Red Hat

# Editors on Linux

## Text is our friend

# Editors on Linux

- Why care about text editors
- Default GUI editor
- Nano a terminal based editor
- Vi/Vim/NeoVim terminal based editors

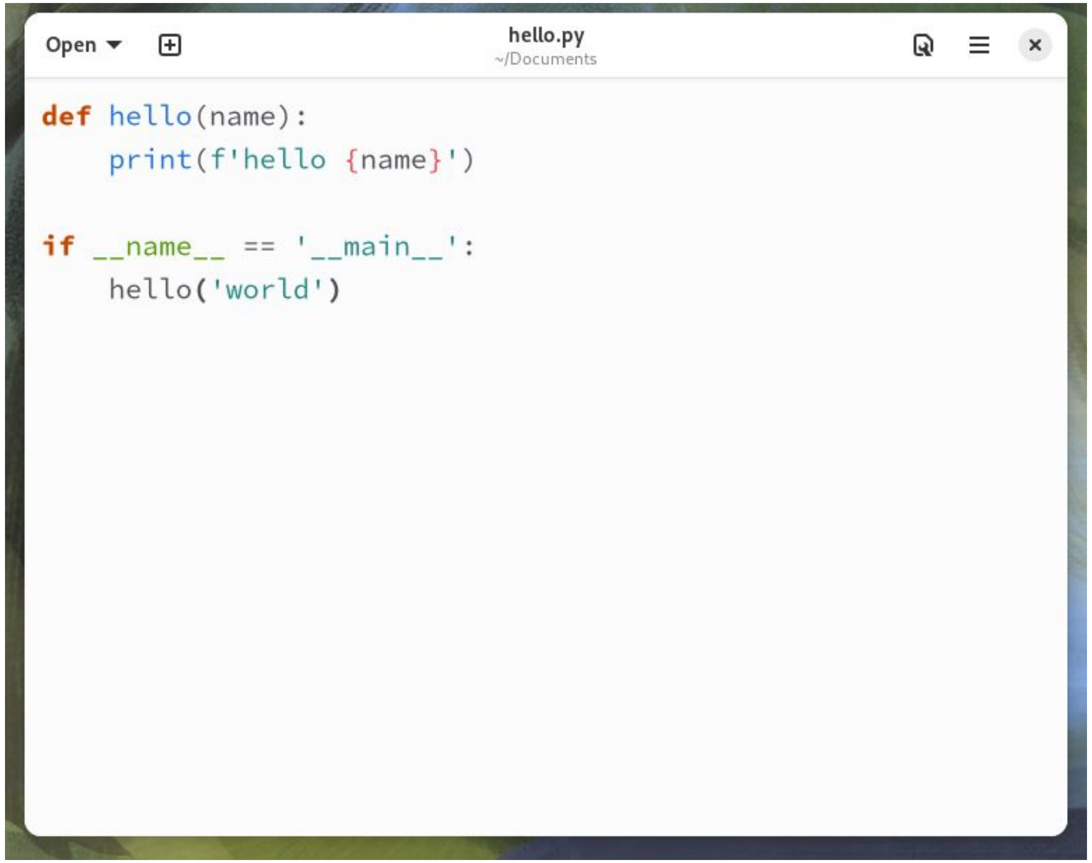
# Why care about text editors

# Why care about text editors

- Text editors are simple
- Being simple can make them faster
- All code is just text
- Code is not the only text we work with. Think log files.

# Default GUI Editor

# Default GUI Editor



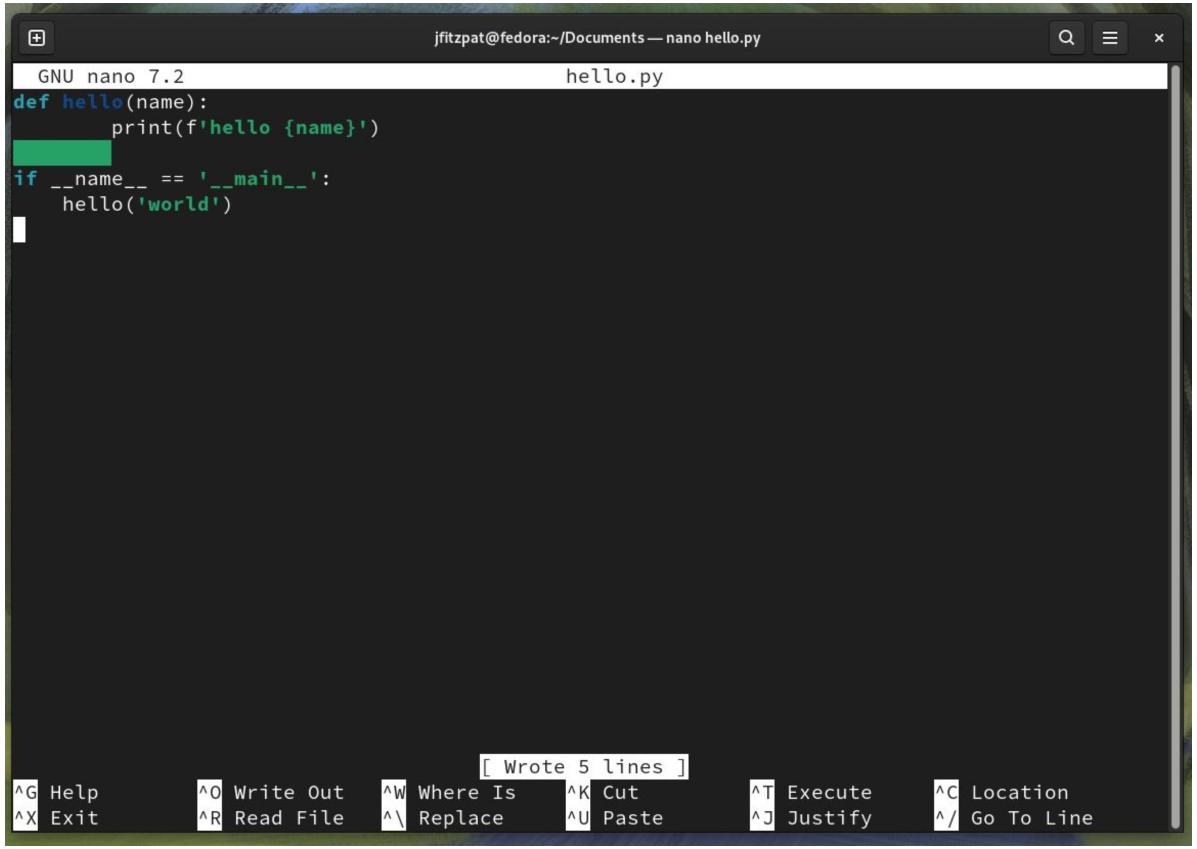
```
def hello(name):
    print(f'hello {name}')

if __name__ == '__main__':
    hello('world')
```

In Fedora the default GUI text editor is called **Text Editor**

- Supports syntax highlighting
- Supports a number of languages
- Allow opening multiple files in tabs
- Uses standard keyboard shortcuts

# Nano a terminal based editor

# Nano a terminal based editor



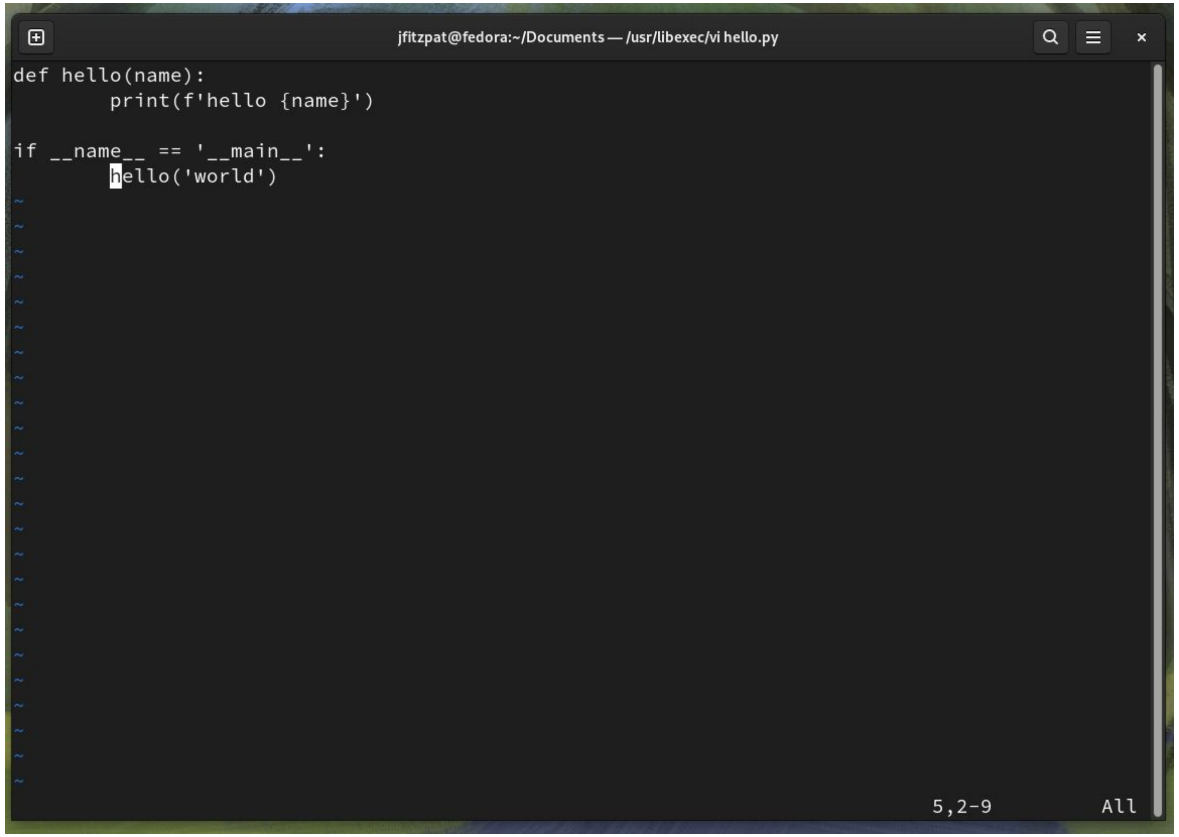
```
  GNU nano 7.2                              hello.py
def hello(name):
        print(f'hello {name}')

if __name__ == '__main__':
    hello('world')


                                      [ Wrote 5 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line
```

Nano is a lightweight terminal editor

- Supports syntax highlighting
- Supports a number of languages
- Launched from within a terminal
- List commands on bottom windowed
- Uses custom keyboard shortcuts

# Vi/Vim/NeoVim terminal based editors

# Vi/Vim/NeoVim terminal based editors



```
def hello(name):
        print(f'hello {name}')

if __name__ == '__main__':
        hello('world')
~
~
~
                                                              5,2-9         All
```

The vim family of  editors extremely powerful and customizable

- No syntax highlighting by default
- Supports a number of languages
- Launched from within a terminal
- Uses custom navigation called **Vim Motions**
- Lots of features, such as file explore
- Steep learning curve

# The influence of Vim and Vim Motions

Many applications have taken influence from Vim and Vim Motions. Lots command can be used across these applications.

- Search the current output with **/** and move to the next instances with **n**.
- Scroll the screen with **j** and **k**
- **gg** and **G** being used to move to the top and bottom of a file
- And there is many others

# Quitting Vim

## The source of many memes

Vim has three modes, **normal**, **insert** and **command**. You must be in command mode to quit.

- Press **ESC** to get from **insert** mode to **normal** mode.
- Press **:** to get from **normal** mode to **command** mode.
- Use **q + ENTER** to quit with no changes to a file
- Use **wq + ENTER** to save changes and quit
- Use **q! + ENTER** to quit and disregard any unsaved changes

# Recap

- Why use text editors
- Text Editor the default GUI editor
- Terminal based editors
- Nano
- Vim family

# Thank you

Red Hat is the world’s leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat