

Practical Project: Random Sentences Generator

This is an additional practical project, and it is not mandatory and it is not included in the final score. The main purpose is to use the gained knowledge in different types of problems and to improve your portfolio and GitHub skills.



This **random sentence generator** is just for fun! These sentences can provide humor and be a cool way to surprise others by sharing a standout sentence on social media platforms and gathering your network's reaction.

```
Hello, this is your first random sentence:  
Michell from Varna slowly sees cake  
Click [Enter] to generate a new one.  
Jane from Varna happily brings stones  
Click [Enter] to generate a new one.
```

1. Create GitHub Repository

Create a **new repository** from <https://github.com/new>. Choose a **meaningful name**, e.g...

"RandomSentencesGeneratorByUsername", add a **short description**, and make your repo **public**. Also, add a **README.md** file and **.gitignore** for **Python**. Finally, **change the license** to "MIT" and click on the **[Create]** button to **create your repository**.

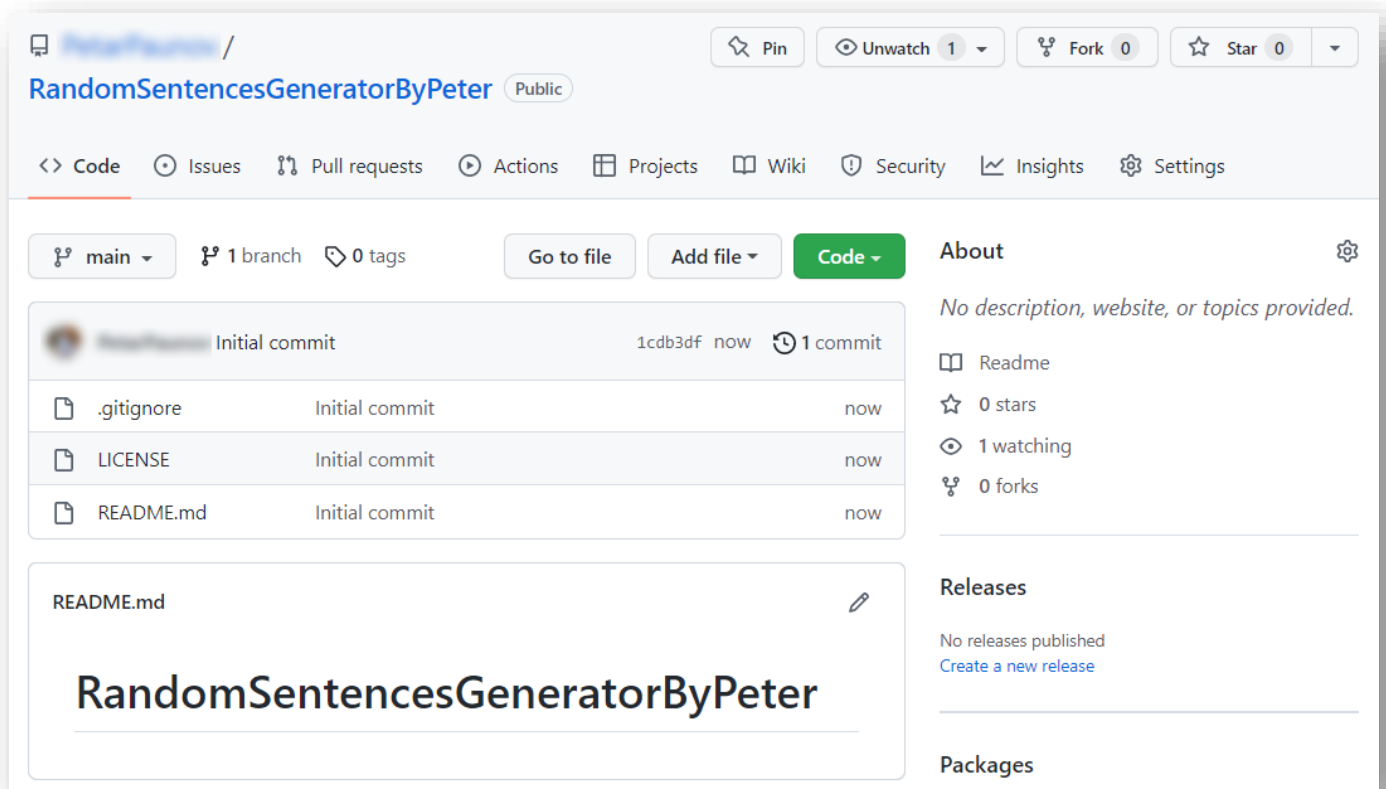


Please choose **your original and unique name** for your project!

Your GitHub profile should be **unique**.

You can follow this tutorial, but you can also **make changes** and **implement your project differently**.

Now your **repository is created** and should look like this:



Now let's see how to **write the code** of our application.

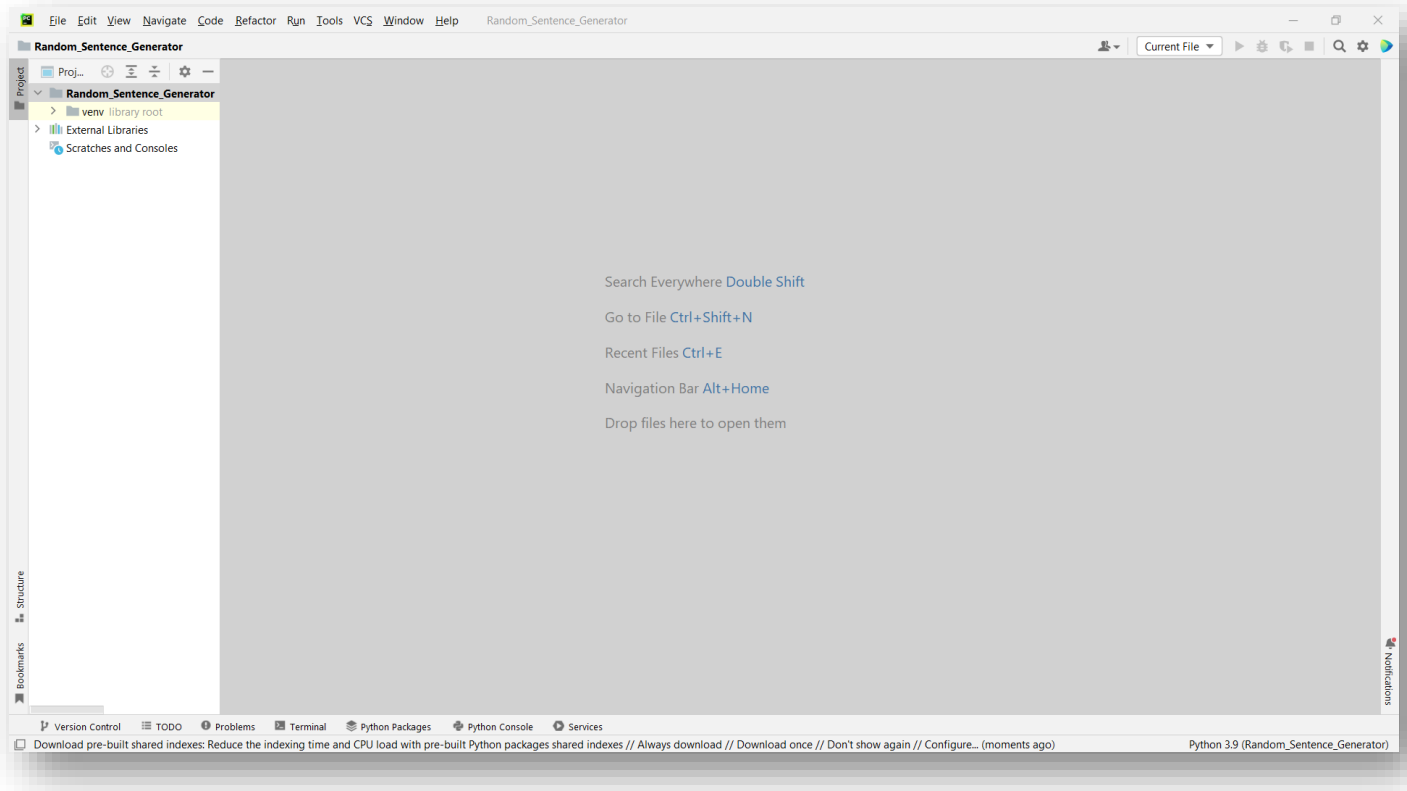
2. Write the Sentences Generator Code

Let's create the application and play with it.

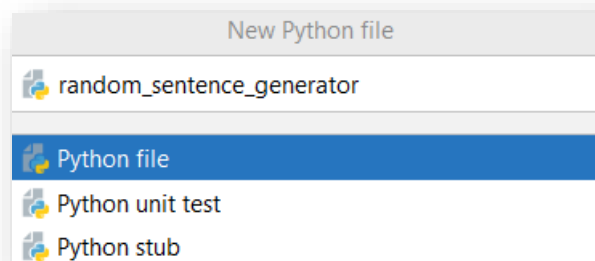
Create a PyCharm Project

First, we should **start PyCharm** and **create a new Python project**. Then, **choose an appropriate name** and a **place to save the project**.

Our project should be created and should look like this:



We should create a **new Python file** with the name of the game.



Implement the Generator Logic

Now let's start working on our project.

Create the Sentence Model

To create our **sentences** we are going to need: **names**, **places**, **verbs**, **nouns**, **adverbs**, and **details**. The **sentence** that we will create is based on the following **model**:

- One **sentence** needs [Who from where] [Action] [Detail] to be created.
 - "**Who from where**" example: [Name + from + Place] ("David from London").
 - "**Action**" example: [Adverb] + [Verb] + [Noun] ("calmly watched the sunset").
 - "**Detail**" example: "near the river", "at home", "in the park".

Add Words for the Sentences

Let's start by creating **lists** with all the **words** we will use to create a **random sentence**. **Lists** are used to **store multiple** values in a **single variable**, instead of **declaring separate variables** for each **value**.

To **declare** a **list**, we use **square brackets** or **"list()"**.

Now let's create our first **list** and call it **"names"**. To fill the **list** we will use **brackets**. Inside the **brackets**, write **names, separated** by a **comma**. These are some example names that you can use:

```
"Peter", "Michell", "Jane", "Steve"
```

Your list should look like this:

```
1 names = ["Peter", "Michell", "Jane", "Steve"]
```

Now we need to create **lists** with words for **"places"**, **"verbs"**, **"nouns"**, **"adverbs"** and **"details"**. Do this by yourself. Here are some **words** you can use:

- **Places:**

```
"Sofia", "Plovdiv", "Varna", "Burgas"
```

- **Verbs:**

```
"eats", "holds", "sees", "plays with", "brings"
```

- **Nouns:**

```
"stones", "cake", "apple", "laptop", "bikes"
```

- **Adverbs:**

```
"slowly", "diligently", "warmly", "sadly", "rapidly"
```

- **Details:**

```
"near the river", "at home", "in the park"
```

Finally, the lists should look like this:

```
1 names = ["Peter", "Michell", "Jane", "Steve"]
2 places = ["Sofia", "Plovdiv", "Varna", "Burgas"]
3 verbs = ["eats", "holds", "sees", "plays with", "brings"]
4 nouns = ["stones", "cake", "apple", "laptop", "bikes"]
5 adverbs = ["slowly", "diligently", "warmly", "sadly", "rapidly"]
6 details = ["near the river", "at home", "in the park"]
```

Create a Function for Getting a Random Word

Now we are going to create a **function**. Generally, **functions** are useful to **improve** code **reusability** by **reducing** code **duplication**. If we have the same **functionality** to perform in **multiple places**, then we can create one **function** with the required **functionality** and reuse it wherever it is **necessary** for the **application**. In our case, the **function** will help us choose **random words** every time.

To create a **function**, you need the following things:

- First, the reserved word **"def"**.
- Second, we need a **name** for the **function**.
- Third, we should define **parameters** that the **function** will receive

Do it as follow:

```
1 def get_random_word(words):  
2     pass
```

Now let's write the function logic. First, we need to import the library "**random**", then we will return the value from the method "**random.choice()**":

```
1 import random  
2  
3  
4 def get_random_word(words):  
5     return random.choice(words)
```

Note: The "**choice()**" method returns a randomly selected element from the specified sequence.

More info [here](#).

It's time for the easy part – let's make the generator work.

First, we should create an endless **while loop** and create **variables** for all different **random words**. To do this we will use our **function get_random_word()**, which will do all the work for us.

Second, create a **variable** and name it "**random_name**". Make the **variable** keep the result from our **get_random_word()** function and **pass our words list** as an **argument** to the function. Do it as follow:

```
16 while True:  
17     random_name = get_random_word(names)
```

Now try to create **variables** for the other **words** yourself. They should all **pass the necessary lists** and **keep the results** from the **get_random_word()** function. Finally, it should look like this:

```
16 while True:  
17     random_name = get_random_word(names)  
18     random_place = get_random_word(places)  
19     random_verb = get_random_word(verbs)  
20     random_noun = get_random_word(nouns)  
21     random_adverb = get_random_word(adverbs)  
22     random_detail = get_random_word(details)
```

The next thing is to **construct** our **random sentence** and **print it** on the console. Remember the **model** that we are working on - we need "**Who from where**", then "**Action**" and last "**Details**":

```
24 print(f"{random_name} from {random_place} {random_adverb} {random_verb} {random_noun}")
```

Now what is left is to **write the sentence** on the **console**. Next, write a **message** to the user to press [**Enter**] to **generate** a new **sentence** and **read his input**. You know how to do that:

```
25 print(f"{random_name} from {random_place} {random_adverb} {random_verb} {random_noun}")  
26 input("Click [Enter] to generate a new one.")
```

You can also **write** a **greeting message** before the **while** loop:

```
15 print("Hello, this is your first random sentence:")
16
17 while True:
```

This is all it takes to **finish** our **project** after you run it, the generator should look like this:

```
Hello, this is your first random sentence:
Jane from Sofia rapidly eats cake
Click [Enter] to generate a new one.
```

```
Hello, this is your first random sentence:
Michell from Varna slowly sees cake
Click [Enter] to generate a new one.
Jane from Varna happily brings stones
Click [Enter] to generate a new one.
```

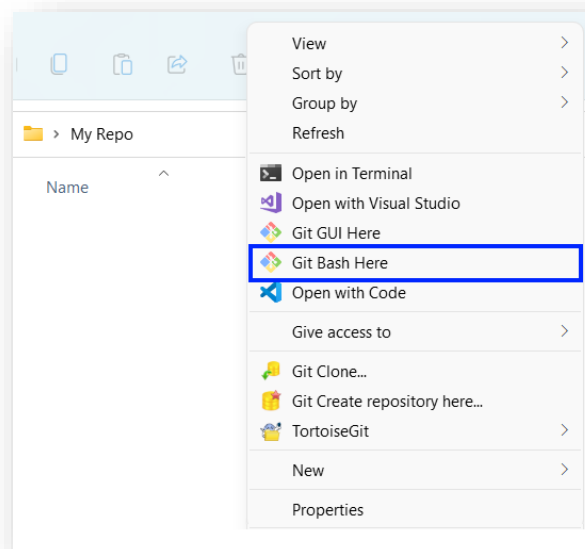
Now let's upload it to **GitHub**.

3. Upload Your Project to GitHub

We already know how to clone our repository by using **Git Bash** or **GitHub Desktop**.

Use GitBash (Option 1)

Go to the desired **directory**, right-click on a blank space **anywhere** in the folder, and select "**Git Bash Here**" to open the Git command line console. If the "**Git Bash Here**" menu is missing, you should first install Git.



Type the "**git clone**" command followed by the link to your **repository**:





```
git clone
```

This command is for cloning with **Git Bash**, paste your **repository URL** after the command.

```
@DESKTOP-8KNC31S MINGW64 ~/PycharmProjects/Random_Sentence_Generator
$ git clone https://github.com/DiyanKalaydzhiiev23/RandomSentenceGeneratorByPeter.git
Cloning into 'RandomSentenceGeneratorByPeter'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Your files from your GitHub repo will be downloaded to a **sub-folder** called as your project in GitHub, "**RandomSentencesGeneratorByPeter**" in our case.

The next thing to do is to add your project files to your cloned repository folder. It should look like this:

 .git	7.8.2022 г. 12:33	Папка с файлове	
 .gitignore	7.8.2022 г. 12:33	Текстов документ	2 КБ
 LICENSE	7.8.2022 г. 12:33	Файл	2 КБ
 random_sentence_generator	7.8.2022 г. 12:23	JetBrains PyCharm	1 КБ

Now we are ready to upload our changes from the "**Git Bash clone**". Go to the desired **folder**, right-click on a blank space anywhere in the folder, select "**Git Bash Here**" and run the following **commands**.

Type the following command:

```
git status
```

The **git status** command displays the state of the working directory and the **staging area**.

```
@DESKTOP-8KNC31S MINGW64 ~/PycharmProjects/Random_Sentence_Generator/RandomSentenceGeneratorByPeter (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    random_sentence_generator.py

nothing added to commit but untracked files present (use "git add" to track)
```

Now type:

```
git add .
```

This command **adds** all modified files.

Next type:

```
git commit -m "Your message here"
```

This command **commits** your changes. We also should **add** an appropriate **message**.

Second to the last type.

```
git pull
```

This command **updates** your local **repository**.

Now the last thing that we should do is to **push** our changes by using the command:

```
git push
```

This command **pushes** your changes to our local **repository**.

```
@DESKTOP-8KNC31S MINGW64 ~/PythorProjects/Random_Sentence_Generator /RandomSentenceGeneratorByPeter (main)
$ git add .

@DESKTOP-8KNC31S MINGW64 ~/PythorProjects/Random_Sentence_Generator /RandomSentenceGeneratorByPeter (main)
$ git commit -m "Added new project Random Word Generator"
[main f8f121f] Added new project Random Word Generator
1 file changed, 26 insertions(+)
create mode 100644 random_sentence_generator.py

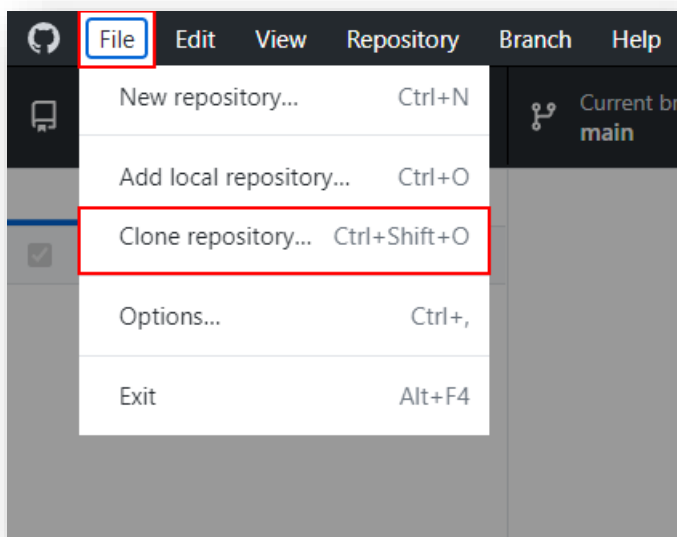
@DESKTOP-8KNC31S MINGW64 ~/PythorProjects/Random_Sentence_Generator /RandomSentenceGeneratorByPeter (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 756 bytes | 756.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DiyanKalaydzhiev23/RandomSentenceGeneratorByPeter.git
dc8e57c..f8f121f main -> main
```

This is all you need to **update** your **repository** with **Git Bash**.

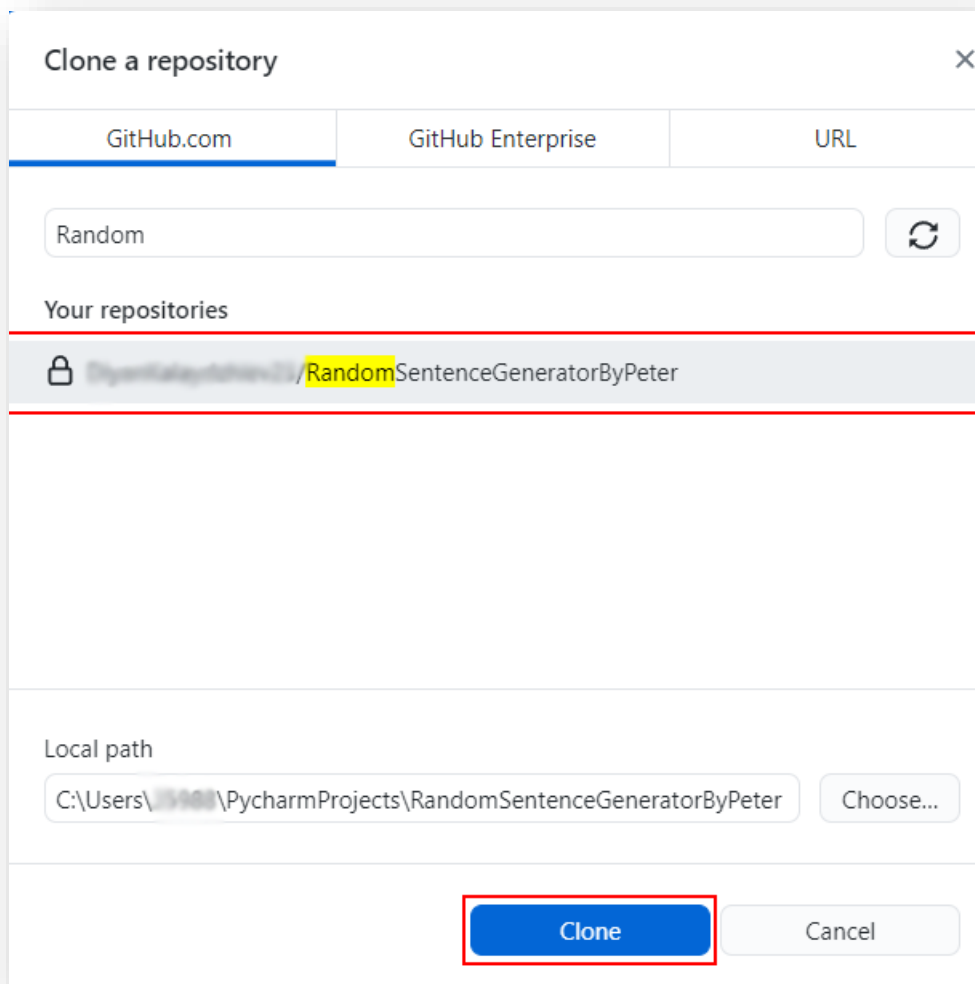
Use GitHub Desktop (Option 2)

If you don't have GitHub Desktop on your computer, download and install it from [here](#).

Go to "**File**" and choose "**Clone repository**".



Choose the repository for the project, in our case "RandomWordsGeneratorByPetar" and hit the "**Clone**" button.

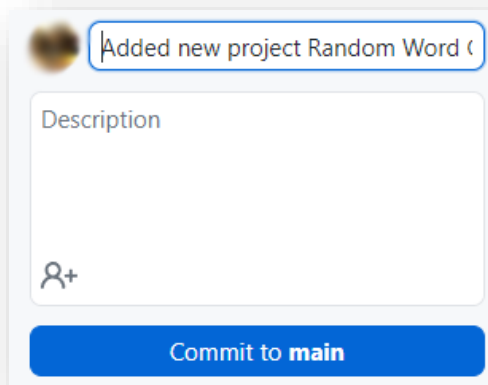


Your files from your GitHub repo will be downloaded to a **sub-folder** called as your project in GitHub, "**RandomSentencesGeneratorByPeter**" in our case.

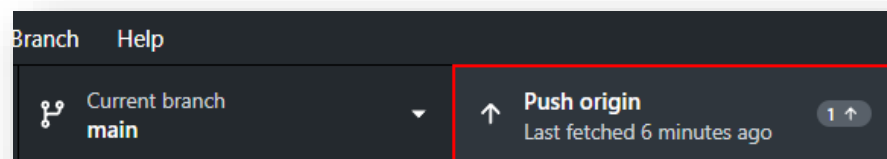
The next thing to do is to add your project files to your cloned repository folder. It should look like this:

.git	7.8.2022 г. 12:33	Папка с файлове	
.gitignore	7.8.2022 г. 12:33	Текстов документ	2 КБ
LICENSE	7.8.2022 г. 12:33	Файл	2 КБ
random_sentence_generator	7.8.2022 г. 12:23	JetBrains PyCharm	1 КБ

After that go to GitHub Desktop and **create a commit**, just like this.



Then **push the commit** to the repository.



This is all you need to **update** your **repository** using **GitHub Desktop**.

4. * Modify the Code, Write Your Own Features

Now, it's time to **play with the code** and **modify it**.

	<p>This is your own project. Be unique. Don't be a copy/paster!</p> <ul style="list-style-type: none">• Implement your own features.• Implement the code yourself, using your own coding style, code formatting, comments, etc.• Make the project more interesting. Learn by playing with the code and adding your own changes.
--	---

Below are a few **ideas** of what you can implement or modify as an addition to your code.

Add More Words

You can think of **more words to add** to make the sentences more interesting and fun.

Try Different Sentence Structures

You can **change your sentence** and make it more complex:

- You can turn your **sentence into a question**: ["Who" question word/phrase] + [Verb] + [Subject] + [Main Verb] + [Object or Other Information].
- You can add **more sentence parts** in the right places or **change the place of the current ones**.
- You can think of more ways to change your sentence.

Additional Ideas

- Consider a way to create a more **complex sentence generator**.

- Example of a more complex generator: <http://lomacar.github.io/Random-Sentence-Generator>.
- You can add anything else in your code, based on your ideas.

Commit to GitHub

Now **commit** and **push** your code changes to your GitHub repo!

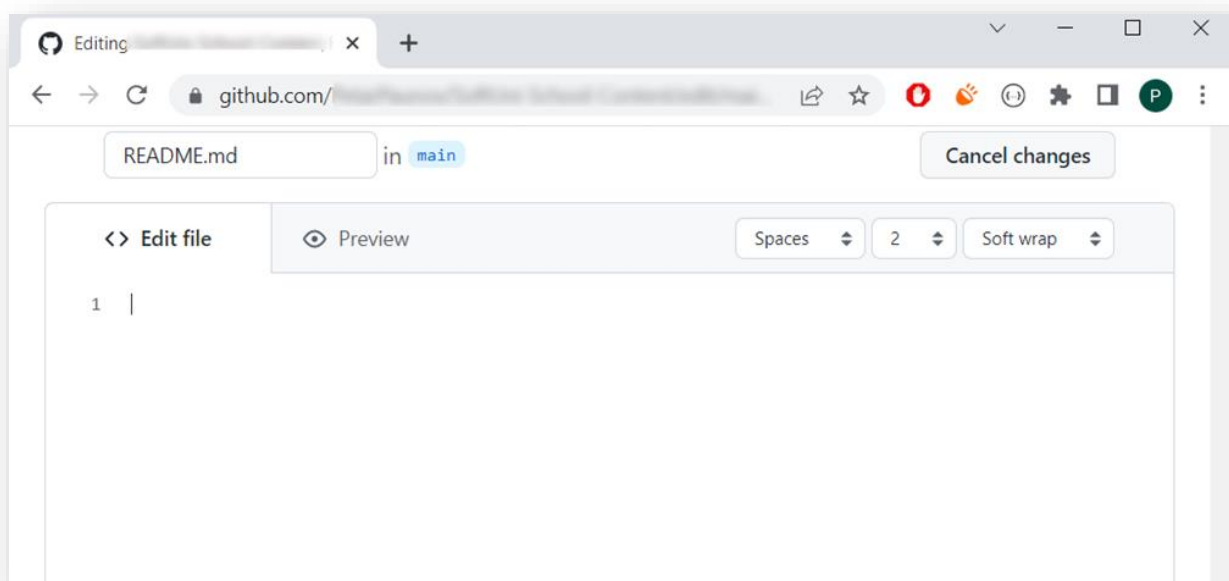


It is very important to **commit frequently** your code to GitHub. This way you create a **rich commit history** for your project and your GitHub contribution graph will grow:

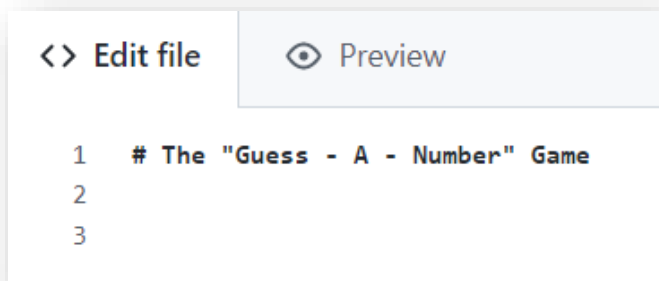


5. Create a README.md File

It's highly recommended to provide documentation as part of your 'project on GitHub to describe what the project is doing. So, let's make one for this **project**. Let's start by editing the **README.md** file from our repo on GitHub:



Add a project name. Use **"#"** in front of the text to indicate the **title**:



You can **view** the current progress by pressing the [**Preview**] button:

Documentation Sections

Add **information** about your project in your **README.md** file: project goals, technologies used, screenshots, live demo, etc. Typically, you should have the following **sections**:

- **Project title** (should answer the question "What's inside this project")
- **Project goals** (what problem we solve, e. g. we implement a certain game)
- **Solution** (should describe how we solve the problem → algorithms, technologies, libraries, frameworks, tools, etc.)
- **Source code link** (give a direct link to your source code)
- **Screenshots** (add screenshots from your project in different scenarios of its usage)
- **Live demo** (add a one-click live demo of your code)

Use Markdown

Note that the GitHub **README.md** file is written in the **Markdown language**. Markdown combines text and special formatting tags to describe formatted text documents.

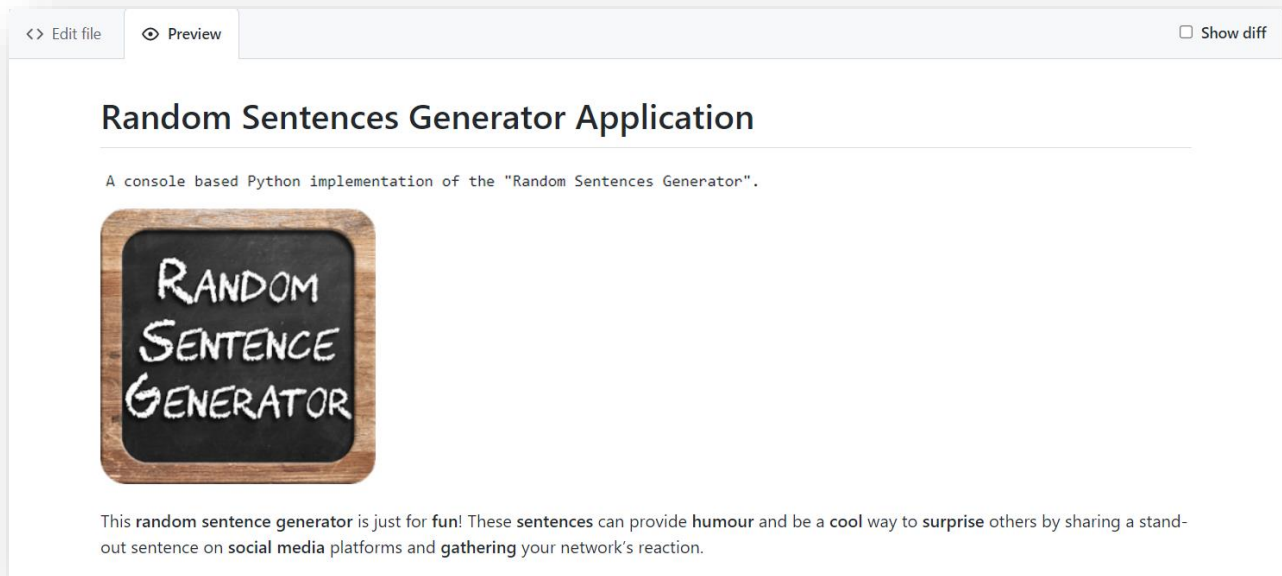
You can learn more about **Markdown** [here](#).

Project Goals

Start your documentation by describing your **project goals**. What problem does your project solve?

Sample Documentation

This is an **example** of how you can document your project. Don't copy-paste it!



Write the project documentation yourself. Don't copy/paste it!

This is your **unique GitHub profile** and your unique project. **Be different** from others.

Find an **appropriate image** and add it. You can add **images** as follows:

```
 />
```

Your Solution

Describe how you **solve the problem**: algorithms, technologies, libraries, frameworks, tools, etc:

Solution

The **Generator** is based on the following **model**:

- [Sentence] = Who + Action + Details .
 - Who = Name | Name from Place
 - Names = {Peter, Michell, Jane, Steve, ...}
 - Places = {Sofia, London, New York, Germany, ...}
 - Action = Verb + Noun | Adverbs + Verb + Noun
 - Verbs = {eats, holds, sees, plays with, brings, ...}
 - Nouns = {stones, cakes, apples, laptops, bikes, ...}
 - Adverbs = {slowly, diligently, warmly, sadly, rapidly}
 - Details = {near the river, at home, in the park}

You can use the **backtick** (```) at the **start** and **end** of the **word** to make it **grey**:

```
`Who` + `Action` + `Details`.
```

You can also use the **double-asterisk** (`**`) at the **start** and **end** of the word to **bold** it:

```
**Who** = `Name` | `Name` from `Place`
```

Link to the Source Code

Add a **link** to your **source code** as follows:

```
[Source Code](random_sentence_generator.py)
```

Screenshots

Add **screenshots** of your project:

1. **Take a screenshot** with your favorite tool (e.g., Windows).
2. **Paste** the screenshot in the GitHub Markdown editor, using **[Ctrl+V]**:

Example screenshots for the "**Random Sentences Generator**" game:

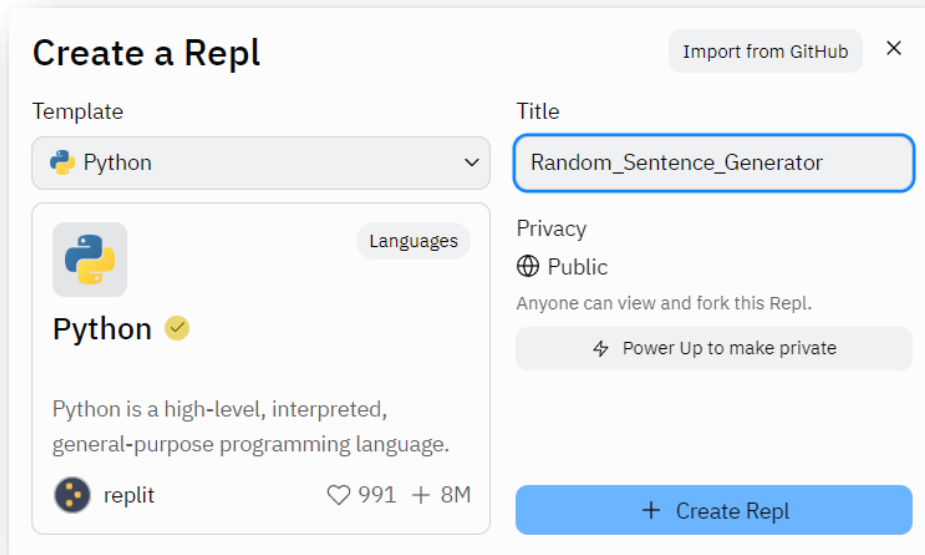
```
Hello, this is your first random sentence:  
Michell from Varna slowly sees cake  
Click [Enter] to generate a new one.  
Jane from Varna happily brings stones  
Click [Enter] to generate a new one.
```

```
Hello, this is your first random sentence:  
Jane from Sofia rapidly eats cake  
Click [Enter] to generate a new one.
```

6. Upload Your App to Replit

You already should have a **Replit** profile. Now let's add our **project** there so we can share it with our **friends** and add it to our **GitHub** profile. You already should know how to do that.

Open the **menu** in the upper **left corner**. Click "**Create**", then select the **language** in which your project is **written**, select a name, and **create** the project. Choose Python.




Create a Repl Import from GitHub ×


Template: Python

Title: Random_Sentence_Generator

Privacy: Public
Anyone can view and fork this Repl.
[Power Up to make private](#)

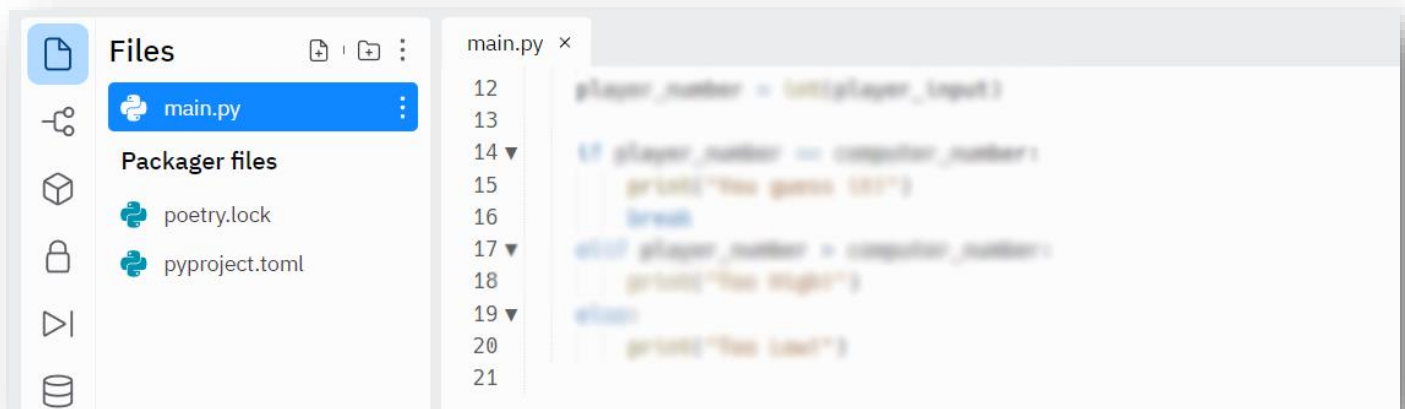
Python  Languages

Python is a high-level, interpreted, general-purpose programming language.

 replit ♡ 991 + 8M

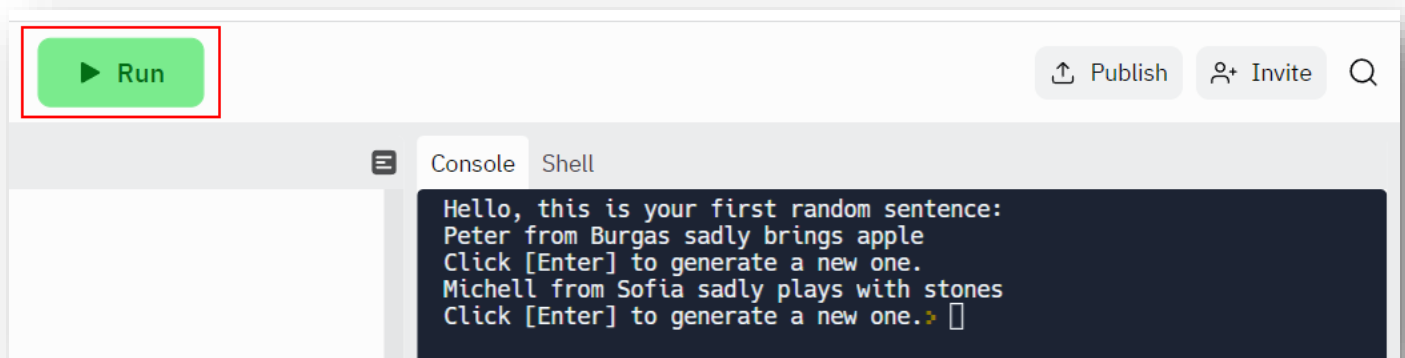
[+ Create Repl](#)

Paste your code in the "main.py" file:



```
12 player_number = int(player_input)
13
14 if player_number == computer_number:
15     print("You guess is!")
16     break
17 elif player_number > computer_number:
18     print("Too high!")
19 else:
20     print("Too low!")
21
```

Click **[Run]** and enjoy your console application.



You can now **share** your app with your friends.

7. Add Replit Link to Your README.md

Now add a "**one-click live demo**" of your project from your **GitHub** project documentation. You can do it as follows:

```
## Live Demo
```

```
You can try the generator directly in your Web browser here:
```

```
[]  
(https://replit.com/@mihailovskiy/Random-Sentences-Generator#main.py)
```

You can take a **screenshot** from Replit.com and **paste it** into the GitHub documentation editor directly with **[Ctrl+V]**.

Now we have completed our **Random Sentences Generator** and we have a new **project** in our **GitHub** portfolio.