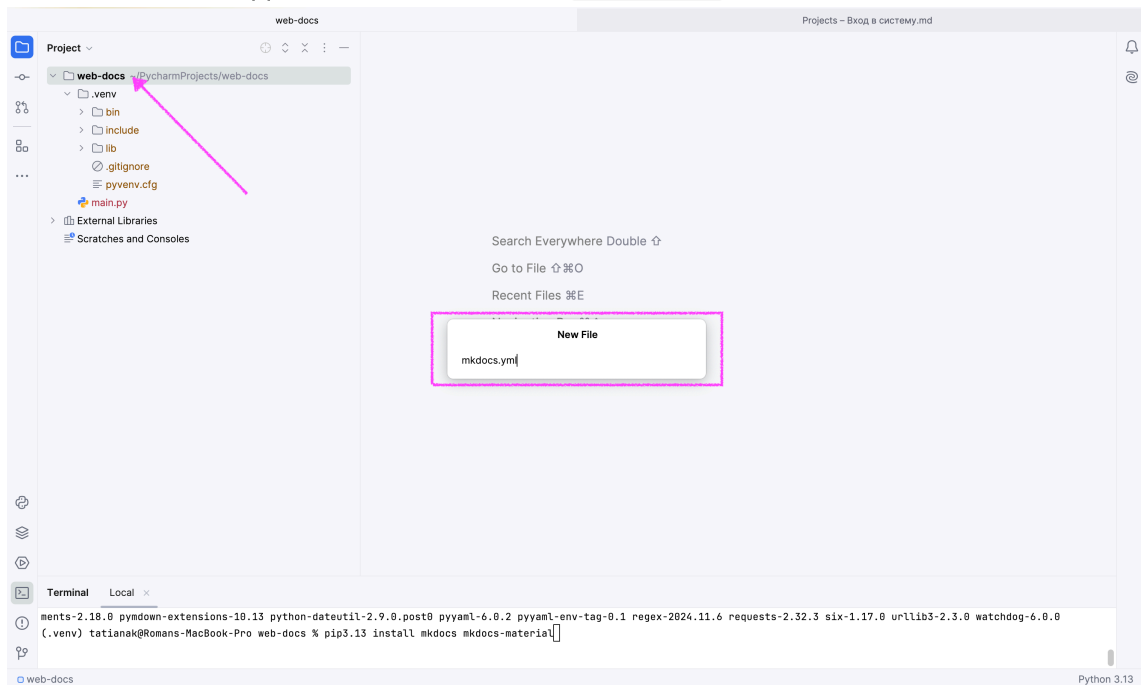


Step 5: Create MkDocs Configuration (mkdocs.yml)

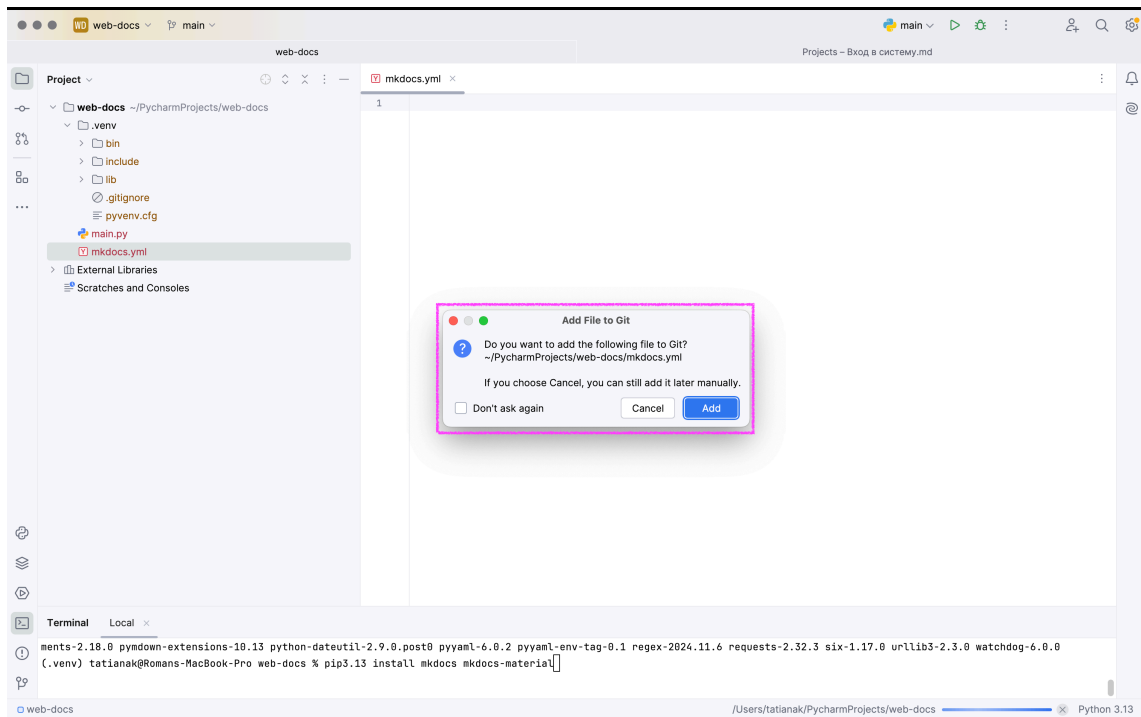
Create the `mkdocs.yml` file:

- In the root of the project, right-click the folder and choose "Create File".
- In the window that appears, enter the filename `mkdocs.yml`.



► To create a Yaml file, just add the `.yml` extension to the file name.

- Add the file to Git.



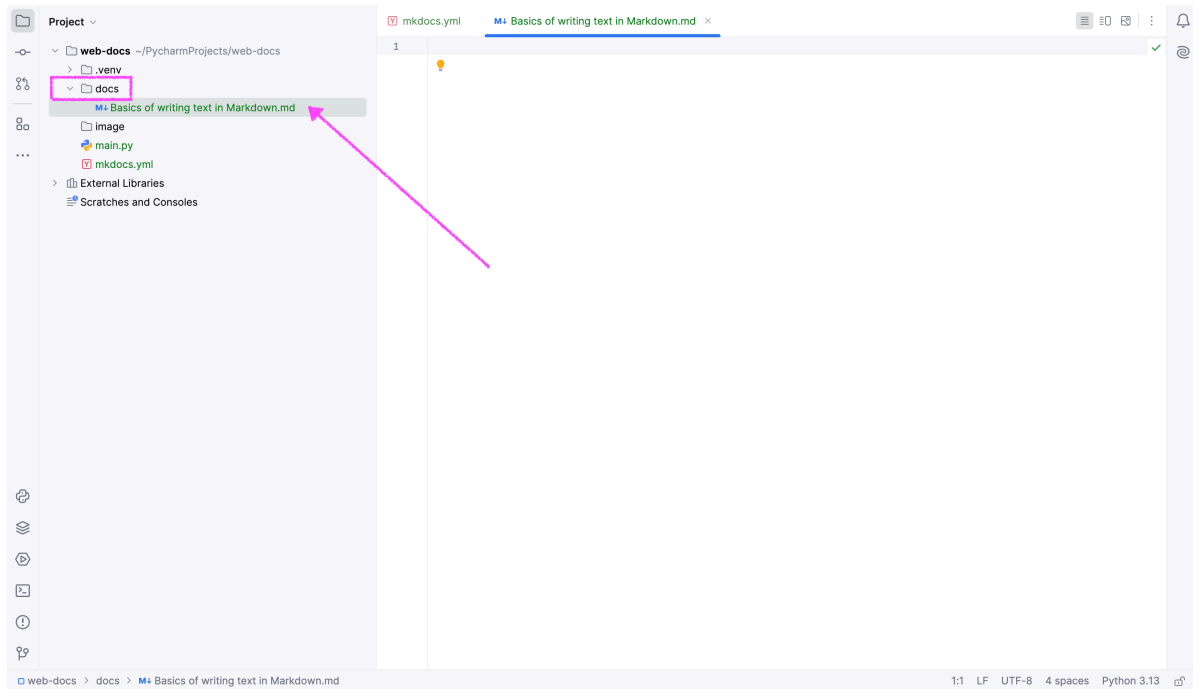
➤ After this step, you will have a `mkdocs.yml` file in your project tree.

- Create a **docs** folder for storing the source text files.
- Create an **image** folder for storing images.

As a result, your **project structure** will be scalable and look like this:

```
Web-docs/ # Parent directory
├── .venv/ # Virtual environment of the project
├── docs/  # Documentation source files
├── image/ # Documentation images
└── mkdocs.yml # MkDocs configuration file
```

Next, within the **docs** folder, create your first Markdown file.



► To create a Markdown file, just add the `.md` extension to the filename and start adding content according to the rules described below.

Customize your config (mkdocs.yml)

As a result of editing, your config may have the following structure.

```
# The name of your site, displayed in the page header
site_name: Web-docs using MkDocs

# The URL of your site after deployment (essential for correct link
functionality)
site_url: https://MkDocs-web-docs.github.io/how-to-create-web-docs/

# Specify an additional [CSS](# "Cascading Style Sheets: a style language for
describing the appearance of HTML documents, including colors, fonts, indents,
and other visual aspects") file for custom styles
extra_css:
  - extra.css

# Navigation (site page structure)
nav:
  - Creating a web manual using MkDocs: index.md      # The main page of the site
  - Links: Links.md                                  # Link to the Links page
  - Project Setup: Project Setup.md                   # Link to the Project Setup
page
  - Documentation Setup: Documentation Setup.md       # Link to the Documentation
Setup page
  - Deploy and Customize: Deploy and Customize MkDocs.md # Link to the Deploy and
Customize page

# Site theme and localization
theme:
  name: mkdocs      # The theme being used (in this case, the standard MkDocs
theme)
  locale: ru        # Localization (Russian language)

# Step 6: Create Documentation in Markdown Format
```

You need to include index.md in your navigation.

Basics of writing text in Markdown

Markdown is a simple markup language for formatting text. It allows you to create structured text using simple symbols.

Here are the main features:

1. Headings

Use the `#` symbol to create headings of different levels. The more `#` symbols, the lower the heading level.

Example: ``markdown # Heading Level 1 ## Heading Level 2 ### Heading Level 3

2. Paragraphs and Line Breaks

3. To separate paragraphs, leave a blank line.
4. To force a line break, add two spaces at the end of a line.

Example:

```
This is the first paragraph.  
  
This is the second paragraph.  
And this is the second line without a paragraph.
```

1. Text Formatting

2. Bold: Wrap text in double asterisks (**) or double underscores (). Example:********

```
**Bold text**  
__Bold text__
```

- Italics: Wrap text in single asterisks (*) or single underscores (). Example:******

```
*Italics*  
_Italics_
```

- Strikethrough: Use double tildes (~~). **Example:**

```
~~Strikethrough text~~
```

1. Lists

2. Bulleted list: Use -, , or +. *Example:******

```
- Item 1  
- Item 2  
  - Sub-item  
  ``  
  
* Numbered list: Use numbers with a period.  
**Example:**  
``markdown  
1. First  
2. Second
```

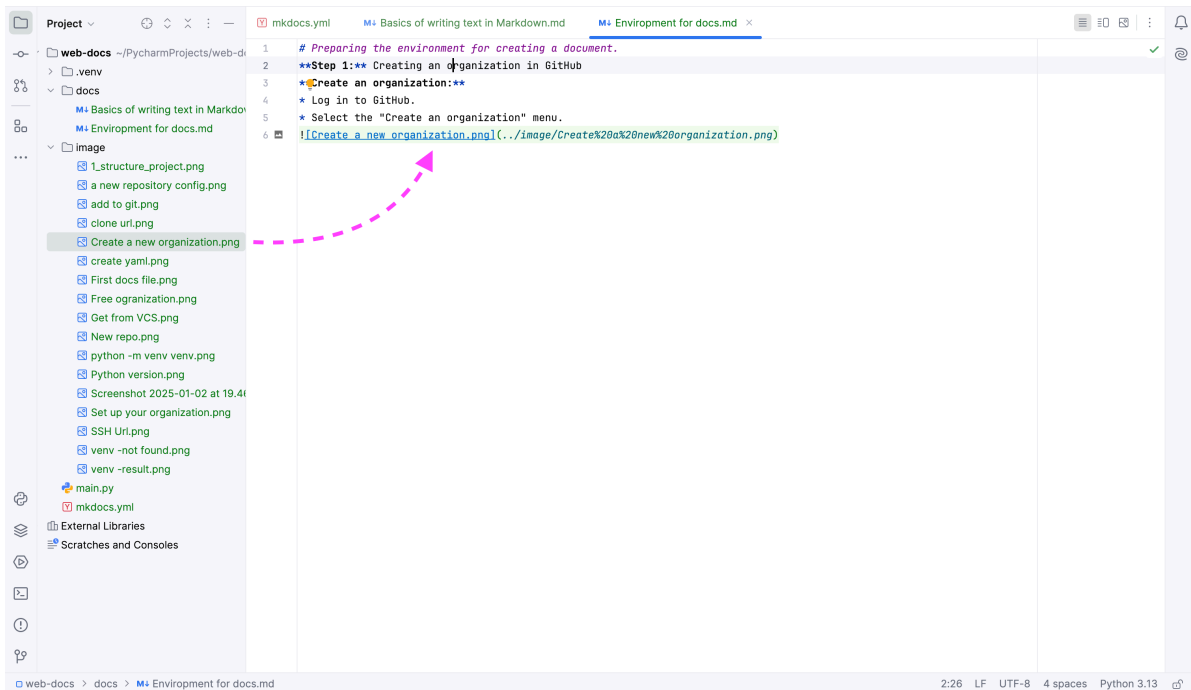
1. Links

```
[Link text](URL)
```

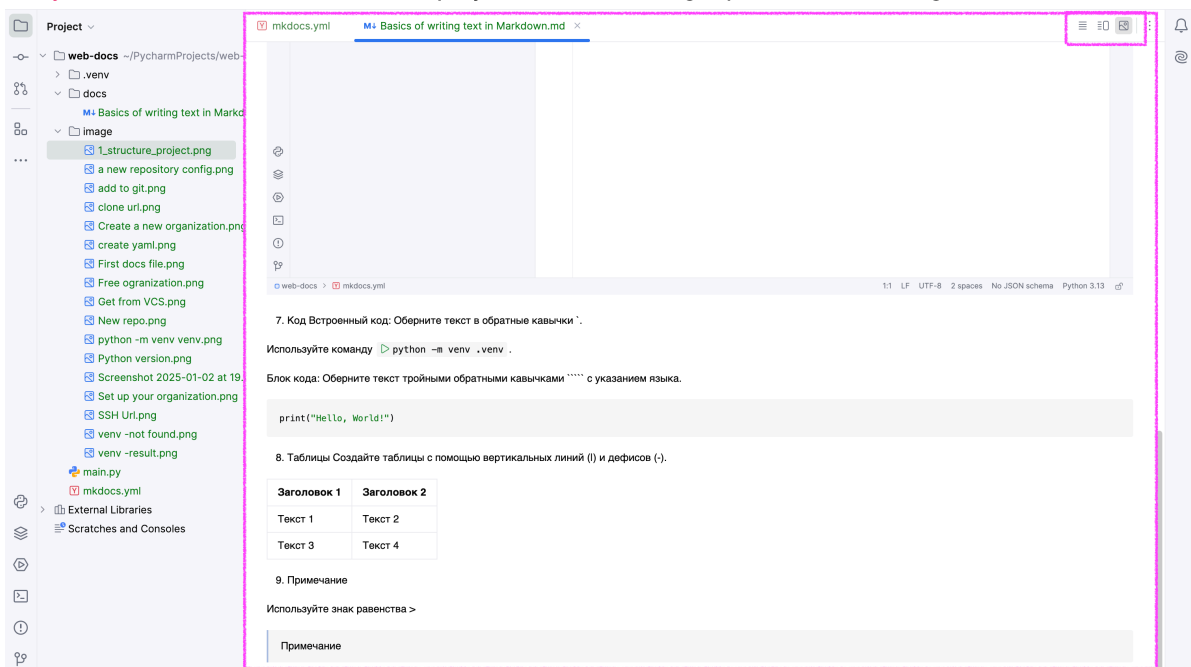
1. Images

```
![Image name](../image/1_structure_project.png)
```

► To add an image, you don't need to specify the path manually. Just drag the image from the left menu to the desired location, and **PyCharm** will automatically generate the path.



► **PyCharm** has three document display modes, including a preview with images.



7. **Code** * Inline code: Wrap the text in backticks (`). **Example:**

```
`python -m venv .venv`
```

- Code block: Wrap the text in triple backticks (```) followed by the language. **Example:**

```
```print("Hello, World!")```
```

1. **Tables** Create tables using vertical bars (|) and hyphens (-). **Example:**

```
| Heading 1 | Heading 2 |
|-----|-----|
| Text 1 | Text 2 |
```

1. **Note** Use the > symbol to create notes. **Example:**

```
> Note: Always test your code before deploying.
```

## More about numbering

To ensure that mixed numbering is displayed correctly in MkDocs (or any Markdown document), it is important to use proper indentation and formatting. Here is an example of how to do it:

!Indentation must be a multiple of 4 spaces.

1. First item
  - Unnumbered item inside a sub-item
    - Another unnumbered item
2. Second item
  - Unnumbered item in main list
    - Nested unnumbered item
      - Deeper nested unnumbered item

In **MkDocs**, as in **Markdown**, the correct display of nested numbering depends on the theme used and the way Markdown is interpreted. For standard Markdown, automatic support for "sub-items with nested numbers" (e.g. 2.1.1) may not be available, since Markdown does not support multi-part numbering by default.

Later we will look at working with other, more flexible, themes.

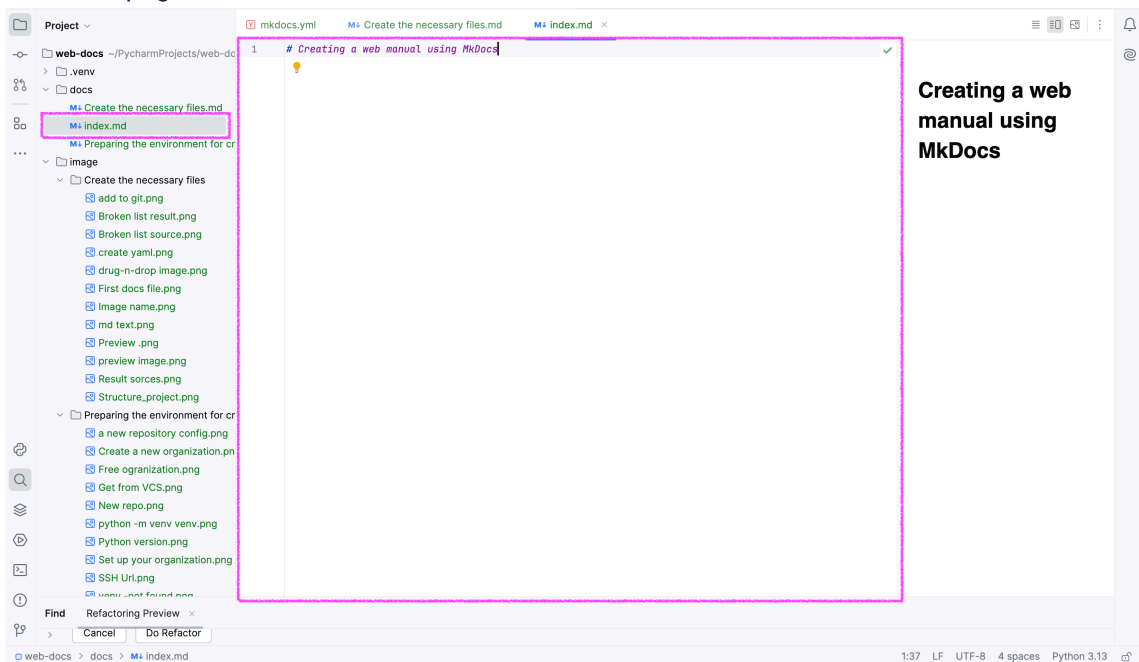
In some cases, the numbering may be broken. To fix this, I use HTML to continue the numbering:

```
<ol start="5">
 Links:
 Link text

```

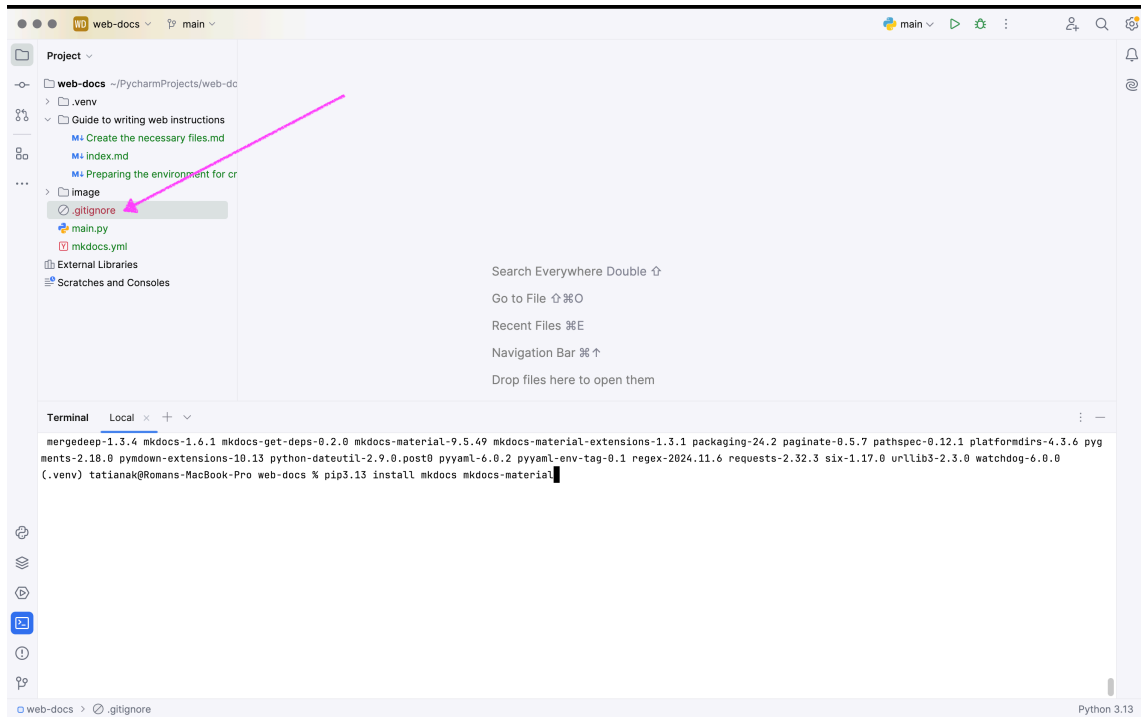
## Step 7: Complete the Structure of Your Project

1. In the source folder, create the file `index.md` — the text of this document will be displayed on the home page of the site.



2. Create the `.gitignore` file in the root of the project and write `site` in it — this will prevent Git from tracking the compiled files.





➤ The "site" folder will be automatically generated during deployment.

Now, your **project structure** looks like this:

```
Web-docs/ # Parent directory
├── .venv/ # Virtual environment
├── docs/ # Documentation source files
│ └── index.md/ # Home page
├── image/ # Documentation images
├── mkdocs.yml # MkDocs configuration file
└── .gitignore # Ignored files
```