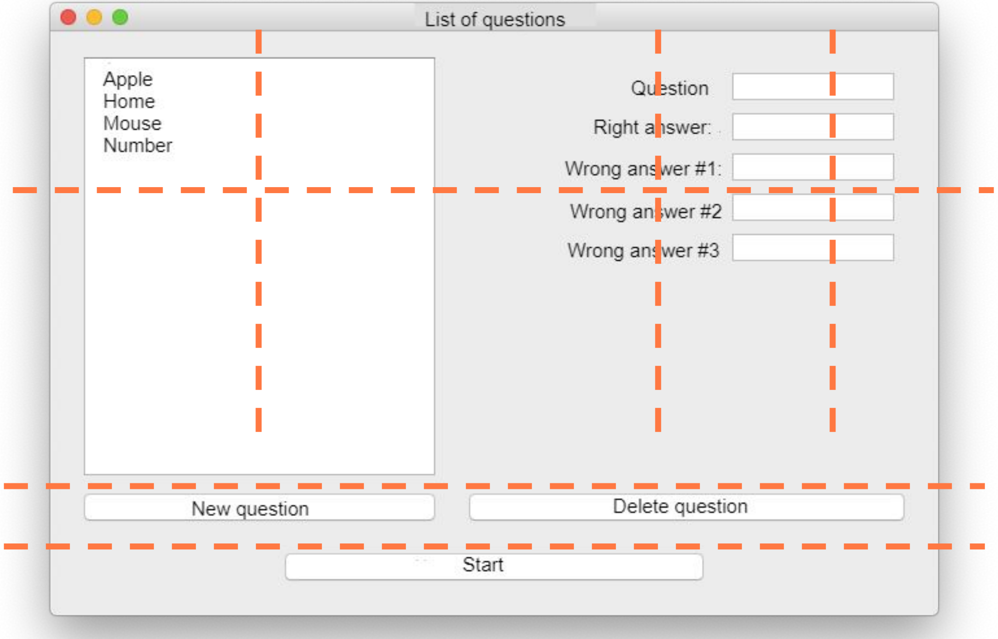
Question
Right answer:
Wrong answer #1:
Wrong answer #2
Wrong answer #3

There may be several options!

Qualifications

Possible answer:

There may be several options!



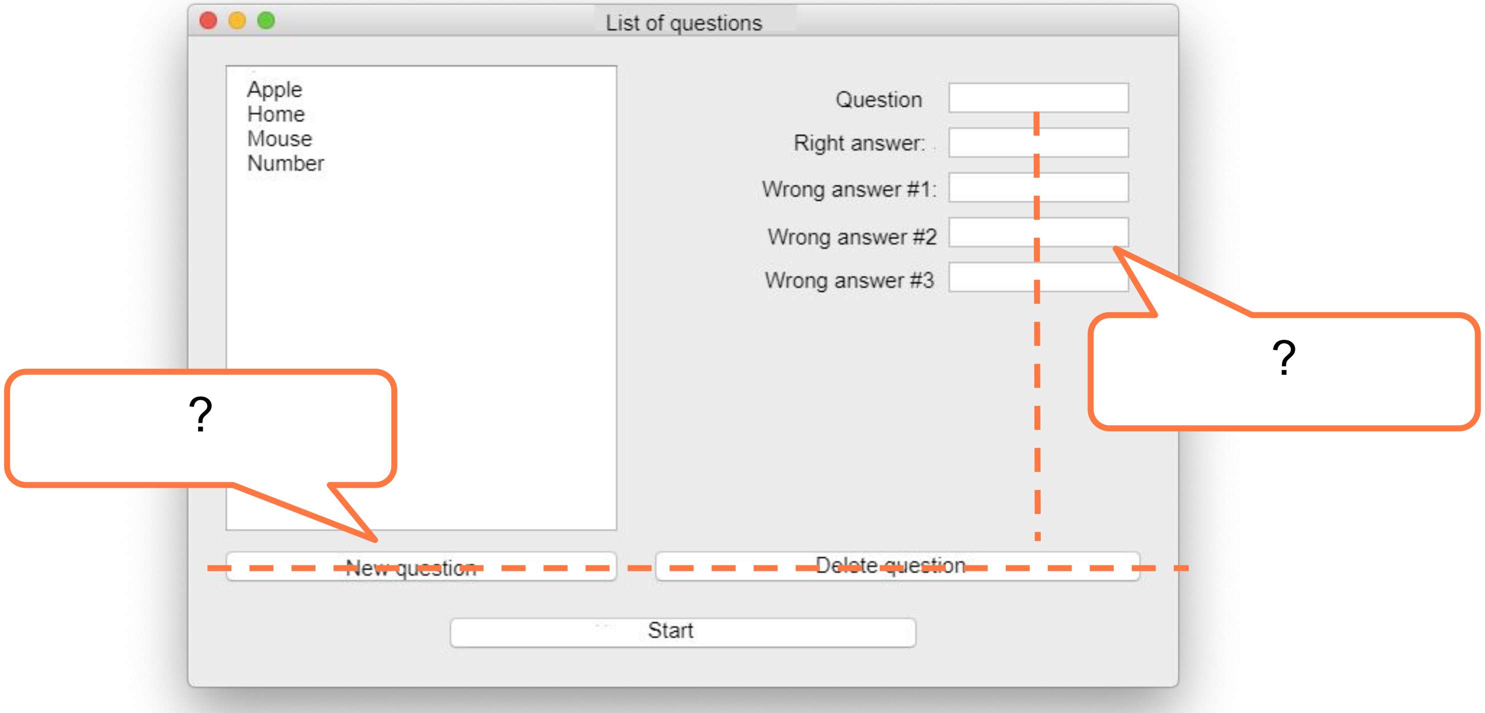
List of questions

Apple
Home
Mouse
Number

Question
Right answer:
Wrong answer #1:
Wrong answer #2
Wrong answer #3

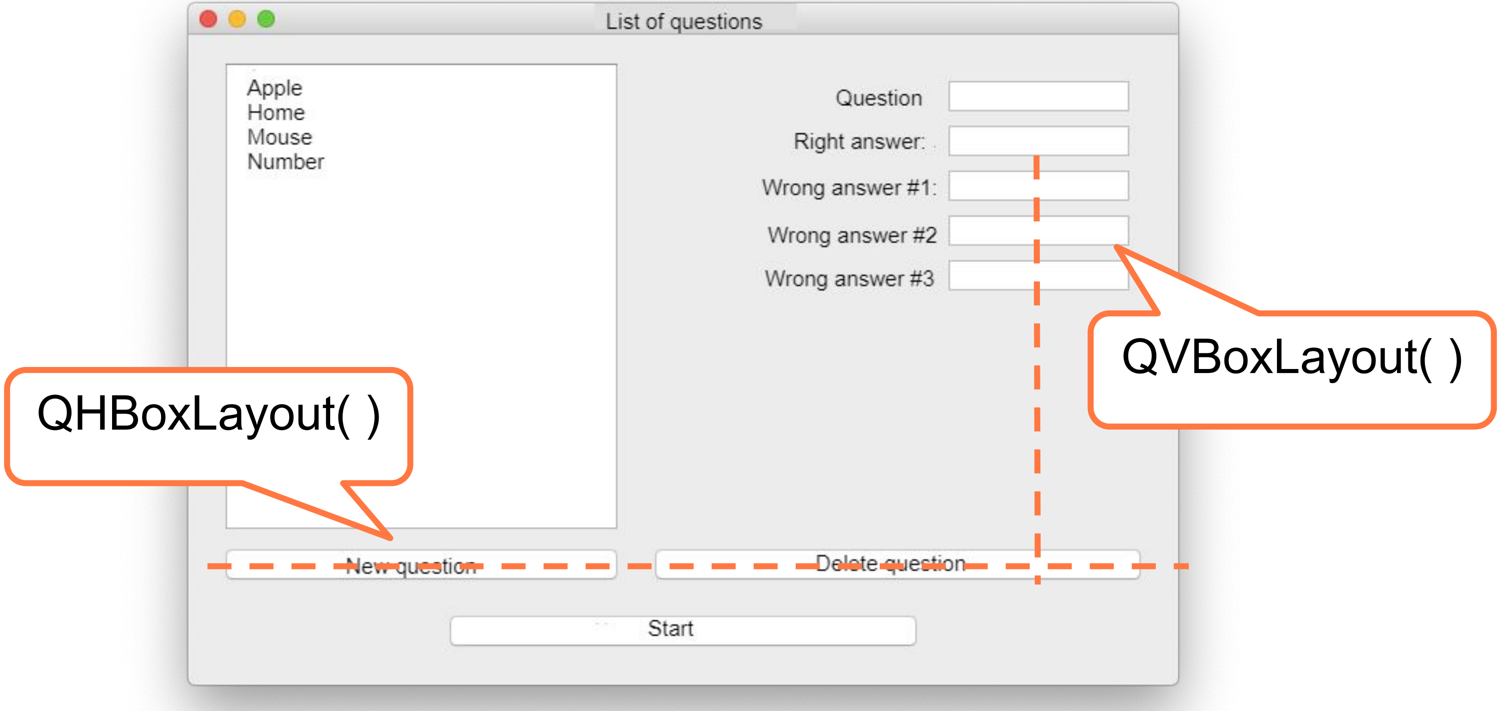
Qualifications

Which command creates these lines?



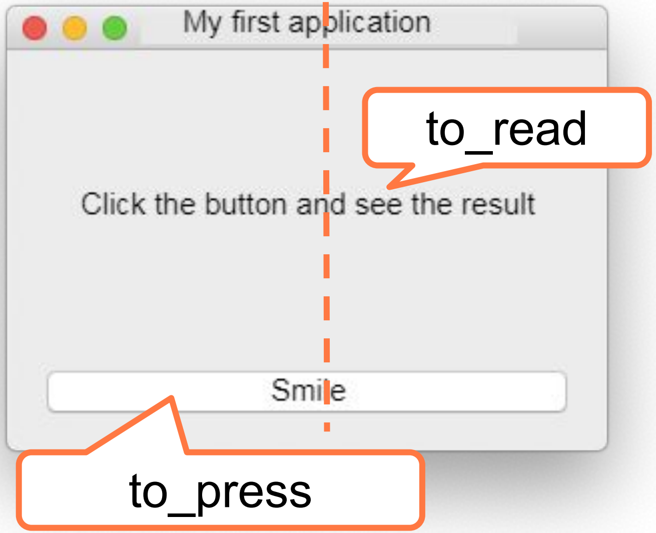
Qualifications

Which command creates these lines?



Qualifications

How do we arrange the widgets and run the application?



My first application

Click the button and see the result

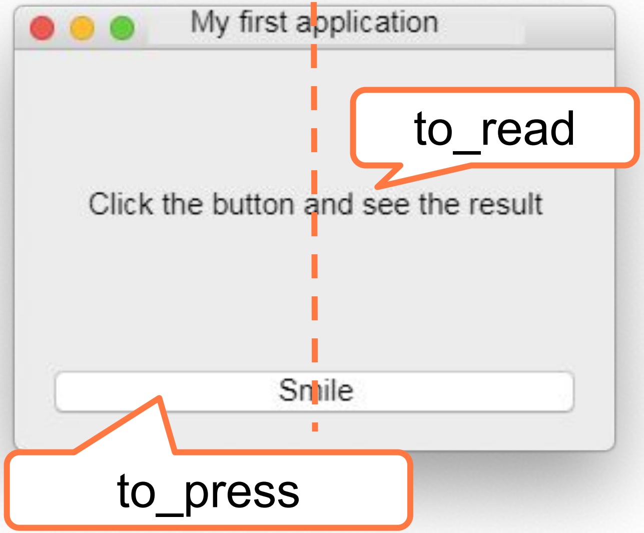
```
#which layouts do we create?




#how do we display the
application?
```

Qualifications

How do we arrange the widgets and run the application?



```
#which layouts do we create?
col = QVBoxLayout()
col.addWidget(to_read)
col.addWidget(to_press)
main_win.setLayout(col)


#how do we display the application?
main_win.show()
app.exec_()
```

Qualifications

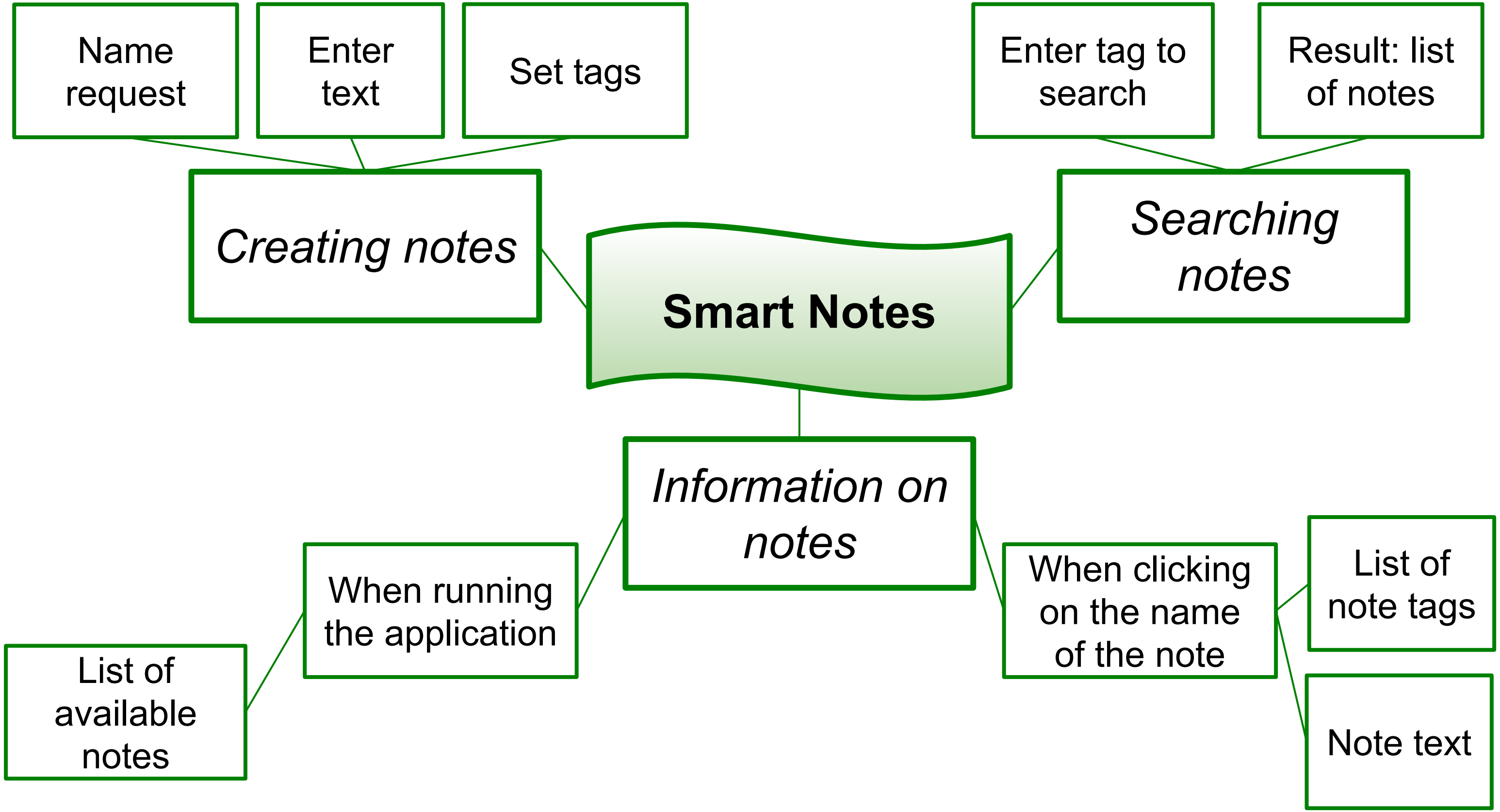
Qualifications confirmed!

Great, you are ready to brainstorm and work on your tasks!

Discussion:

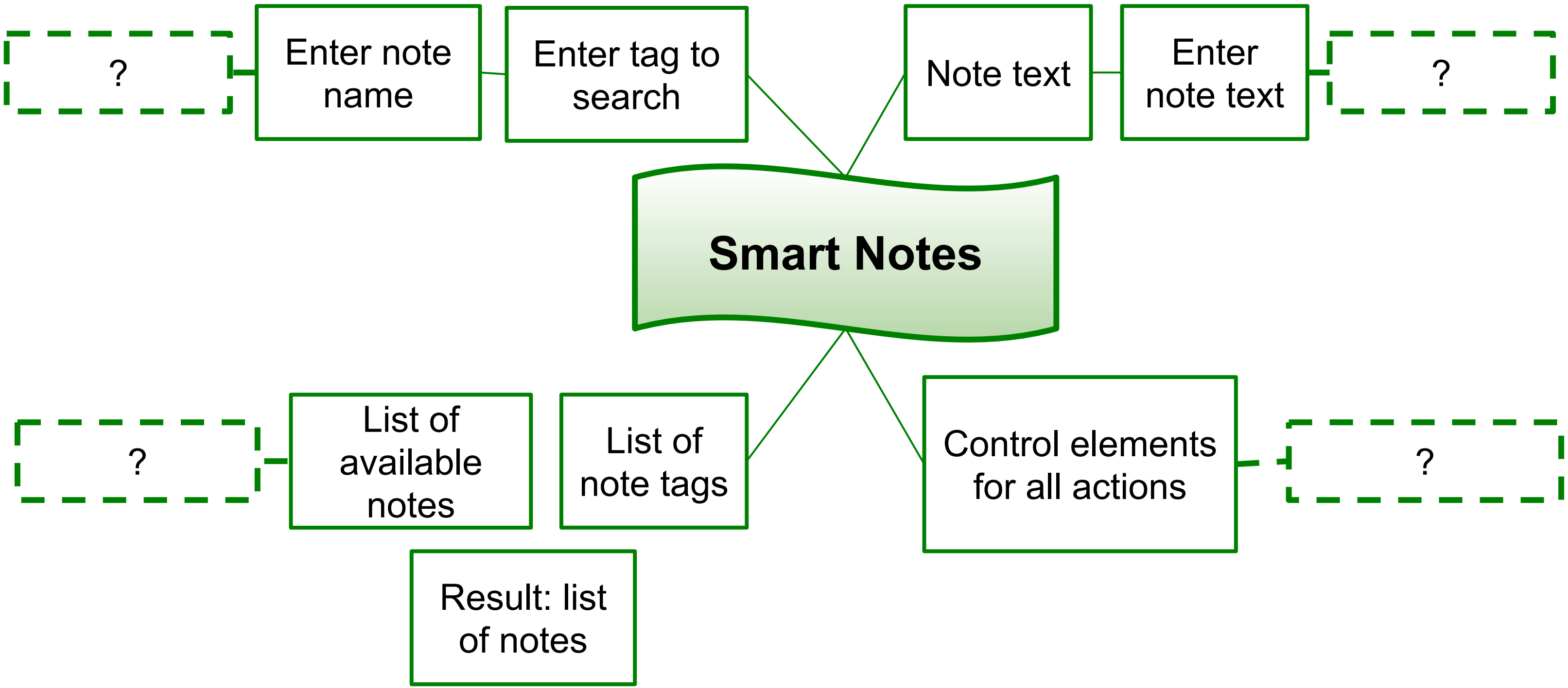
Smart Notes Interface

Smart Notes should have the following:



Discussion: Smart Notes

What widgets do we need for this?



Smart Notes

?

Enter note name

Enter tag to search

Note text

Enter note text

?

?

List of available notes

List of note tags

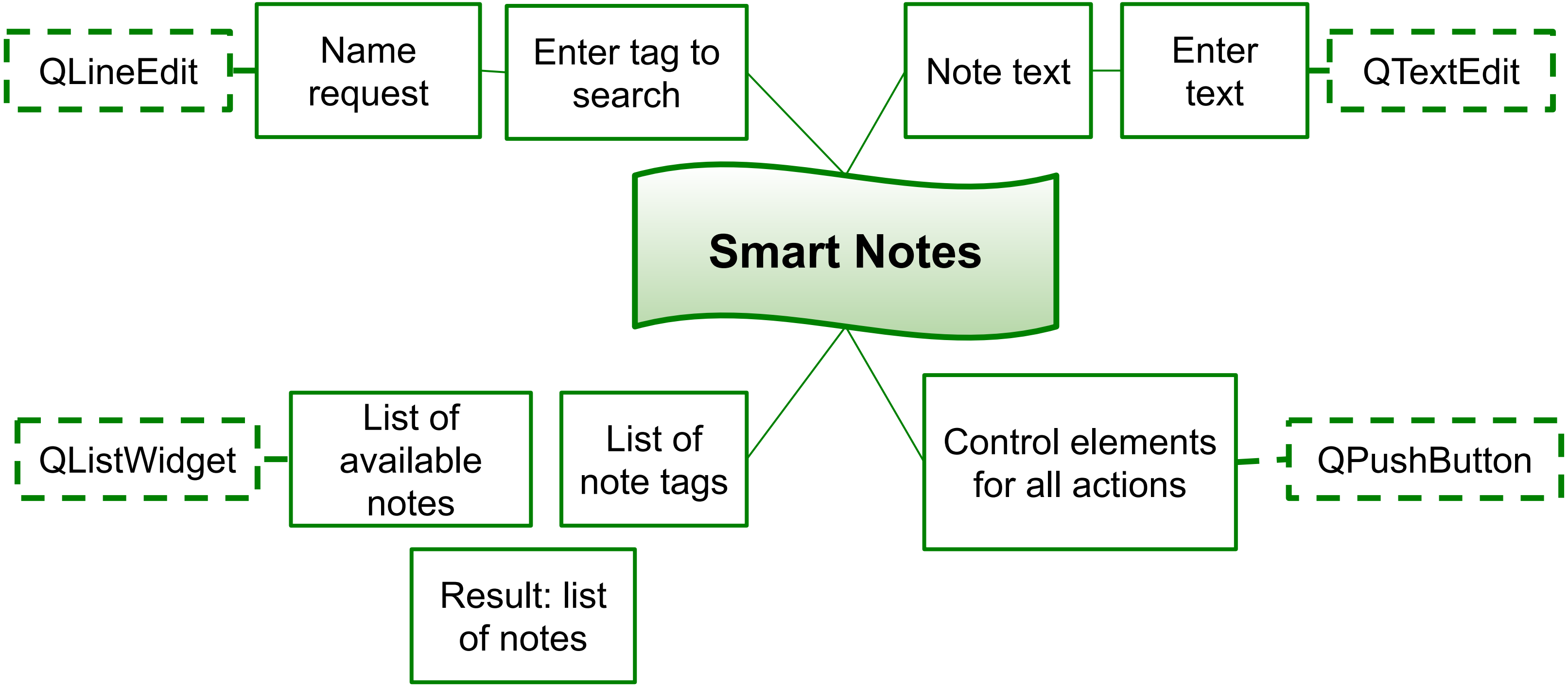
Result: list of notes

Control elements for all actions

?

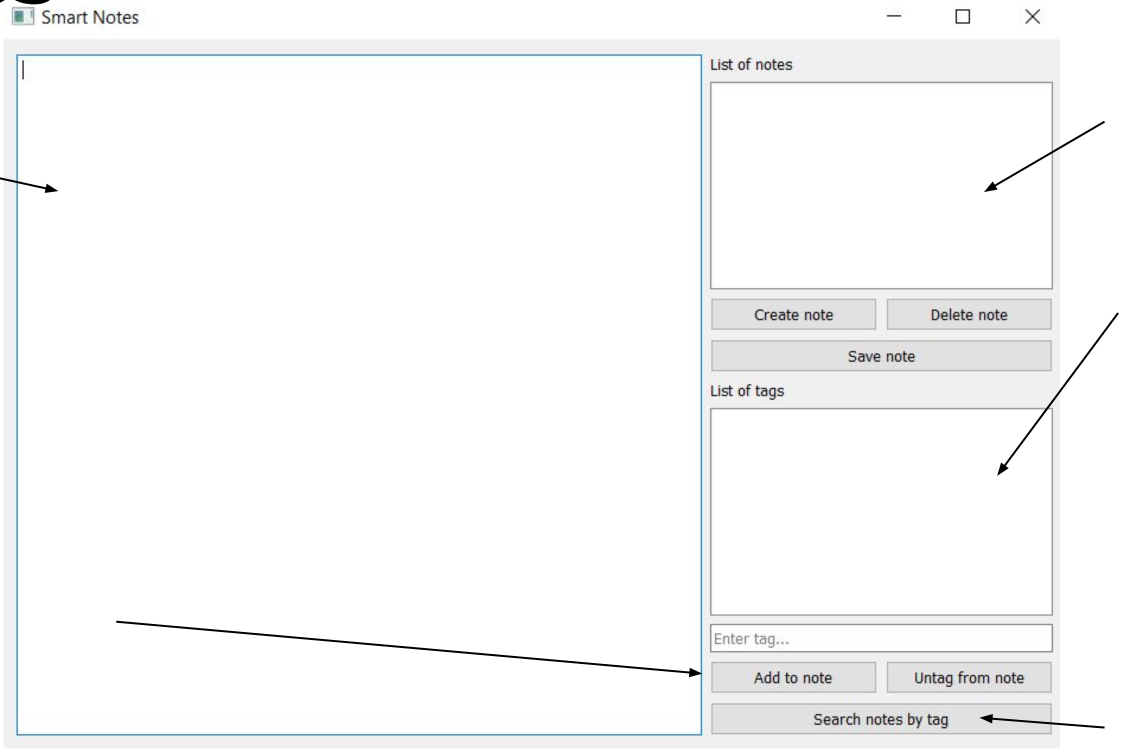
Discussion: Smart Notes

What widgets do we need for this?



Discussion:
Smart Notes

Possible interface for Smart Notes



Enter note text

Note list

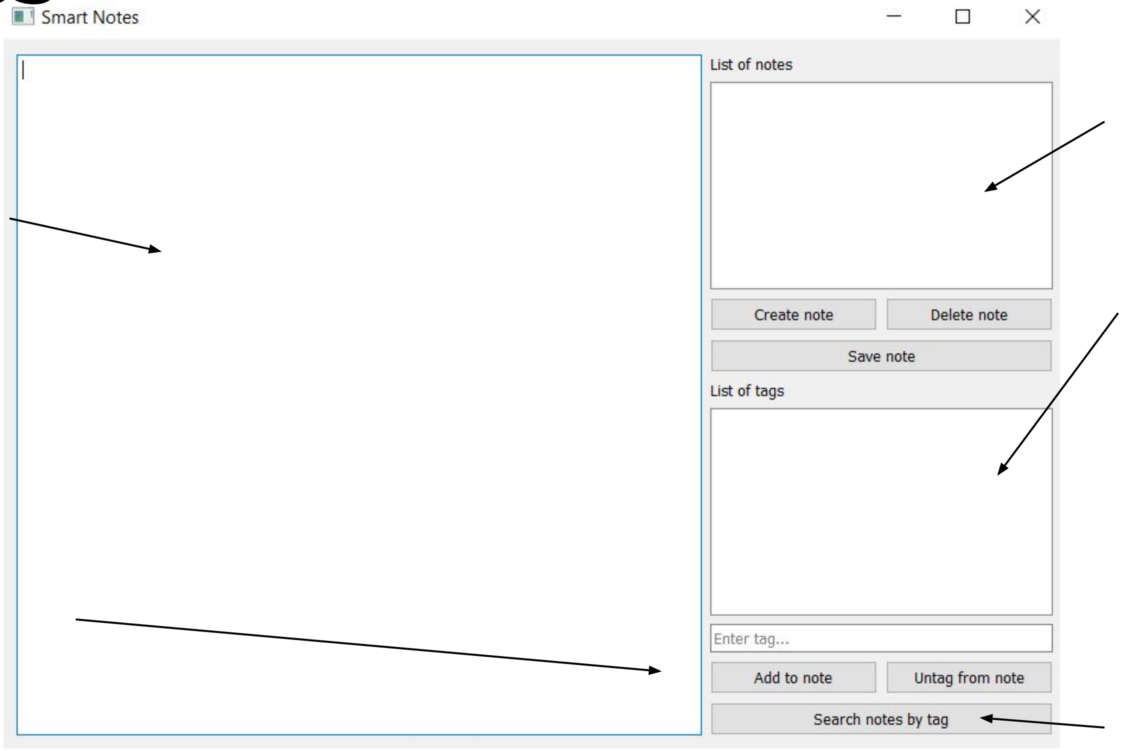
Tag list

Add tag

Search by tag

Discussion: Smart Notes

Possible interface for Smart Notes



QTextEdit

QListWidget

QListWidget

Add tag

Search by tag

Discussion: Smart Notes

Useful methods QTextEdit

Method	*Purpose*
`field_text = QTextEdit()`	Constructor for creating a QTextEdit field for entering text
`field_text.setText(Text)`	Set the text in parentheses in the field

Discussion:
Smart Notes

Useful methods QListWidget

Method	*Purpose*
`list_tags = QListWidget()`	Constructor for creating a QListWidget field for a list
`list_tags.addItems(Title_1)`	Adding items to a list
`list_tags.clear()`	Clearing QListWidget lists
`list_notes.itemClicked`	Is one of the items in the QListWidget list selected?
`list_notes.itemClicked.connect(...)`	*Using the method in event processing

Discussion:
Smart Notes

Tasks:

- Create the Smart Notes application interface.

- If you have any problems, use the tips.

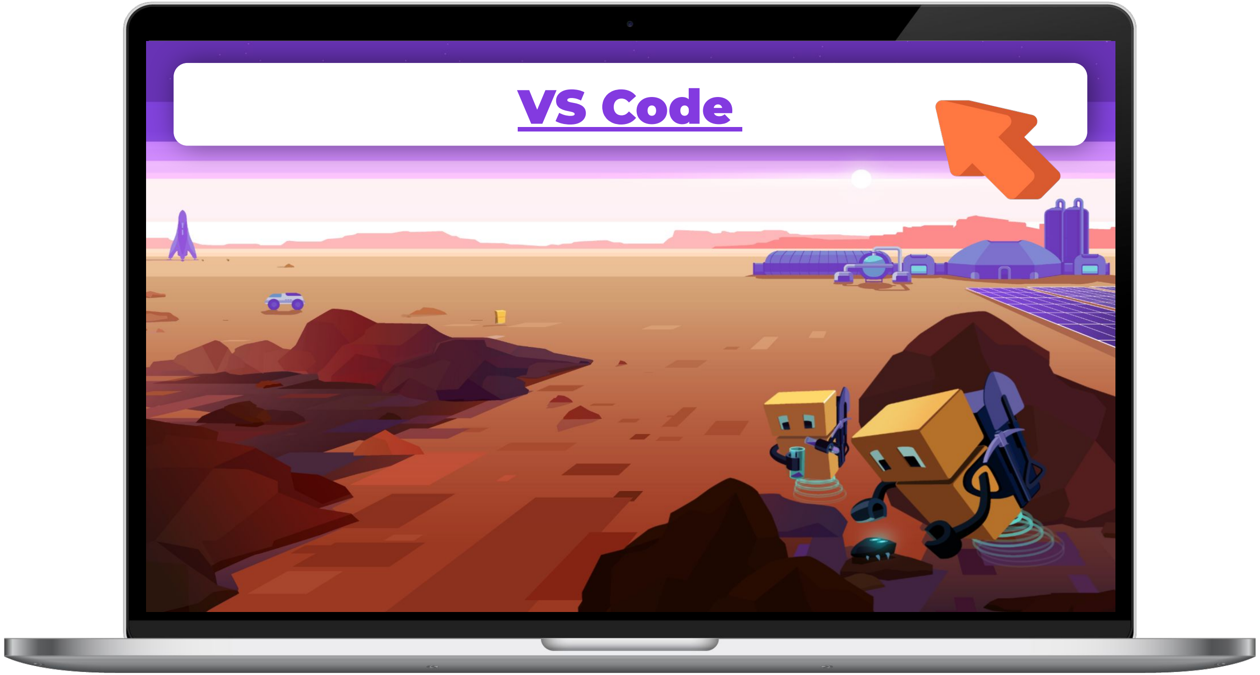
Module 3. Lesson 2. The Smart Notes application. P. 1

Visual Studio Code:

The Smart Notes application

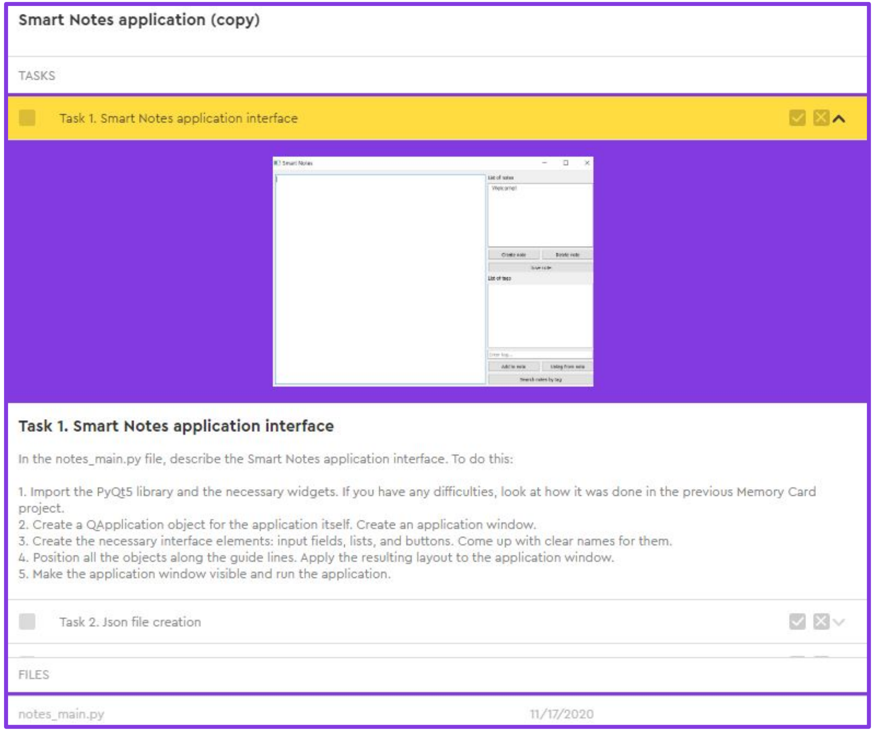
Complete the tasks in VS Code

VSC. Smart Notes application



Complete the tasks in VS Code

VSC. Smart Notes application



Smart Notes application (copy)

TASKS

□ Task 1. Smart Notes application interface

Task 1. Smart Notes application interface

In the notes_main.py file, describe the Smart Notes application interface. To do this:

1. Import the PyQt5 library and the necessary widgets. If you have any difficulties, look at how it was done in the previous Memory Card project.
2. Create a QApplication object for the application itself. Create an application window.
3. Create the necessary interface elements: input fields, lists, and buttons. Come up with clear names for them.
4. Position all the objects along the guide lines. Apply the resulting layout to the application window.
5. Make the application window visible and run the application.

□ Task 2. Json file creation

FILES

notes_main.py 11/17/2020

Complete the task Task 1. Smart Notes application interface.

<code>

Working on the platform

Break

Review:
Data Structures

What is a dictionary ?

Review

A dictionary is an unordered set of “ key : value” pairs

```
notes = {
    "About the sun" : "The sun is a star!",
    "About the earth" : "The earth is a planet!"
    }
```

Review

How do we get the value of a dictionary element using a key?

```
notes = {
    "About the sun" : "The sun is a star!",
    "About the earth" : "The earth is a planet!"
    }
```

Review

Getting the value of a dictionary element using a key:

```
value = notes["About the sun"]

print(value)


>>>The sun is a star!
```

Review

Match the features with the structures:

The elements are ordered

Elements are accessed using an index

The elements are key-value pairs

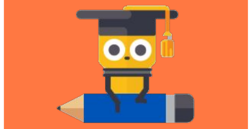
The in operator checks for elements

New elements are added using append()

The elements are not ordered

LISTS

DICTIONARIES



Review

Match the features with the structures:

LISTS

The elements are ordered

New elements are added using append()

Elements are accessed using an index

The in operator checks for elements

DICTIONARIES

The elements are not ordered

The elements are key-value pairs

The in operator checks for elements

Review

Yay! Now our application will definitely be the smartest!

Module 3. Lesson 2. The Smart Notes application. P. 1

New topic:

Storing data in json files

Problem:

- We need a structure for easy storing data about notes.

  (Notes can have transitions to new lines! How do we write them to a file and read from there?)

- This structure should be easy to use when working with PyQt.

Brainstorming

Possible structure:

```
notes = {
    "Note name" :
       {
        "text" : "Very important note text",
        "tags" : ["draft", "thoughts"]
       }
}
```

Brainstorming

Possible structure:

```
notes = {
     "Note name" :
         {
          "text" : "Very important note text",
          "tags" : ["draft", "thoughts"]
         }
}
```

Note name

Fields for one note

```
print(notes[“Note name”][“text”])
>>>Very important note text
```

Brainstorming

Potential problem:

- This structure is convenient. But it may be hard to read and write to the file.

It would be nice if the structure in the file and the structure in the program looked the same.
Then reading and writing information to the file would be very simple.

Potential problem:

- This structure is convenient. But it may be hard to read and write to the file.

Solution:

- It turns out that this structure is used by programmers around the world.

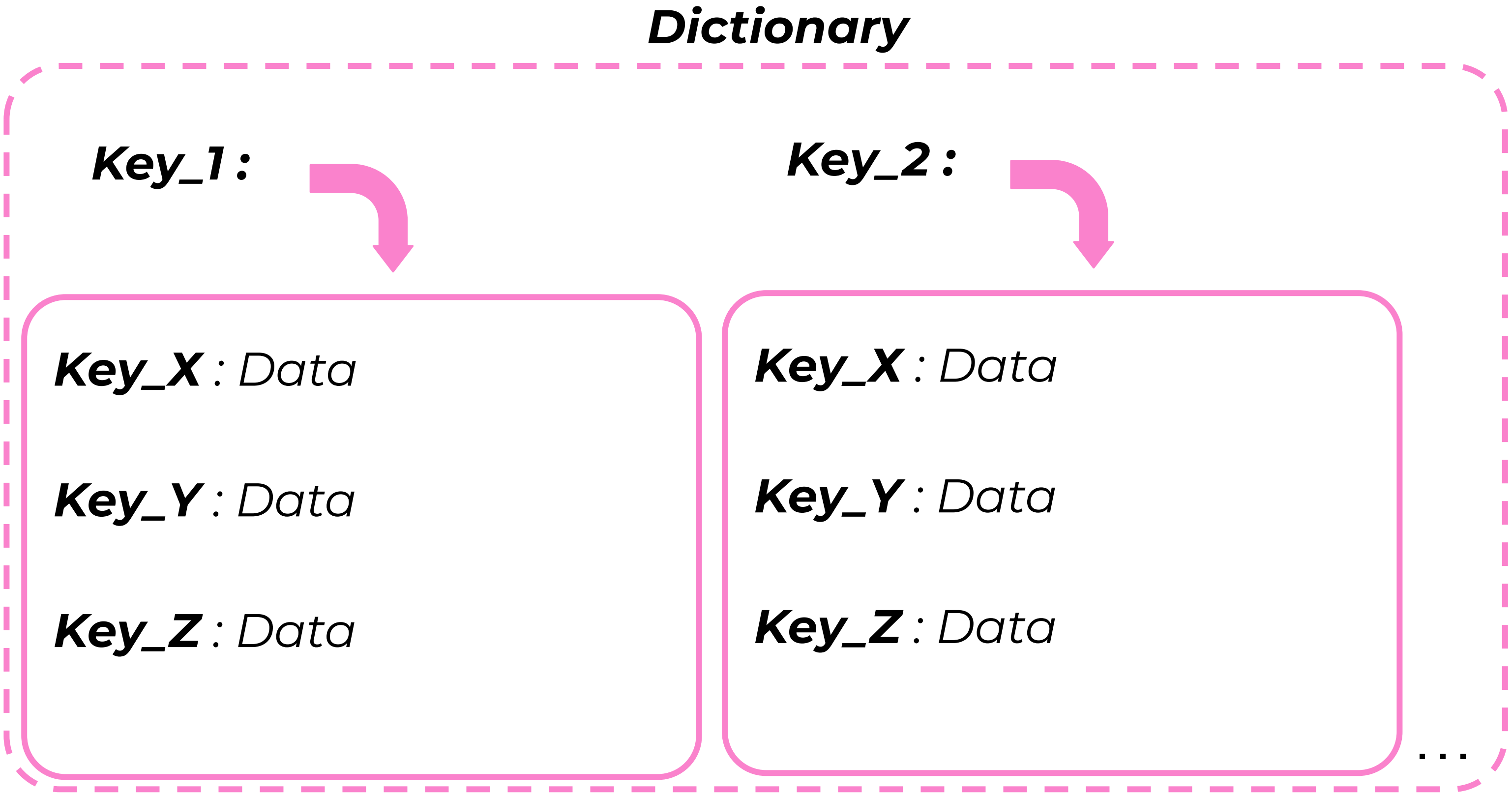
  Let's look at a ready-made solution.

A json file is a file with a ready-made structure that's easy to read and use.

The structure of a json file is very similar to the system of nested dictionaries and lists in Python.

Brainstorming

Json file:



Dictionary

Key_1 :

Key_X : Data

Key_Y : Data

Key_Z : Data

Key_2 :

Key_X : Data

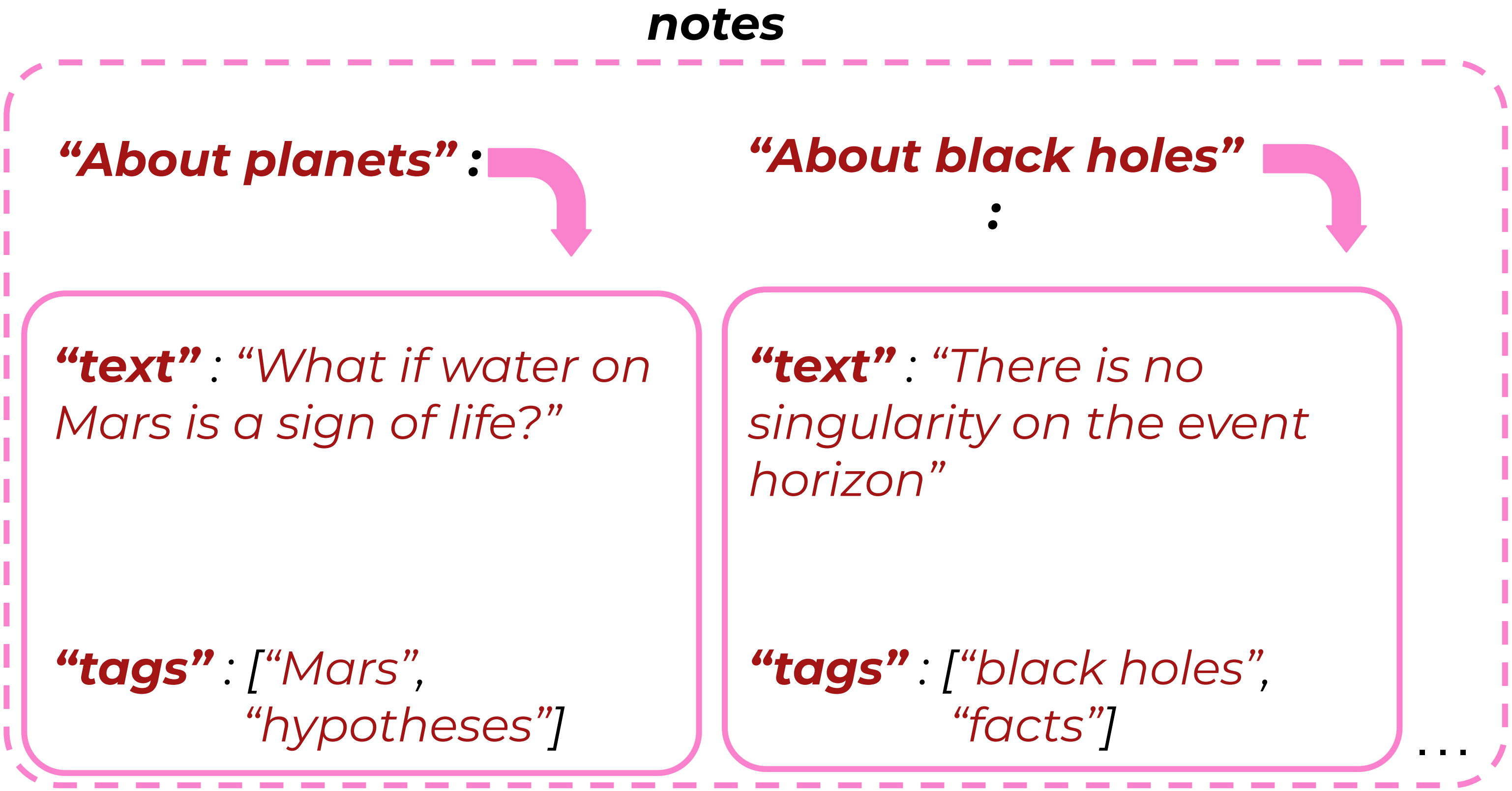
Key_Y : Data

Key_Z : Data

. . .

Brainstorming

Json file with notes:



notes

"About planets" :

"text" : *"What if water on Mars is a sign of life?"*

"tags" : *["Mars", "hypotheses"]*

"About black holes" :

"text" : *"There is no singularity on the event horizon"*

"tags" : *["black holes", "facts"]*

…

Brainstorming

Json file with notes:

```
{
        "About planets" :
                {
                        "text" : "What if water on Mars is a sign of life?",
                        "tags" : ["Mars", "hypotheses"]
                },
        "About black holes" :
                {
                        "text" : "There is no singularity on the event horizon",
                        "tags" : ["black holes", "facts"]
                }
}
```

Brainstorming

Reading json files:

Command	*Purpose*
`import json`	Connecting the json library
`with open("f.json", "r") as file:`	Open the json file for reading
`    data = json.load(file)`	Upload the structure from the json file to the data dictionary

After reading it, data has the same structure as the json file!

Brainstorming

Writing in json files:

Command	*Purpose*
`import json`	Connecting the json library
`with open("f.json", "w") as file:`	Open the json file for writing
`    json.dump(data, file)`	Upload the structure from data to the json file

The json file is being completely overwritten!

Brainstorming

Useful writing parameters:

Command	*Purpose*
`encoding="utf-8"`	Set a universal text encoding (can be useful for writing data)
`sort_keys=True`	Sort master keys (note titles) when writing

Example:

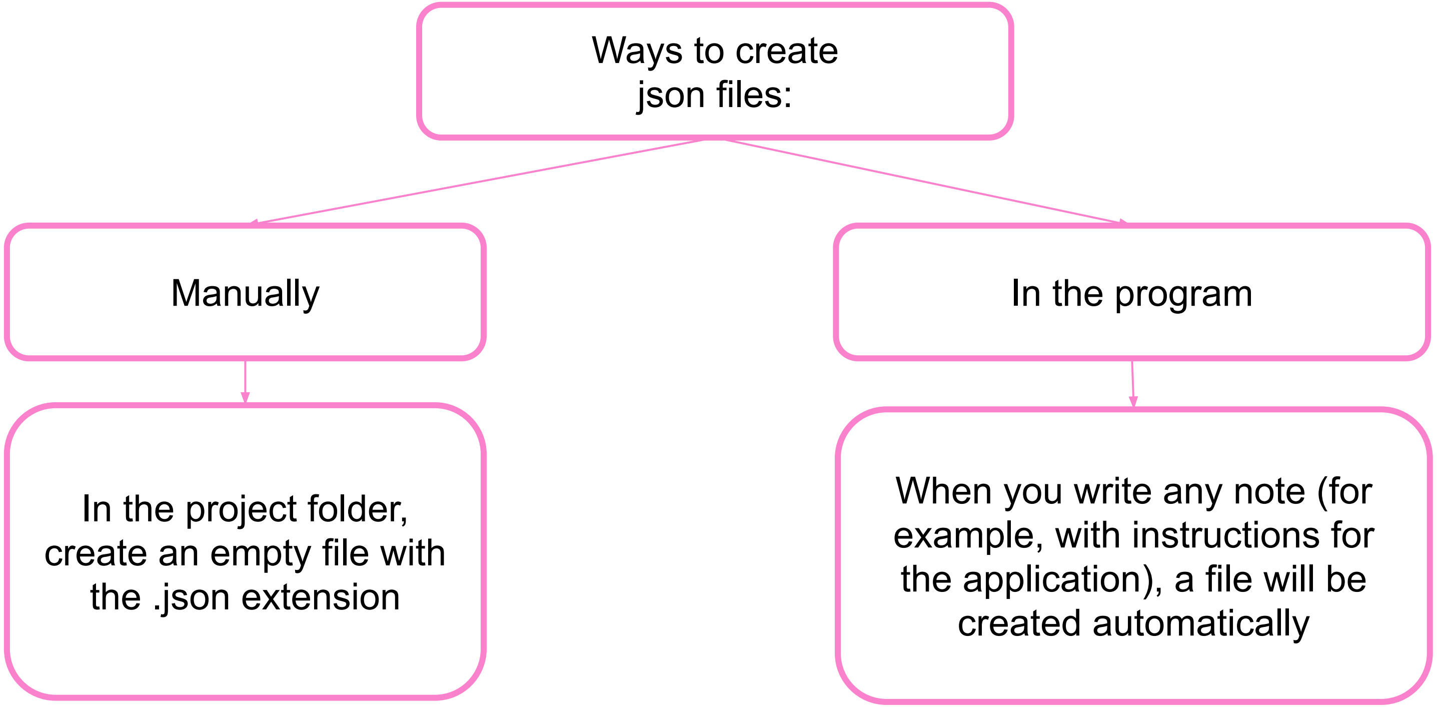
```
json.dump(data, file, sort_keys=True)
```

Load the dictionary data in file...

...after sorting the keys alphabetically...

Brainstorming

Creating json files:



Ways to create json files:

Manually

In the project folder, create an empty file with the .json extension

In the program

When you write any note (for example, with instructions for the application), a file will be created automatically

If a file with the same name has not existed before.

Brainstorming

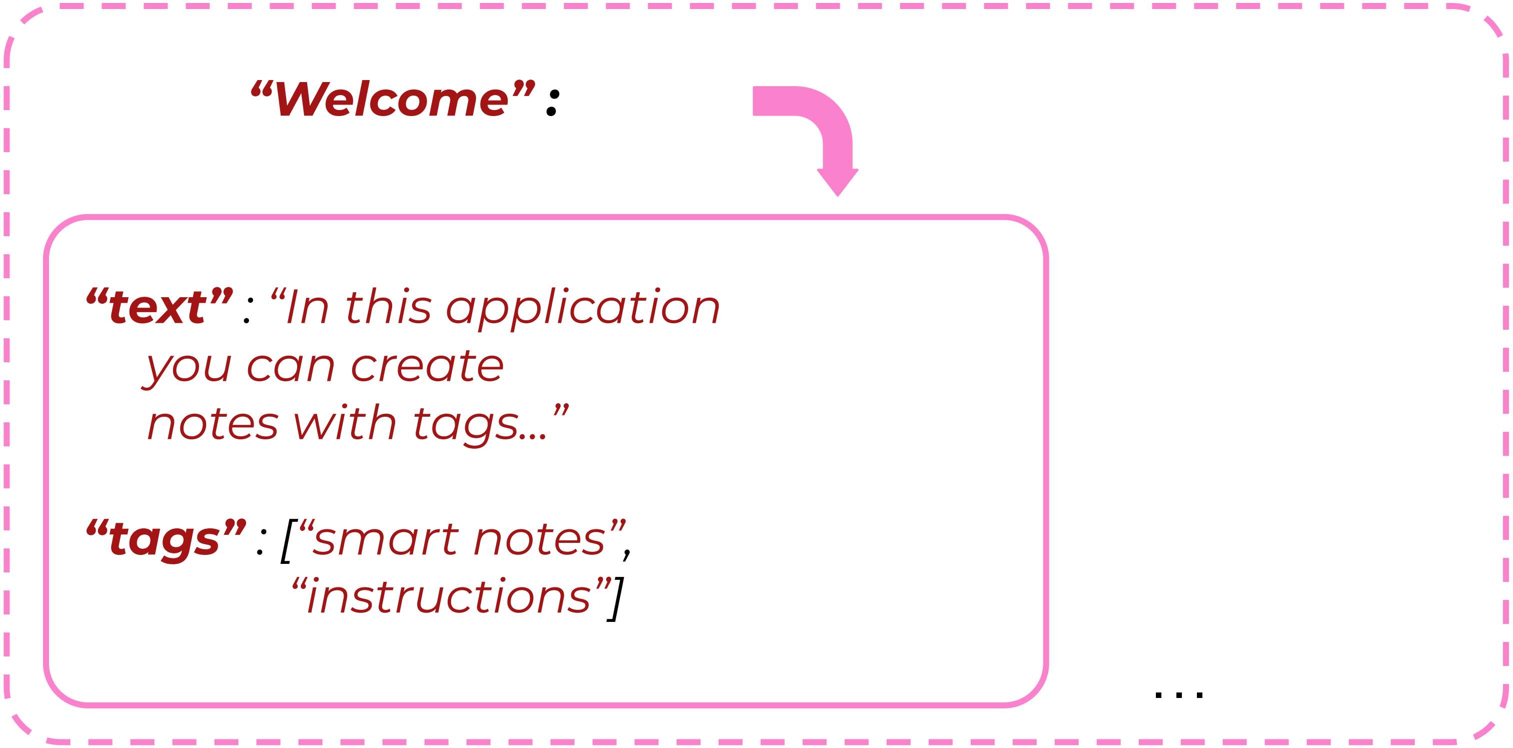
Let's move on to Smart Notes:

- In the notes_main.py file, create a notes dictionary with notes.
- In notes, create one “Welcome” note with instructions for working in Smart Notes.
- Let's write this note to the notes_data.json file (the file will be created automatically).

Brainstorming

The notes dictionary:

notes



“Welcome” :

“text” : *“In this application you can create notes with tags…”*

“tags” : *[“smart notes”, “instructions”]*

…

Brainstorming

Writing notes to a json file:

Open a **json** file to write data

:

Write the **notes** dictionary to the file (if necessary, specify parameters)

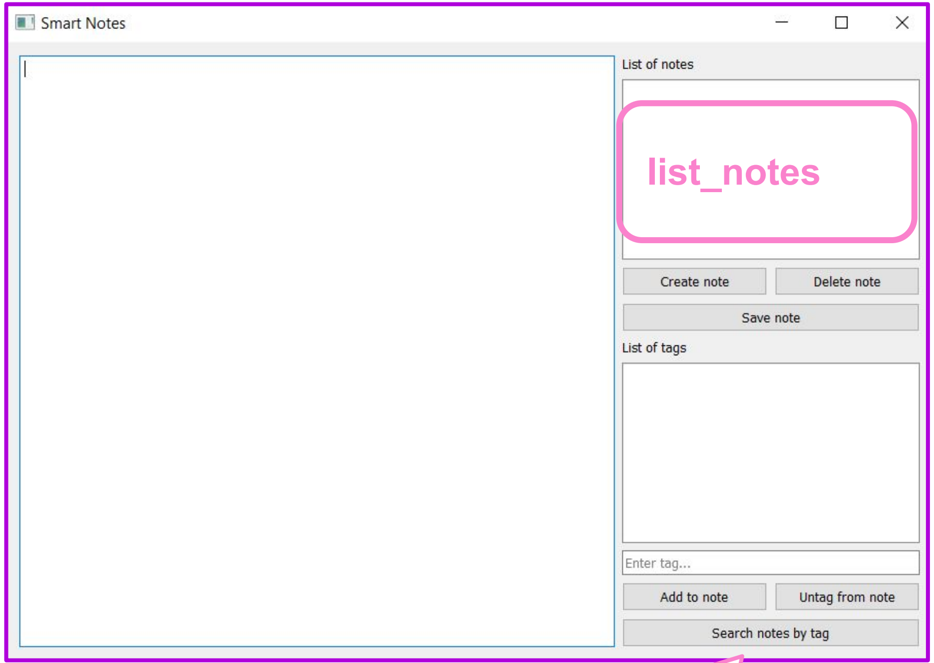
Outcome.
The *notes_data.json* file has been created, which contains the starting note. The structure for notes is now set.

What does the program do with the notes_data.json file data after starting the application?

Brainstorming

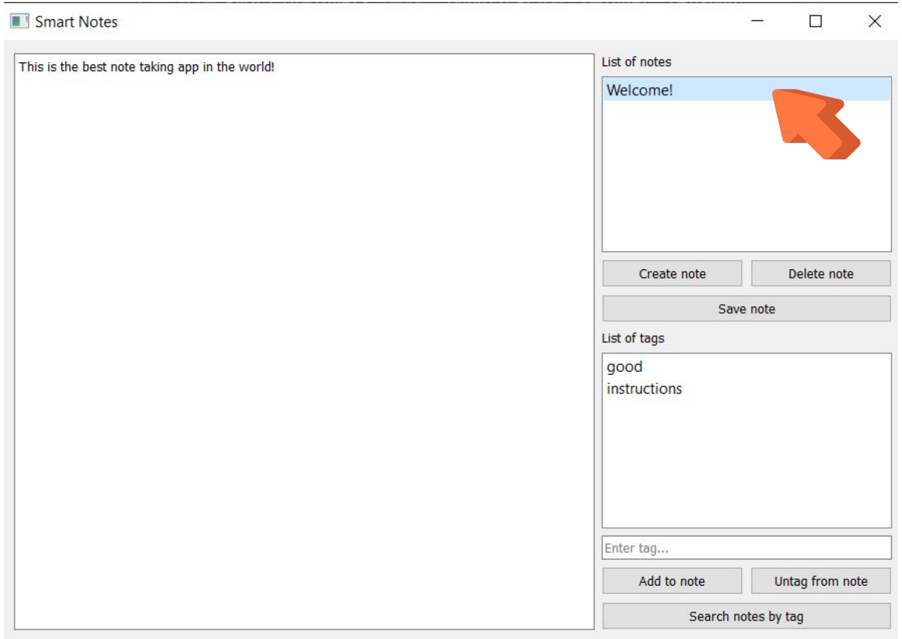
Run the application:

- Opening a json file for reading and loading data into the notes structure.
- Displaying note titles in a QListWidget.



```
list_notes.addItems(notes)
```

What does the program do when you click on a note’s title?



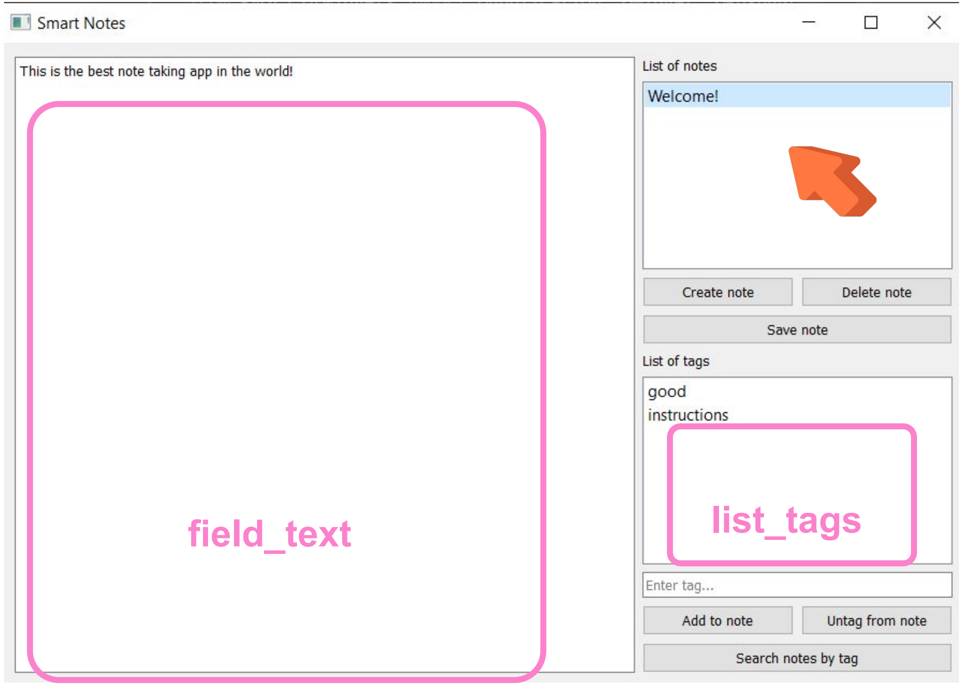
This is the best note taking app in the world!

Brainstorming

Title click handling:

The list of note tags and the text should be displayed.

- Calling the processor function `show_note( )`



This is the best note taking app in the world!

Title click handling:

```
def show_note():
```

We get the **title** of the selected note as a string.

We set the text of the note with the found title in field_text (QTextEdit).

We clear the list of tags (if there was something there) and add the tags of the note with the found title there.

Brainstorming

Title click handling:

```
def show_note():
    name = list_notes.selectedItems()[0].text()
    field_text.setText(notes[name]["text"])
    list_tags.clear()
    list_tags.addItems(notes[name]["tags"])
```

Brainstorming

Tasks:

- Create a note with instructions and write it to a json file, thereby creating it.
- When starting the application, read the information from the json file and place it in the widgets.
- Process clicking on the note title in the list.

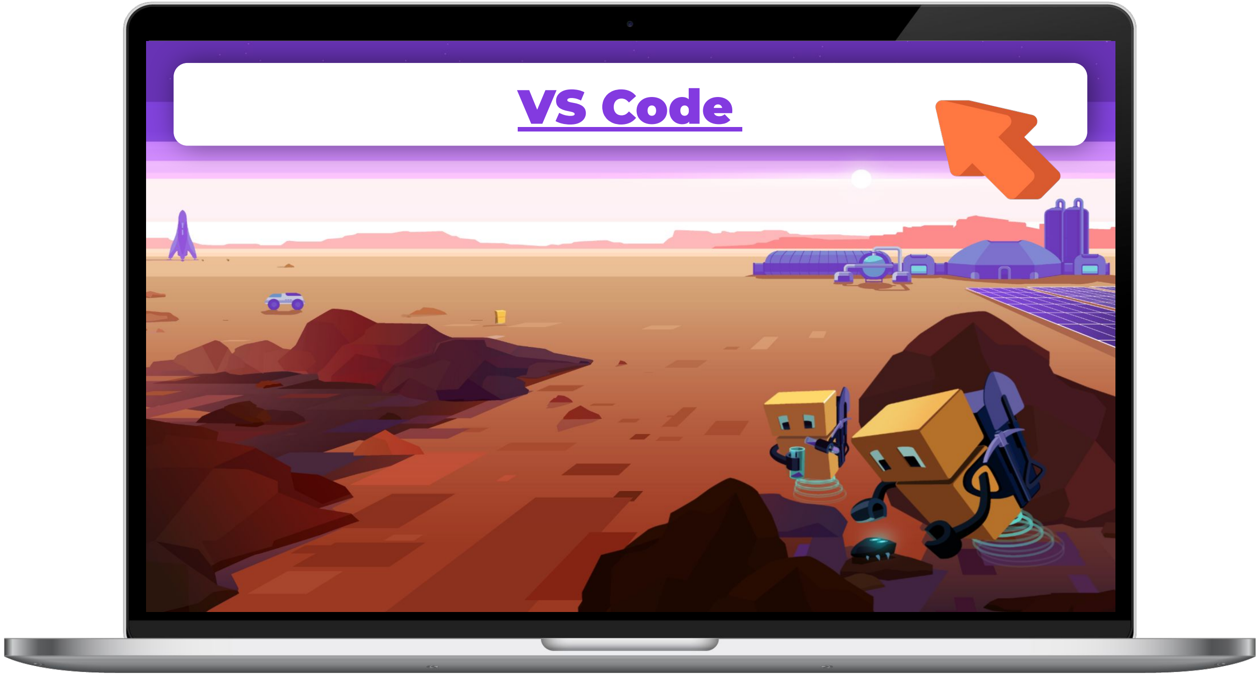
Brainstorming

Visual Studio Code:

The Smart Notes application

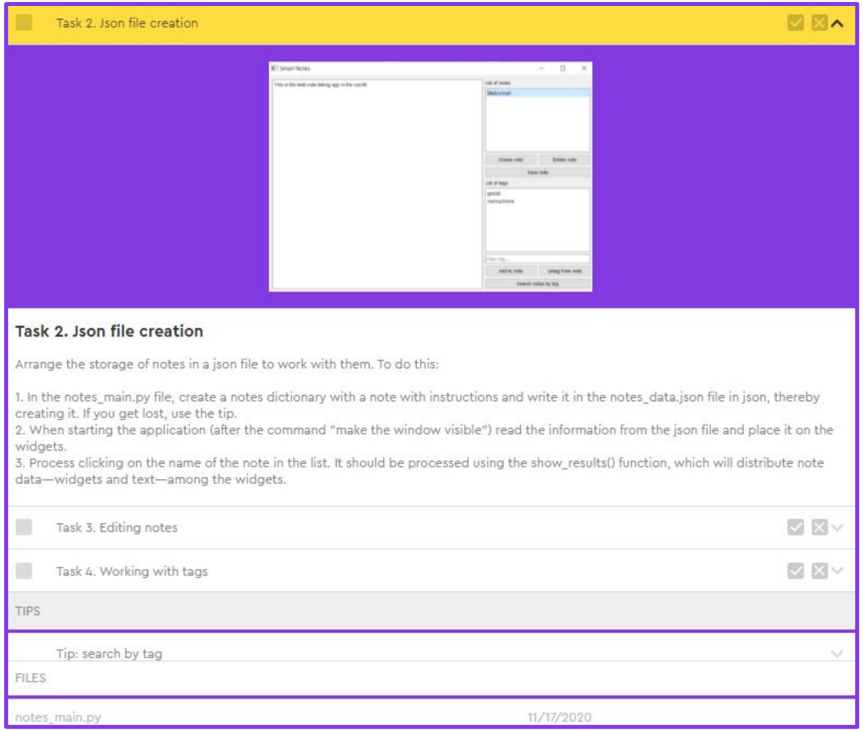
Complete the tasks in VS Code

VSC. Smart Notes application



Complete the tasks in VS Code

VSC. Smart Notes application



□ Task 2. Json file creation ☑ ☒

Task 2. Json file creation

Arrange the storage of notes in a json file to work with them. To do this:

1. In the notes_main.py file, create a notes dictionary with a note with instructions and write it in the notes_data.json file in json, thereby creating it. If you get lost, use the tip.
2. When starting the application (after the command "make the window visible") read the information from the json file and place it on the widgets.
3. Process clicking on the name of the note in the list. It should be processed using the show_results() function, which will distribute note data—widgets and text—among the widgets.

□ Task 3. Editing notes ☑ ☒

□ Task 4. Working with tags ☑ ☒

TIPS

Tip: search by tag

FILES

notes_main.py 11/17/2020

Complete the task
Task 2.
Creating a json file.

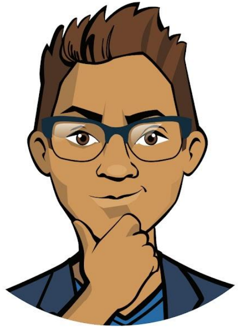
Working
on the platform

Module 3. Lesson 2. The Smart Notes application. P. 1

Wrapping up the work day

Let’s end up the work day by answering these technical questions:

1. What is a json file? What are their advantages over regular text files?
2. How do we read json files and write data to them?
3. What is the best operator to use for opening and closing files?
4. How do we read a file line by line?



Cole,
senior developer

Emily,
project manager

Wrapping up the work day

Excellent work!

Colleagues,

Today you programmed the interface of the Smart Notes application and arranged the storage of notes in a json file.

Outside working hours, be sure to add explanatory comments to the code and look at the theoretical documentation

Wrapping up the work day