



Mid Journey V4 prompt: a human standing in front of a winding road. The bright scene, adventurous, encouraging --ar 3:2

LECTURE 4-1: PYTHON BASICS III - PATH

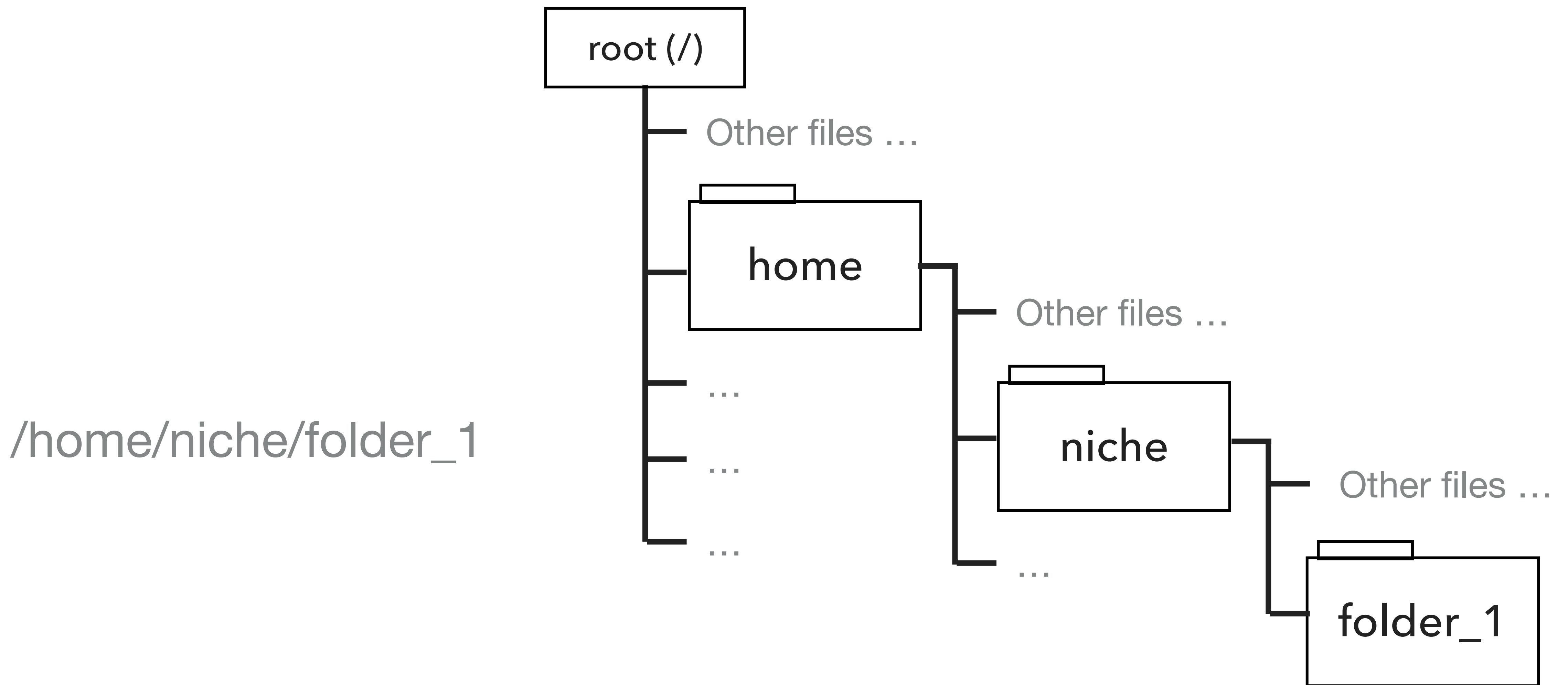
Dr. James Chen, Animal Data Scientist, School of Animal Sciences

2023 APSC-5984 SS: Agriculture Data Science



A path is a string that represents the location of a file or a directory (i.e., folder) in the file system.

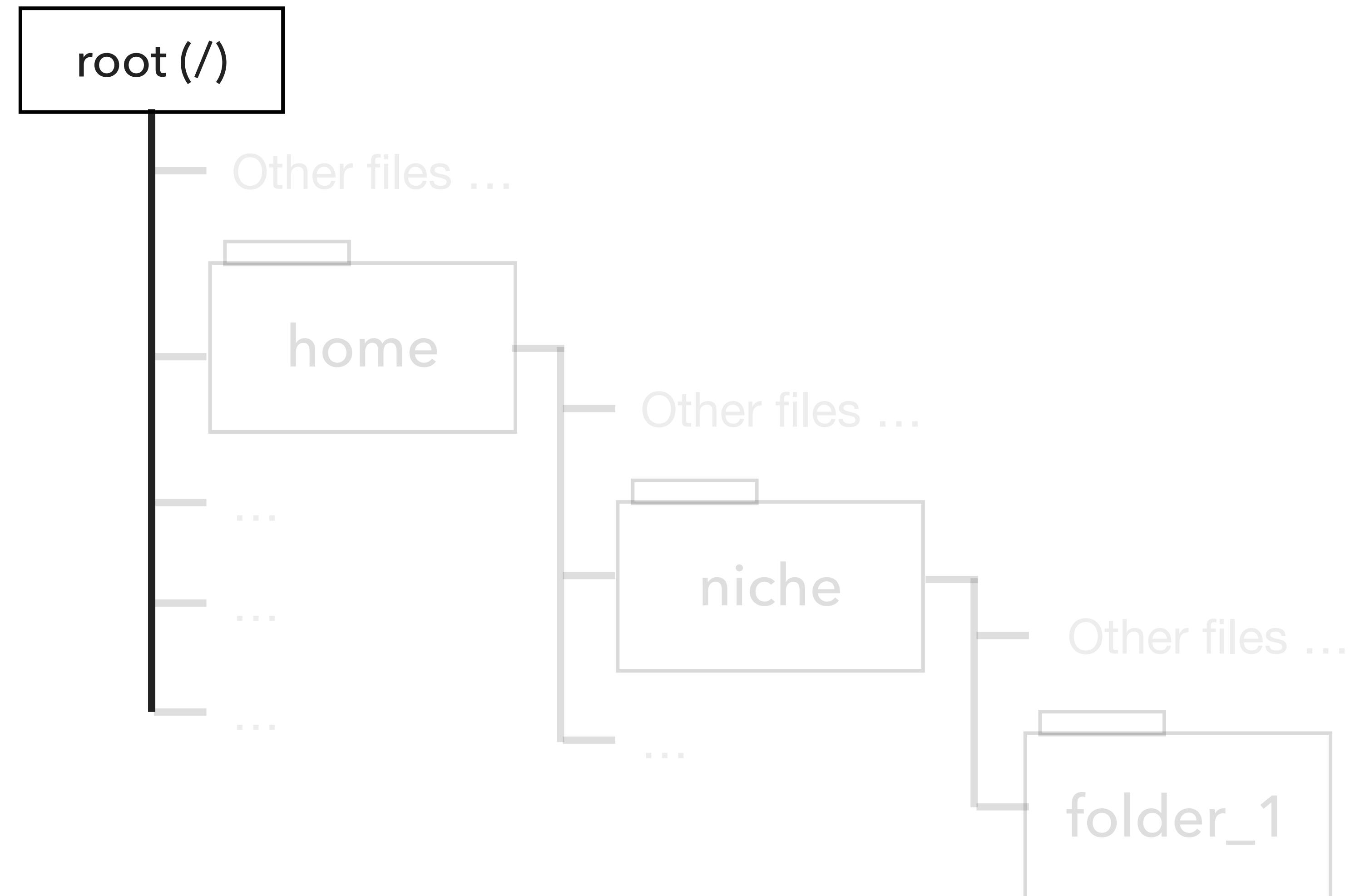
A path is a string that represents the location of a file or a directory (i.e., folder) in the file system.



A path is a string that represents the location of a file or a directory (i.e., folder) in the file system.

A “/“ at the beginning of the path refers to the root folder, which is the top-most directory in the file system.

`/home/niche/folder_1`

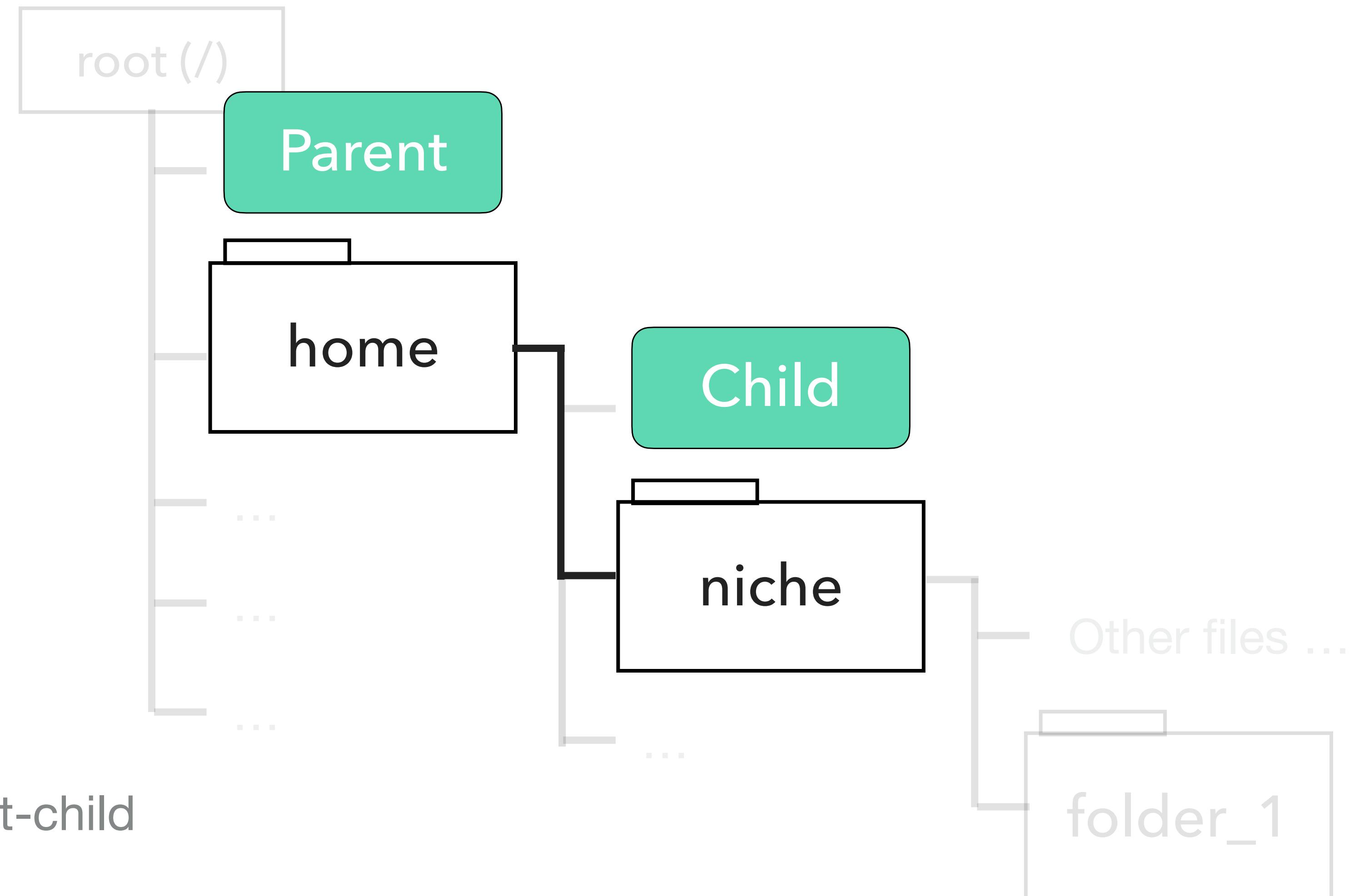


A path is a string that represents the location of a file or a directory (i.e., folder) in the file system.

A “/” at the beginning of the path refers to the root folder, which is the top-most directory in the file system.

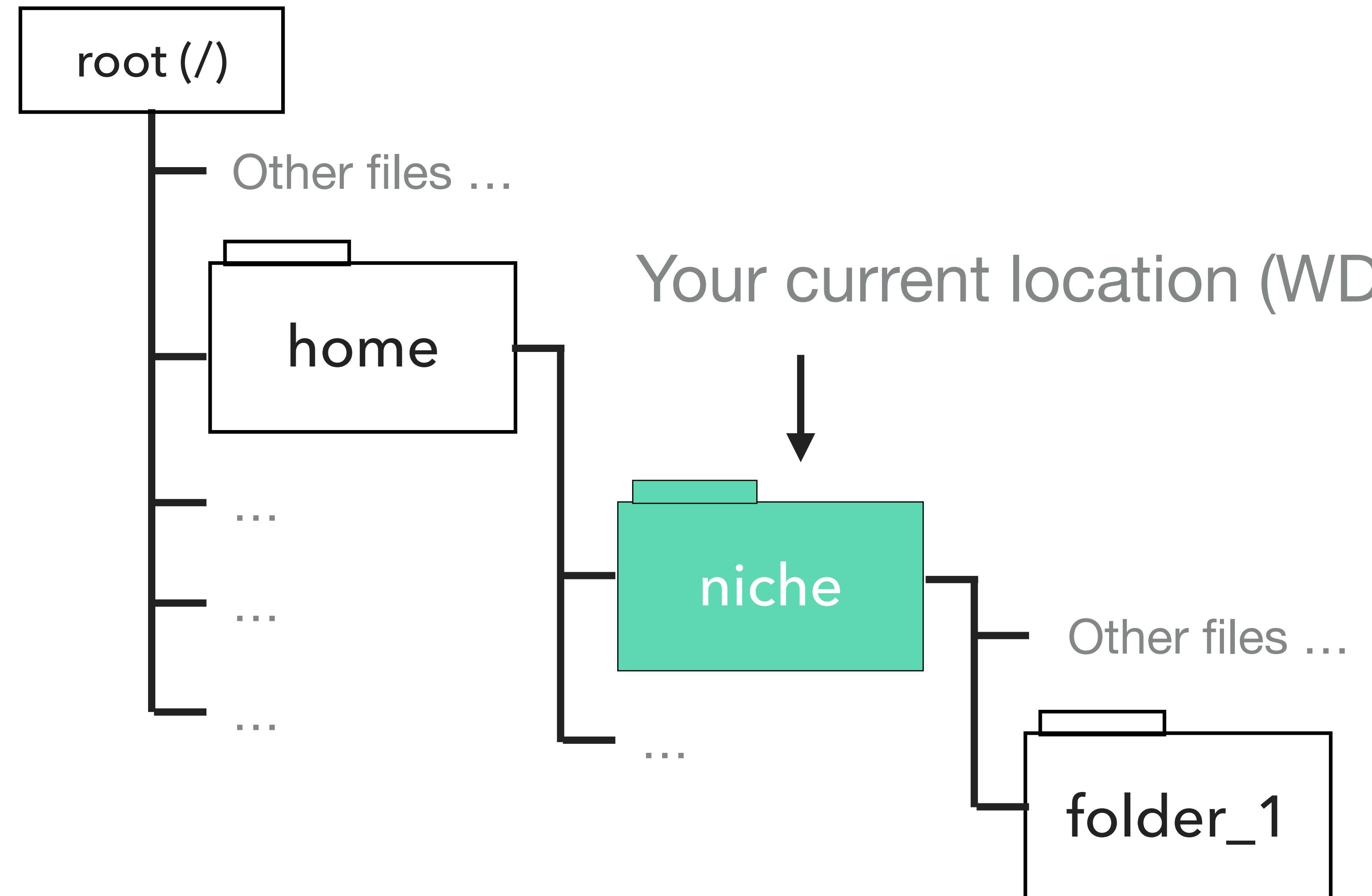
↑
/home/niche/folder_1

↓
A “/” after that represent the parent-child relationship between directories.

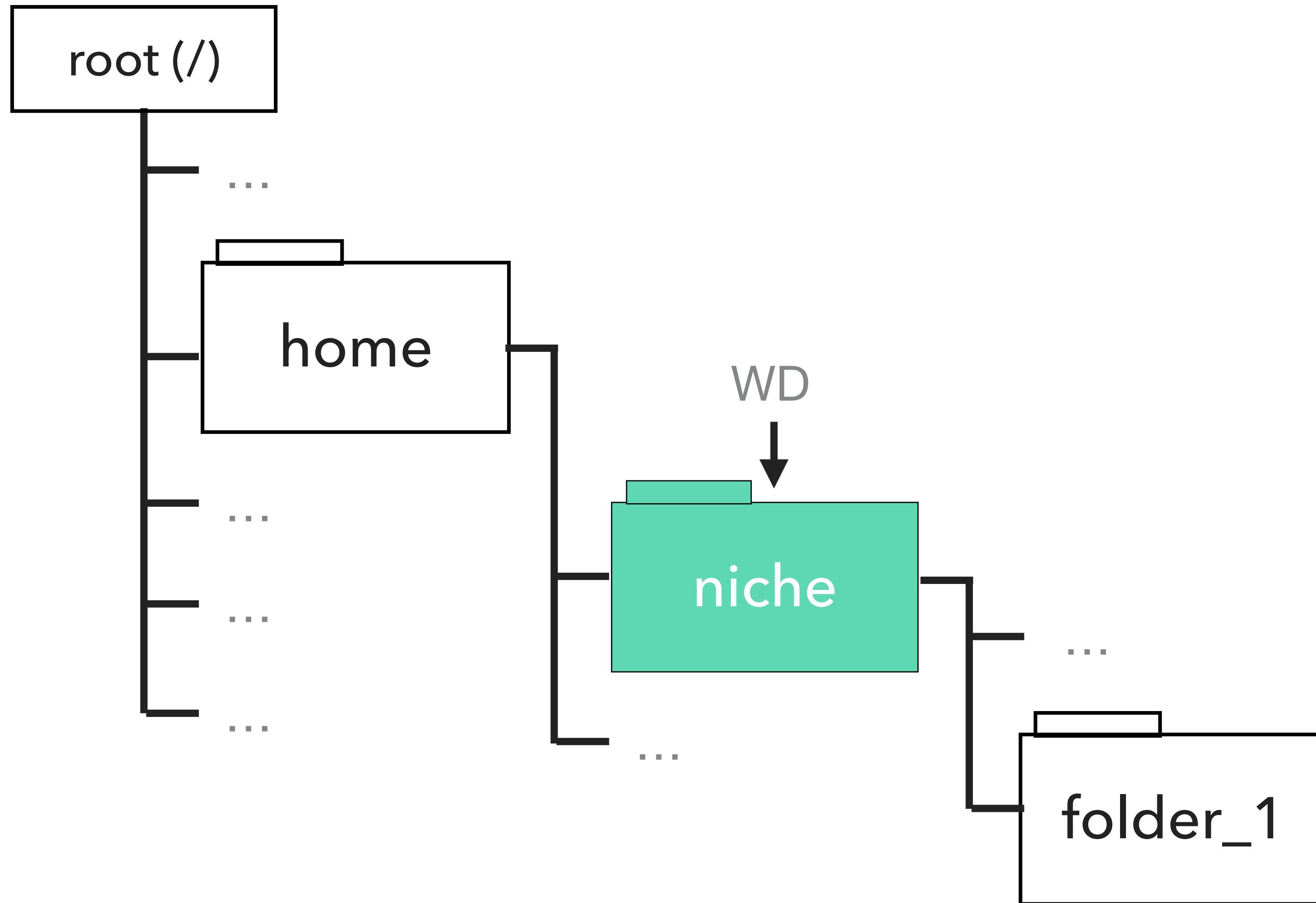


A working directory is a directory where you are currently working in.

When your WD is:
`/home/niche`



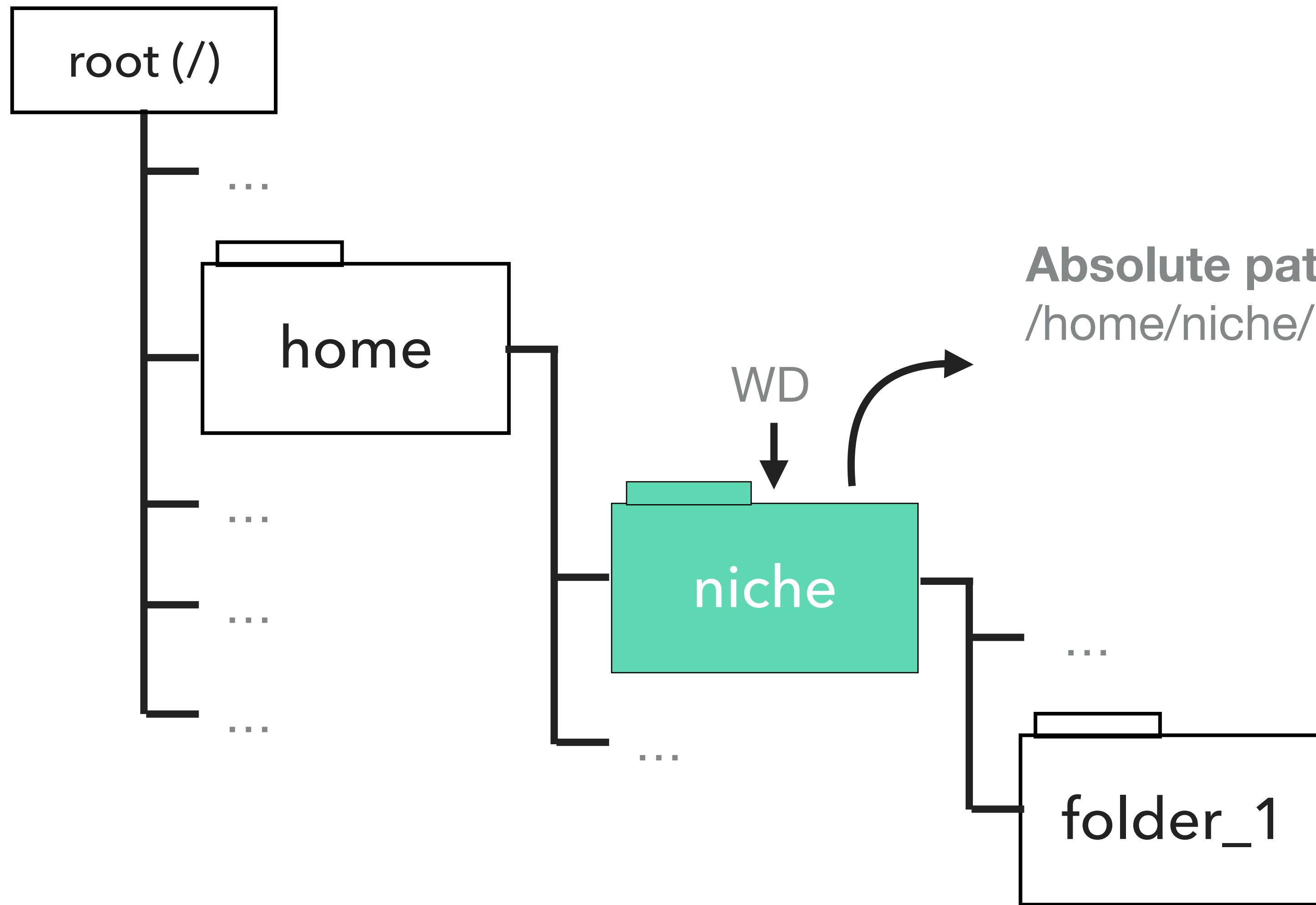
Absolute Path vs. Relative Path



Note:

- . -> current directory
- .. -> parent directory

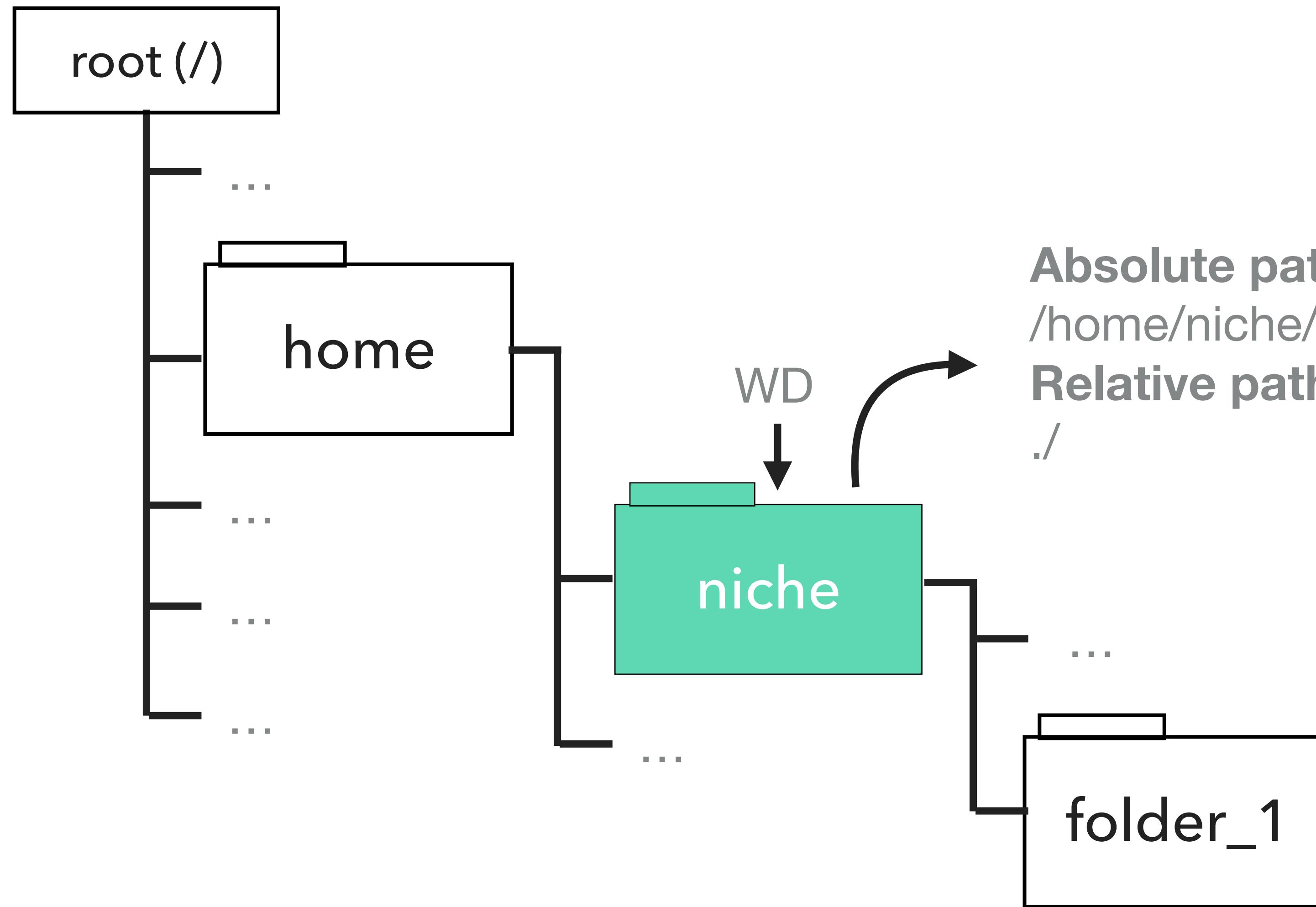
Absolute Path vs. Relative Path



Note:

- . -> current directory
- .. -> parent directory

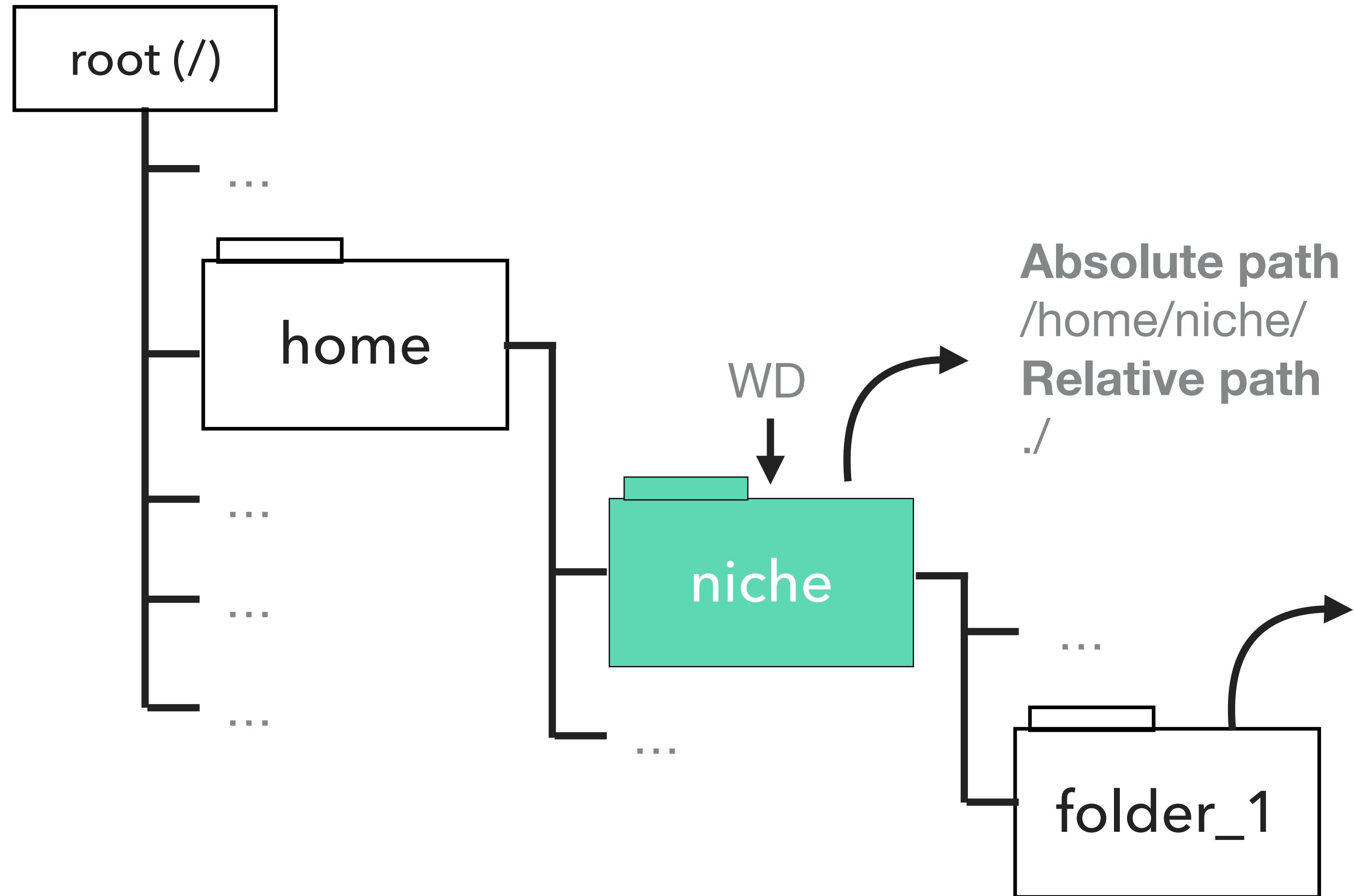
Absolute Path vs. Relative Path



Note:

- . -> current directory
- .. -> parent directory

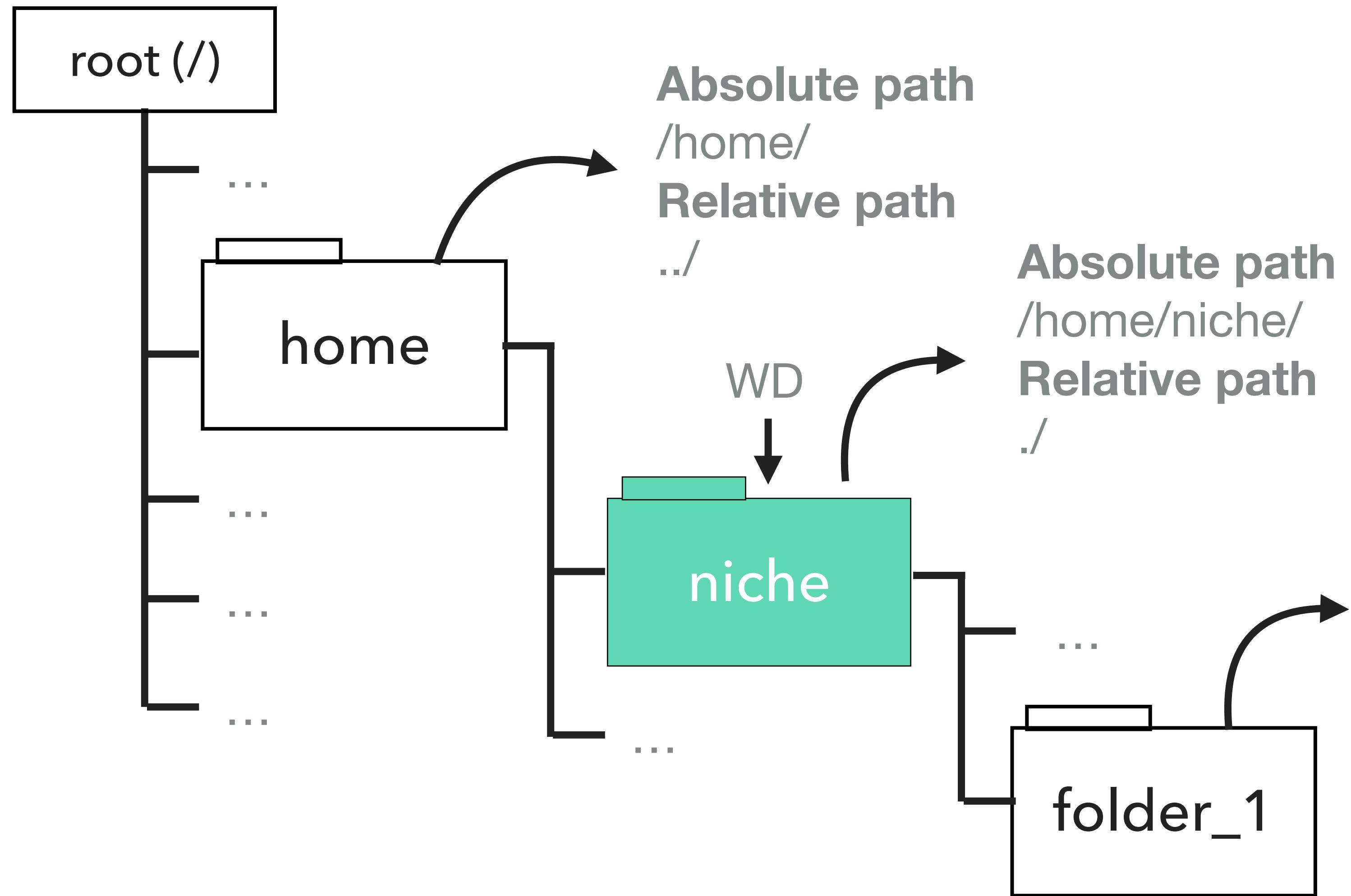
Absolute Path vs. Relative Path



Note:

- . -> current directory
- .. -> parent directory

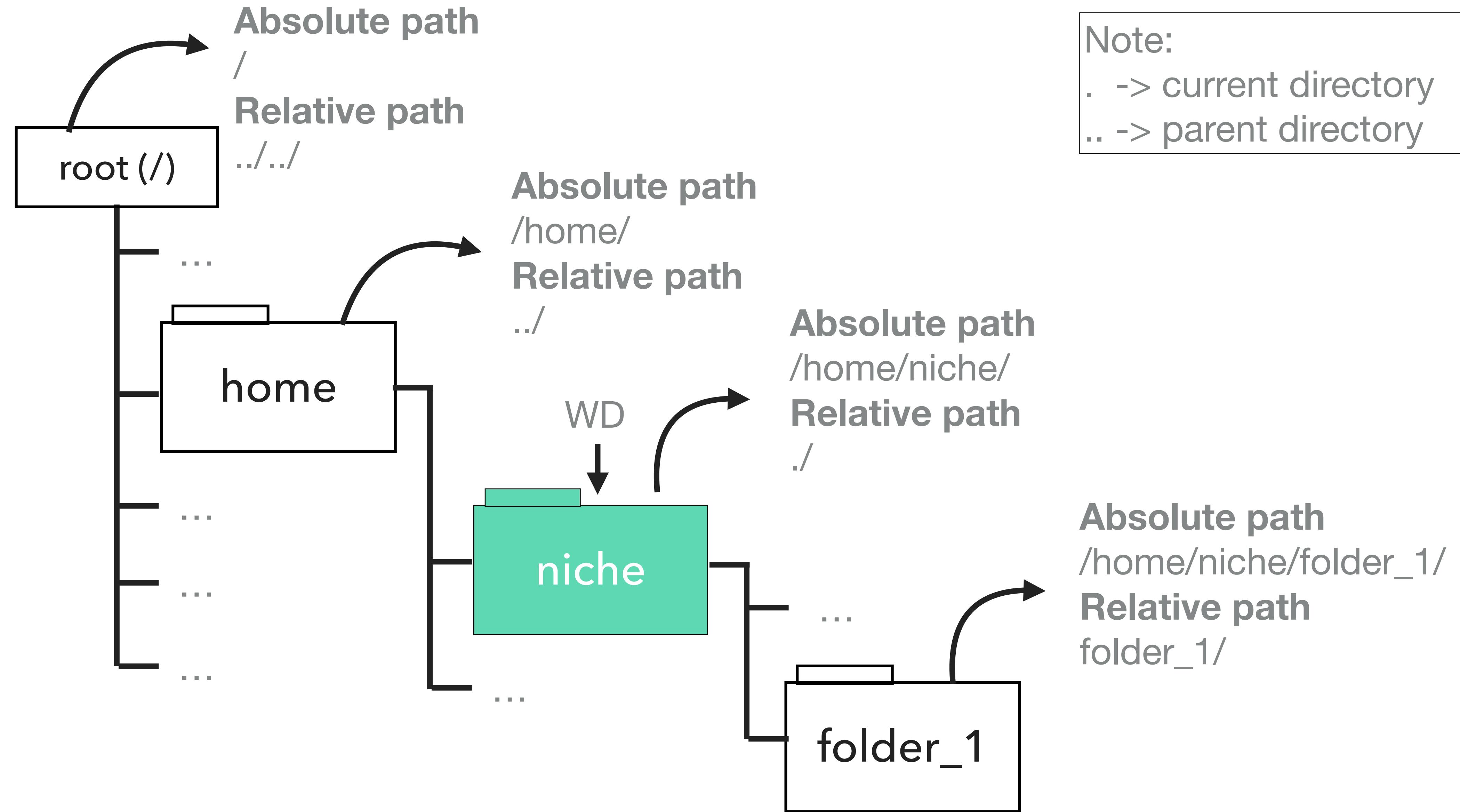
Absolute Path vs. Relative Path



Note:

- . -> current directory
- .. -> parent directory

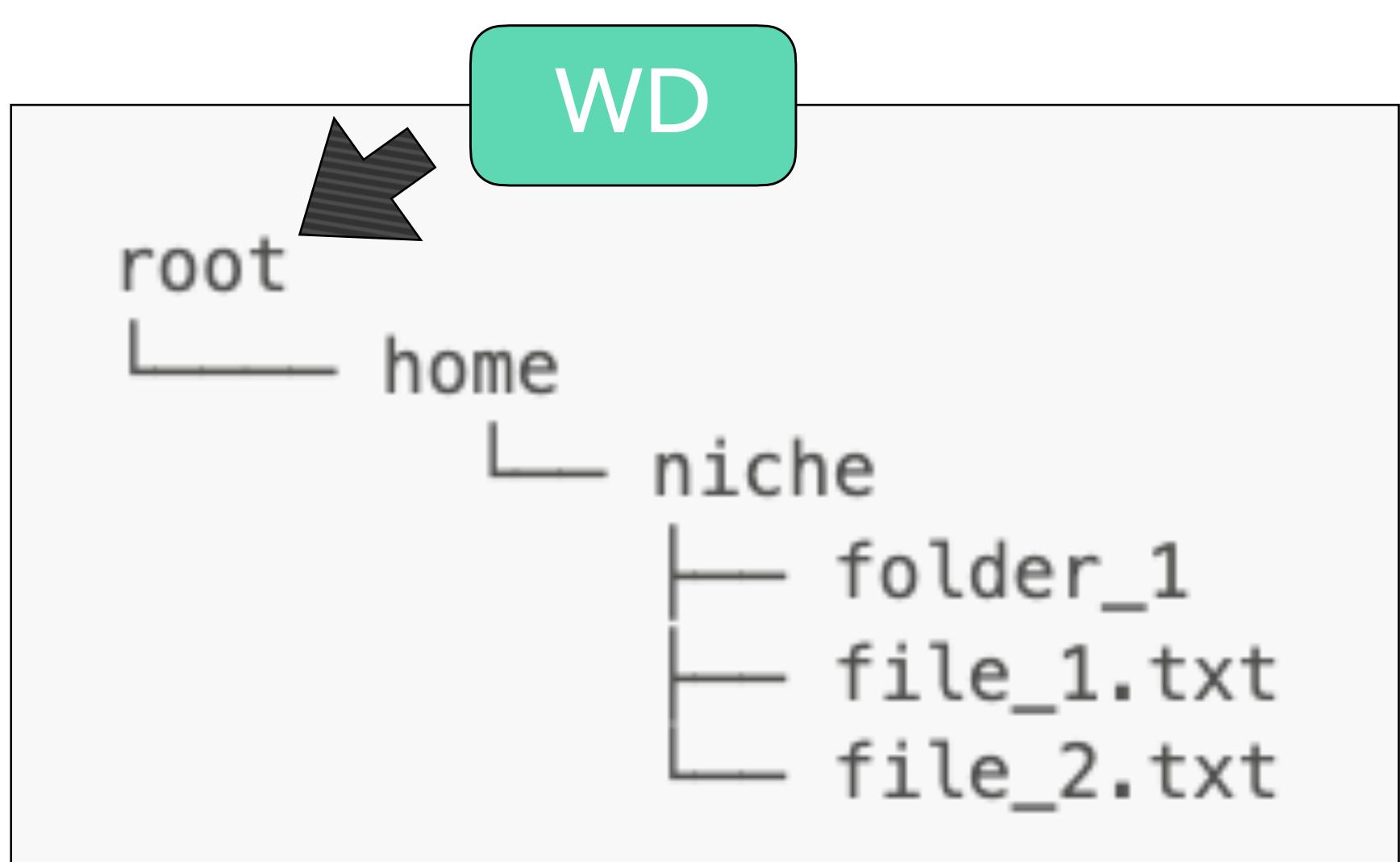
Absolute Path vs. Relative Path





Common commands you will use:

- `ls`: list the files and folders in the current WD.
- `pwd`: print the current WD.
- `cd`: change the current WD.
- `mkdir`: create a new directory.
- `.`: an alias for the current WD.
- `..`: an alias for the parent directory of the current WD.
- Tilde sign `~`: an alias for the home directory of the current user.



```
pwd  
# output: /
```

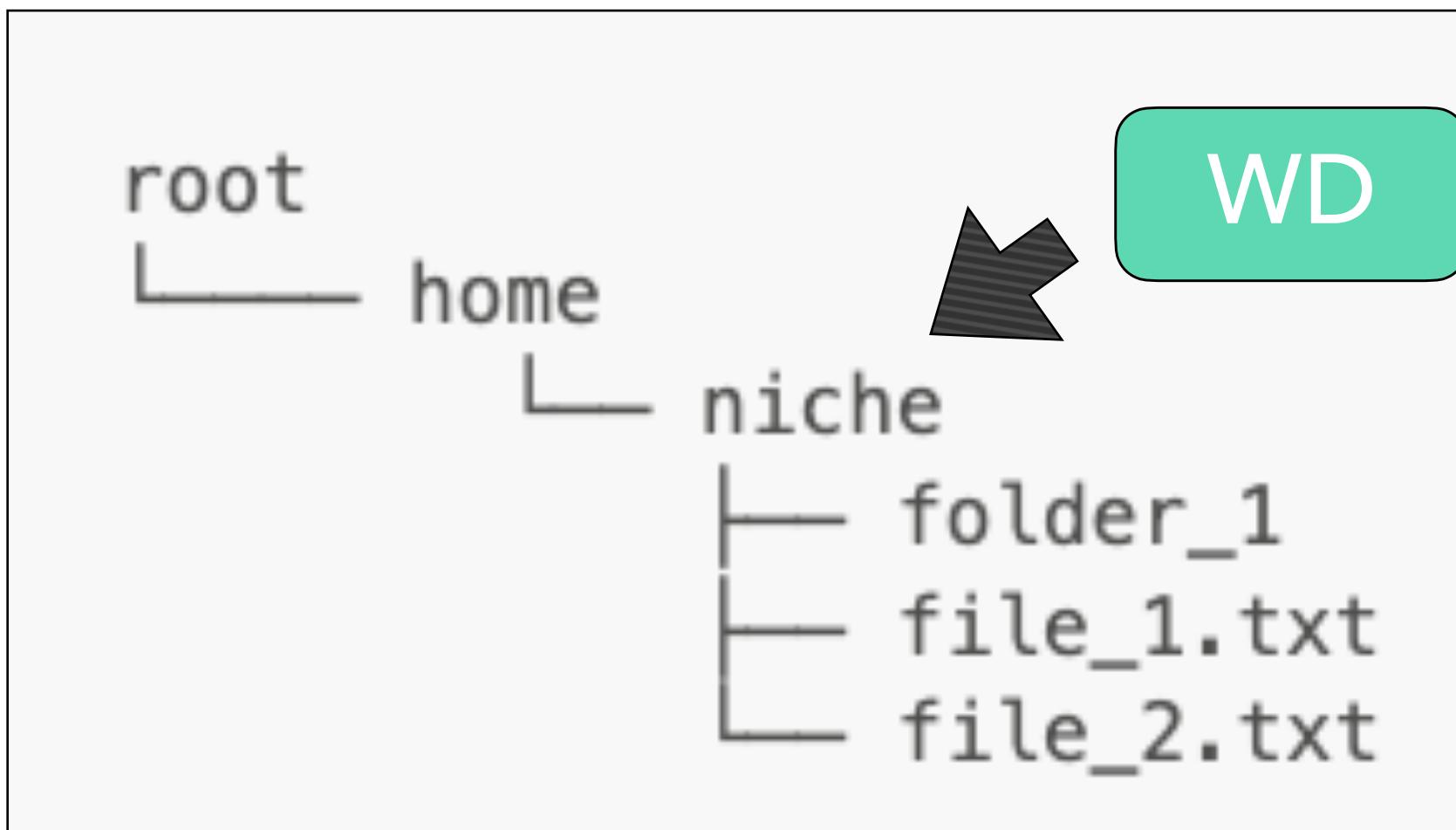
Print the WD

```
ls  
# output: home
```

List files in the WD

Common commands you will use:

- `ls`: list the files and folders in the current WD.
- `pwd`: print the current WD.
- `cd`: change the current WD.
- `mkdir`: create a new directory.
- `.`: an alias for the current WD.
- `..`: an alias for the parent directory of the current WD.
- Tilde sign `~`: an alias for the home directory of the current user.



Common commands you will use:

- **ls**: list the files and folders in the current WD.
- **pwd**: print the current WD.
- **cd**: change the current WD.
- **mkdir**: create a new directory.
- **.**: an alias for the current WD.
- **..**: an alias for the parent directory of the current WD.
- Tilde sign **~**: an alias for the home directory of the current user.

```
pwd
# output: /
```

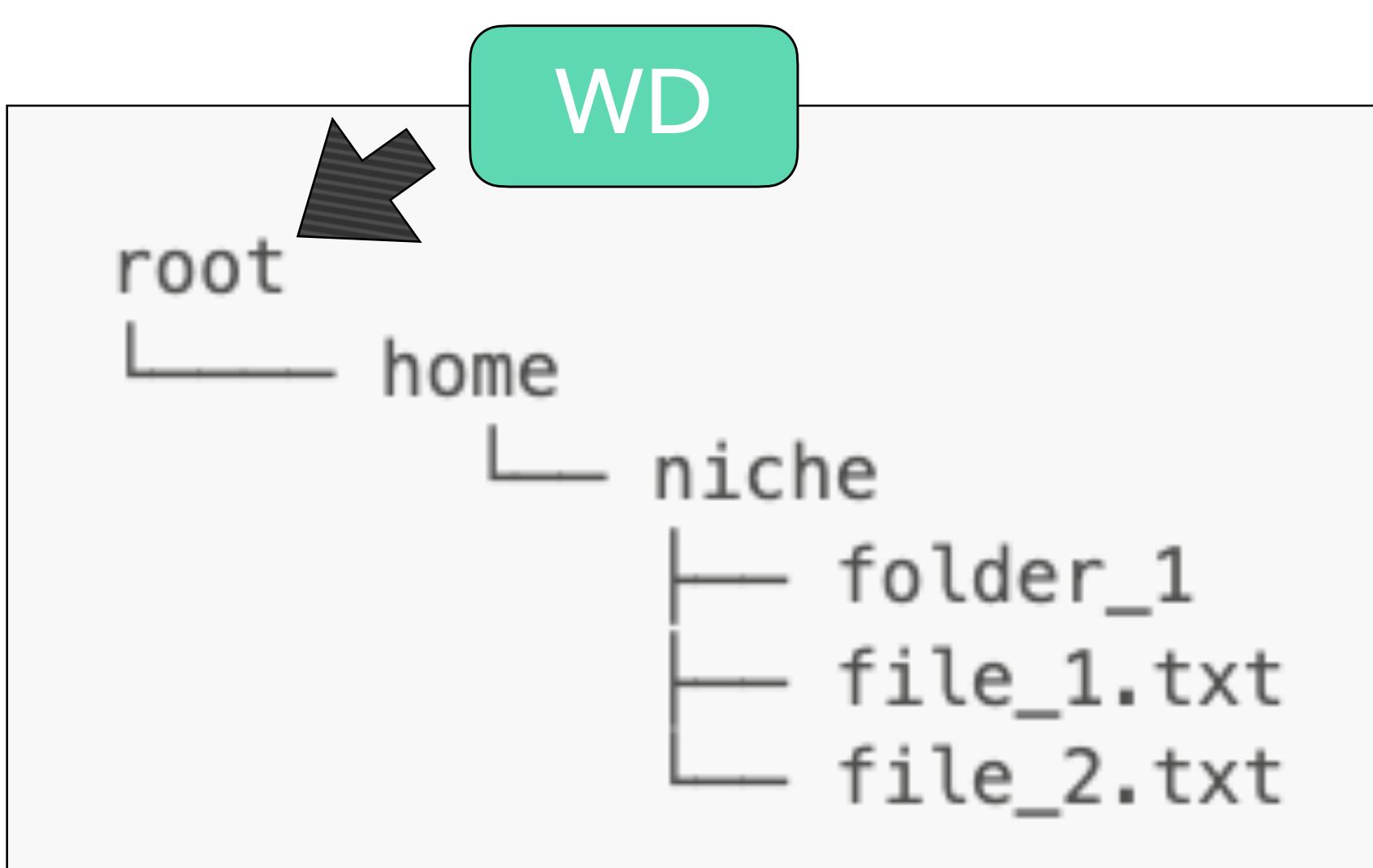
Print the WD

```
ls
# output: home
```

List files in the WD

```
cd home/niche
pwd
# output: /home/niche
```

Change WD to /home/niche and check the WD again



```
pwd  
# output: /
```

Print the WD

```
ls  
# output: home
```

List files in the WD

```
cd home/niche  
pwd  
# output: /home/niche
```

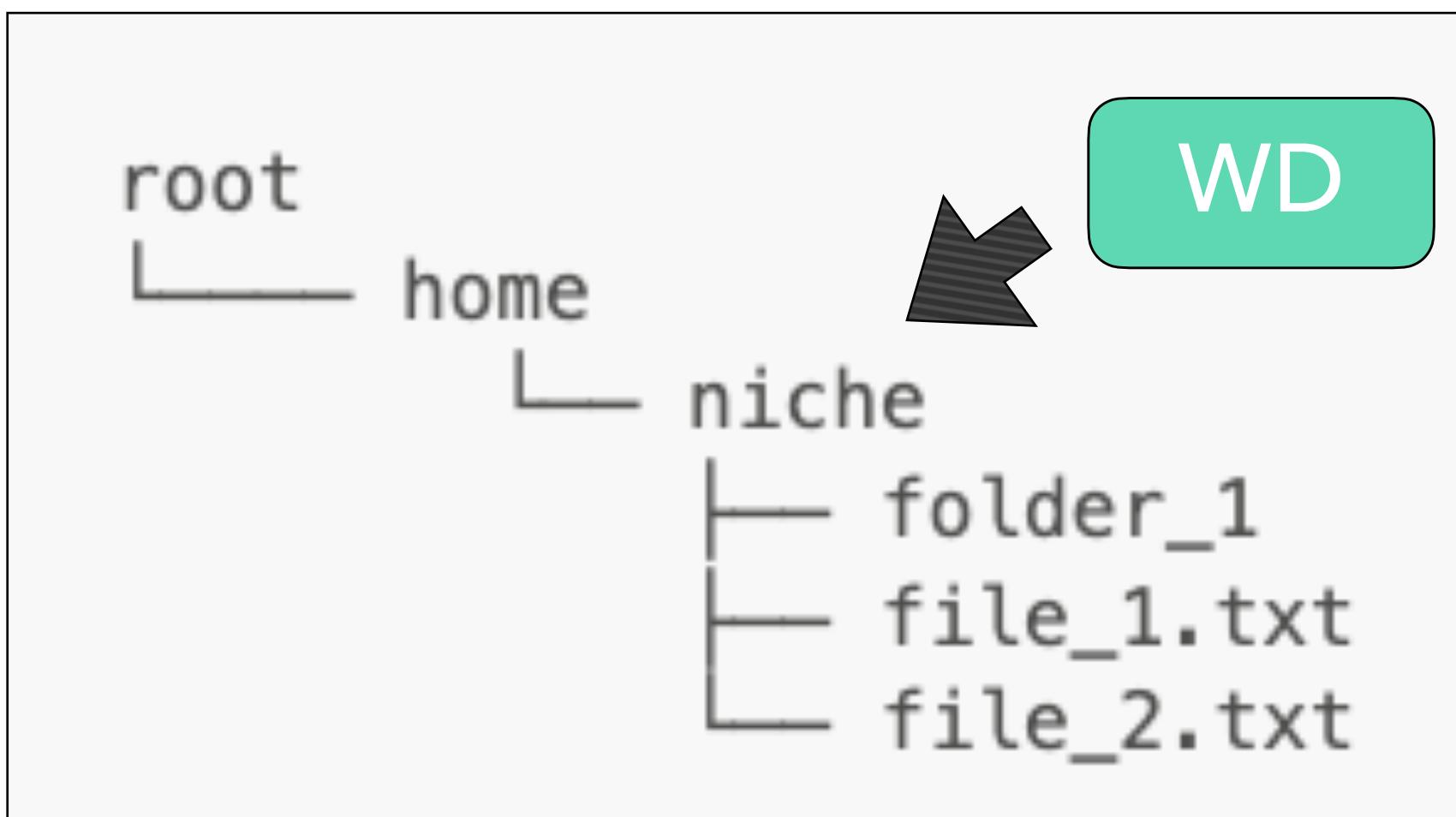
Change WD to /home/niche and check the WD again

Common commands you will use:

- **ls**: list the files and folders in the current WD.
- **pwd**: print the current WD.
- **cd**: change the current WD.
- **mkdir**: create a new directory.
- **.**: an alias for the current WD.
- **..**: an alias for the parent directory of the current WD.
- Tilde sign **~**: an alias for the home directory of the current user.

```
cd ../../  
pwd  
# output: /
```

Move back to the parent folders



```
import os
```

Here are the common Python `os` methods to interact with the file system:

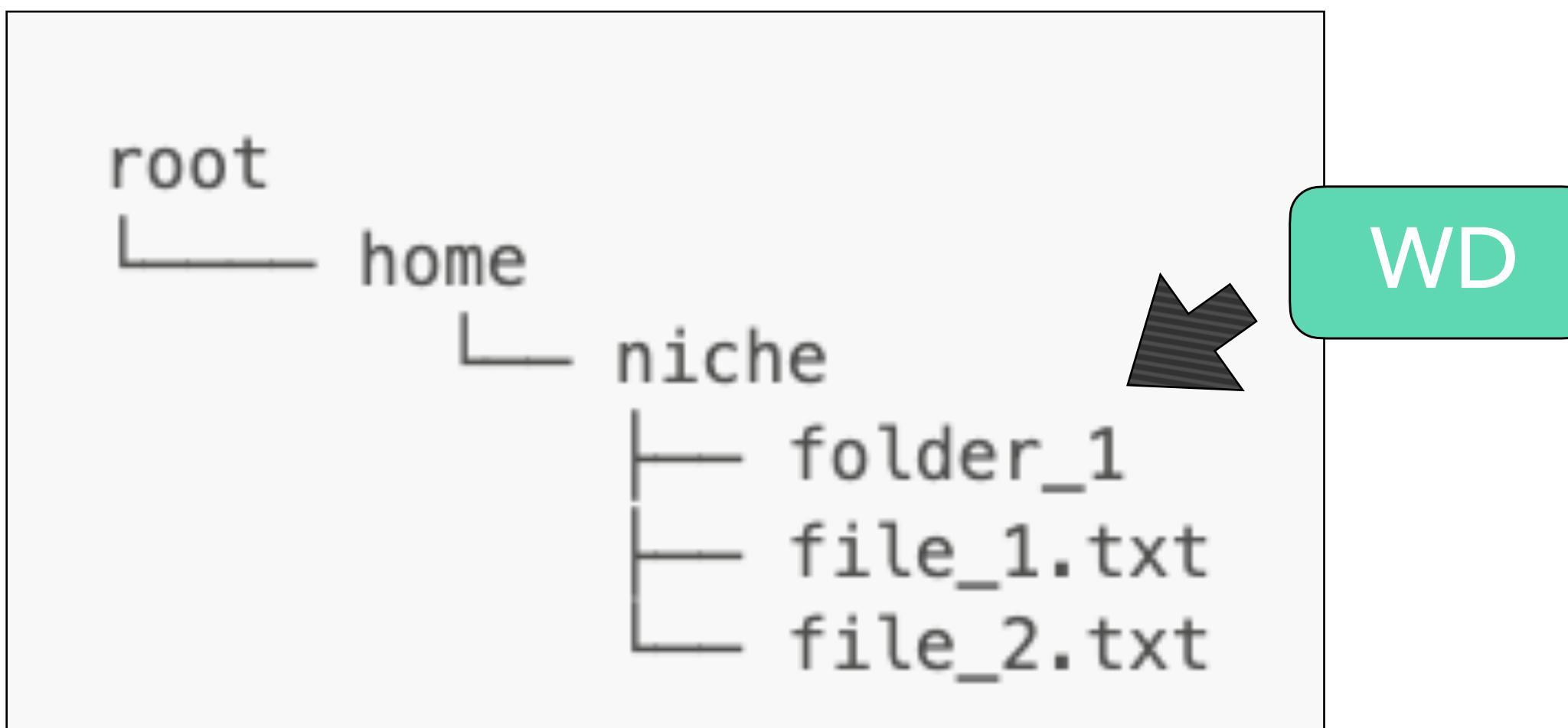
- `os.getcwd()`: get the current WD
- `os.listdir()`: list the content of a directory (default: WD)
- `os.chdir()`: change the WD

```
os.getcwd()
# output: '/home/niche'
```

Print the WD

```
os.listdir()
# output: ['file_1.txt', 'file_2.txt', 'folder_1']
```

List files in the WD



```
import os
```

Here are the common Python `os` methods to interact with the file system:

- `os.getcwd()`: get the current WD
- `os.listdir()`: list the content of a directory (default: WD)
- `os.chdir()`: change the WD

```
os.getcwd()
# output: '/home/niche'
```

Print the WD

```
os.listdir()
# output: ['file_1.txt', 'file_2.txt', 'folder_1']
```

List files in the WD

```
os.chdir('/home/niche/folder_1')
os.getcwd()
# output: '/home/niche/folder_1'
```

Change WD to /home/niche/folder_1
and check the WD again

The `os.path.join()` method is a convenient way to join multiple paths together. It is recommended to use over explicitly typing the path in a string is because it is OS-agnostic. For example, if you are using Windows, the path separator is `\` instead of `/`. Using `os.path.join()` will automatically adjust the path separator based on your OS.

```
os.path.join('home', 'niche', 'folder_1')
# output: 'home/niche/folder_1' if you are using Linux
# output: 'home\\niche\\folder_1' if you are using Windows
```