Software Engineering (CSC440/540)
Software Tools Installation

Dr. Samuel Cho, PhD

Spring, 2018

Software Tools 1

You should have these tools installed properly for your homework and project. Installation of tools is a part of HW1. Without proper installation, your project initiation will be delayed. So, make it your top priority for CSC440/540. All the project related files are on ‘Canvas/Files/project.’

- Pandoc - for document generation
- Python - for a team project
- PyCharm - IDE for Python
- Flask - Web Microservice

Software Tools 2

The installation of the tools will be a part of the team project, not HW assignments.
You are going to use web-based tools for your team project.

- Github - for project features implementation and code management.
- Readmydocs.org - for publishing Github documentation.

Other than that, you are free to use any tools for your project. Any tool is OK as long as your team has a consensus.

- UML design tools
- Scheduling tools (i.e., MS Project)
- Diagraming tools (i.e., MS Visio)
- Other tools for your project communication including Google Doc

Why Python?

- **Why? Why did I choose Python for the default programming language for a team project?**
- Because Python is very an interesting programming language!
- We will discuss Python as a part of SE theory as it's an excellent SE product.
    - Python has evolved over the past 25 years.
    - Python has adopted IP, OOP, and FP paradigms successfully.
    - Python (has tried) to mimic Lisp (the most advanced programming language) somewhat successfully.
    - Python has become one of the major (and the most important AI language) languages.
    - Python has a well-designed core/library structure.
    - Python's documentation is excellent.
    - Python's testing system is excellent.
    - Python is a dynamic language that uses the 'duck typing.' As you know static language such as Java, Python will be a good counterpart to see the differences.

Why not other languages?

- **Can my team choose other languages or other project topics?**
- Of course, you can choose any language and any software tools for your project.
- But you should let me know as early as possible. You should also provide me with language feature assignments (Stage D), software design analysis (Stage C), the possible features (Stage B) and advanced features (Stage A) to be fair with those students who would do the default project implementation (Knowledge Management System in Wiki using RESTFul API).

Installation

Let's install Pandoc/Python/PyCharm

Markdown and Pandoc

- **Markdown** is a file format for both human and computer. It is a text file that you can read and any app that understands the markdown file can raster the file for any format including HTML, pdf, docx (MS Word), and reST (Python's doc format). You can have pandoc demos from https://pandoc.org/demos.html.
  - We use markdown for a homework assignment.
  - We also use markdown for the Knowledge Management System uses markdown.
- **Pandoc** is a command line tool that converts one file format to the other.

Demo I'll show a demo of 'markdown,' and I'll show you how pandoc transforms the markdown into various file formats.

Pandoc installation

Installation of Pandoc is also a part of HW1.

- You need to install LaTeX to generate a pdf file from Pandoc.
- I highly recommend you to install LaTeX, but if you can't install it, please let me know. I'll help you out. You will have a point deduction from a format violation if your submission is from Pandoc/LaTeX.
- Why? **Why I ask you to use Pandoc/LaTeX?**:
  - I need one format (pdf generated from Pandoc/LaTeX) to evaluate your output.
  - It is likely that you need the pdf generation feature for your team project.
  - You can use the combination for your documentation work.

Pandoc installation

- **LaTeX:** Visit TeXLive (https://www.tug.org/texlive/). LaTeX system is small, but the supporting packages are large (total 4G), so it might take a while to install all the LaTeX packages.
  - For Mac Users, download MacTeX(https://www.tug.org/mactex/).
  - For Windows users, Pandoc recommends MikTeX(https://miktex.org), but it's OK to use TeXLive http://www.tug.org/texlive/acquire-netinstall.html.
- **Pandoc:** Visit Pandoc (https://Pandoc.org/installing.html) for installation.

After installation, don't forget to add the path to LaTeX and Pandoc to your PATH environment.
Download 'markdown.zip' from 'Canvas/Files/project' to check if you can generate the HTML/pdf/docx files from the input source file.

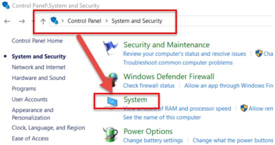
Python installation - Python

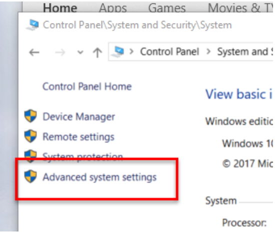
⚠Use Python 2.7.X, not Python 3.X. The team project wiki system is built on Python 2.X. For Linux, please google to find the installation process.

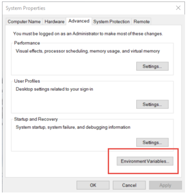
- **Mac OS X:** Python is preinstalled already for Mac OS. Check if your Python version is 2.7.X with 'python –version'. If not, you can install python with Homebrew. Install Homebrew (`https://brew.sh`) and run 'brew install python'.
- **Windows:** Download Python from `https://www.python.org/downloads/`. Make sure you download newest version of 2.7.X. I assume you install Python to the default directory (c:/Python27).

PATH environment variable set on Windows 1

- Open ‘Control Panel’.
- Select ‘System and Security’.
- Select ‘System’.
- Select ‘Advanced system settings’.
- Click ‘Environment Variables’.


Control Panel\System and Security
Control Panel > System and Security
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock, Language, and Region
Ease of Access
Security and Maintenance
Review your computer's status and resolve issues
Cha
Troubleshoot common computer problems
Windows Defender Firewall
Check firewall status
Allow an app through Windows Fir
System
View amount of RAM and processor speed
Allow ren
See the name of this computer
Power Options
Change battery settings
Change what the power buttons





You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

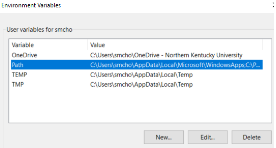
Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

PATH environment variable set on Windows 2

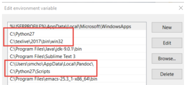
- Click on 'Path' or click 'Edit...' menu.
- Add paths.
- Check all the paths are added.
- Check if your command line tool (cmd.exe) can find the tools installed and configured.



Environment Variables

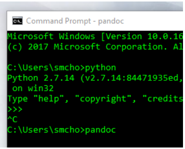
User variables for smcho

Variable	Value
OneDrive	C:\Users\smcho\OneDrive - Northern Kentucky University
Path	C:\Users\smcho\AppData\Local\Microsoft\WindowsApps;C:\P...
TEMP	C:\Users\smcho\AppData\Local\Temp
TMP	C:\Users\smcho\AppData\Local\Temp



Edit environment variable

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Python27
C:\texlive\2017\bin\win32
C:\Program Files\Java\jdk-9.0.1\bin
C:\Program Files\Sublime Text 3
C:\Users\smcho\AppData\Local\Pandoc\
C:\Python27\Scripts
C:\Program Files\emacs-25.3_1-x86_64\bin
```



Command Prompt - pandoc

```
Microsoft Windows [Version 10.0.16
(c) 2017 Microsoft Corporation. Al

C:\Users\smcho>python
Python 2.7.14 (v2.7.14:84471935ed,
 on win32
Type "help", "copyright", "credits
>>>
^C
C:\Users\smcho>pandoc
```

Python installation - Virtualenv 1

- **Mac OS X:** Run ‘sudo easy_install pip’. Maybe this is your first and last time to use easy_install. From now on, always use ‘pip’ not ‘easy_install’ for package installation. Run ‘pip install virtualenv’ to install virtualenv package.
- **Windows:** PIP and virutalenv and pip is pre-installed with Python 2.7.X. Check ‘`c:\Python27\Script`’ has both ‘pip.exe’ and ‘mkvirtualenv.bat’. Make sure that the ‘`c:\Python27\Script`’ is in your PATH environment variable.

Python installation - Virtualenv 2

You create a new python virtual environment.

- **Mac OS X:** Test run virtualenv by executing 'virtualenv D/V' (D is any directory that you want to install your virtual python environment and V is the name of your virtualenv). For example, 'virtualenv `~/virtualenv/riki`' creates a virtualenv named riki in your 'home(~)/virtualenv' directory.
- **Windows:** Use the 'virtualenv' in the '`c:\Python27\Script`' directory. For example, 'virtualenv `c:\virtualenv\riki`' to make riki virtualenv in the directory named `c:\virtualenv`.

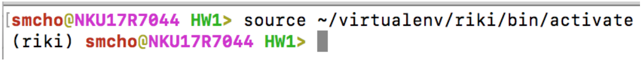
Then, activate the virtual env.

- **Mac OS X:** Execute 'source D/V/bin/activate'. For example, '`source ~/virtualenv/riki/bin/activate`'.
- **Windows:** Execute '`D\V\Scripts\activate`'. For example '`c:\virtualenv\riki\Scripts\activate`'.

Python installation - Virtualenv 3

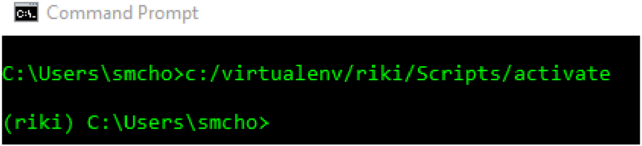
Now you see the name of your virtualenv prepended to show you that you are using Python in the virtualenv you just set with parenthesis.

- **Mac OS X:**



```
[smcho@NKU17R7044 HW1> source ~/virtualenv/riki/bin/activate
(riki) smcho@NKU17R7044 HW1> 
```

- **Windows:**

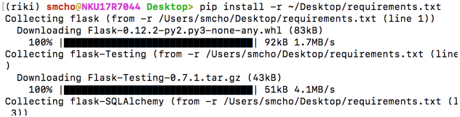


```
C:\Users\smcho>c:/virtualenv/riki/Scripts/activate

(riki) C:\Users\smcho>
```

- You can execute ‘deactivate’ to be out of the virtual environment.
- PyCharm understands virtualenv, and you should teach PyCharm what virtualenv you will use.
- We will discuss over the importance of ‘virtual environment’ from software engineering perspective.

Python: PIP and requirements.txt

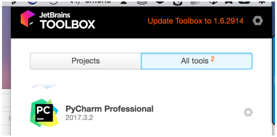
- You can install individual package using PIP. For example, when you install flask package, you use ‘pip install flask’.
- When you install multiple packages, you can use ‘requirements.txt’ that contains package names to install.
- Download the requirements file from Canvas/project. I assume you download the ‘requirement.txt’ at `~/Desktop`.
- Run `pip install -r ~/Desktop/requirements.txt`.



```
[(riki) smcho@NKU17R7044 Desktop> pip install -r ~/Desktop/requirements.txt
 Collecting flask (from -r /Users/smcho/Desktop/requirements.txt (line 1))
   Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
     100% |████████████████████████████████| 92kB 1.7MB/s
 Collecting flask-Testing (from -r /Users/smcho/Desktop/requirements.txt (line
 )
   Downloading Flask-Testing-0.7.1.tar.gz (43kB)
     100% |████████████████████████████████| 51kB 4.1MB/s
 Collecting flask-SQLAlchemy (from -r /Users/smcho/Desktop/requirements.txt (l
  3))
```

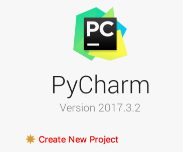
PyCharm installation

- You should use the PyCharm professional version because only the professional version has the Flask debugging feature.
- If you don't have the PyCharm professional in your computer, you need to contact JetBrains to request a student license. https://www.jetbrains.com/shop/eform/students
- Please request the license because it may take some time.
- In a day or so, you'll receive a confirmation email.
- Install 'All Products Pack' and select 'PyCharm Professional'.



PyCharm: Start a project

- You need to setup your PyCharm project. I assume it '`~/PyCharmProjects`'.
- I will use the project name 'Riki' and the Flask app name 'Riki.py', you may need to change the project name and app.
- Start a new project.

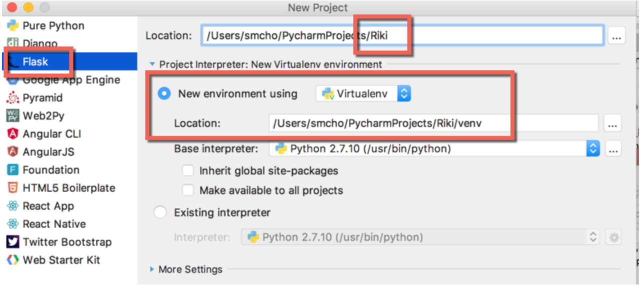


PyCharm

Version 2017.3.2

PyCharm: Select Flask and virtualenv

- You use Flask.
- You set project name.
- You choose to use virtualenv. Remember the default virtualenv name is (venv).



Location: /Users/smcho/PycharmProjects/Riki

Project Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: /Users/smcho/PycharmProjects/Riki/venv

Base interpreter: Python 2.7.10 (/usr/bin/python)

□ Inherit global site-packages

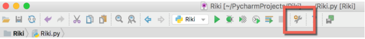
□ Make available to all projects

○ Existing interpreter

Interpreter: Python 2.7.10 (/usr/bin/python)

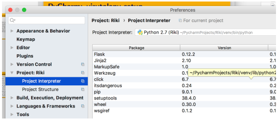
PyCharm: Check preference

- Click the ‘preferences’ button.



Riki [~/PycharmProjects/Riki] /Riki.py [Riki]

- Check correct virtualenv is setup .



Preferences

Project: Riki › Project Interpreter For current project

Project Interpreter: Python 2.7 (Riki) ~/PycharmProjects/Riki/venv/bin/python

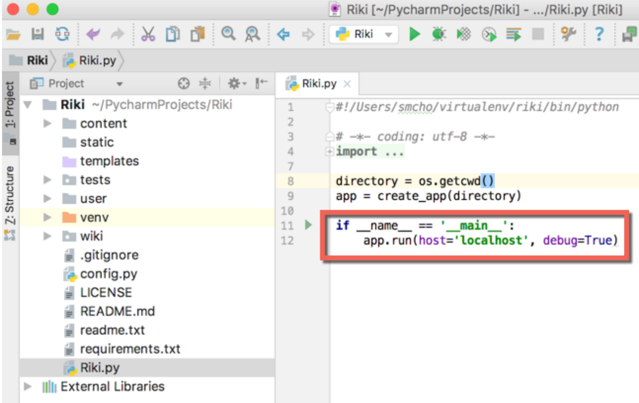
Package	Version	
Flask	0.12.2	0.1
Jinja2	2.10	2.1
MarkupSafe	1.0	1.0
Werkzeug	0.1	
click	6.7	6.7
itsdangerous	0.24	0.2
pip	9.0.1	9.0
setuptools	38.4.0	38.
wheel	0.30.0	0.3
wsgiref	0.1.2	0.1

~/PycharmProjects/Riki/venv/lib/python2

- You need more packages. In command line, activate the PyCharm virtualenv you just created using ‘`source ~/PycharmProjects/Riki/venv/bin/activate`’. You should see the (venv) prepended at your prompt.
- Run ‘pip install -r D/requirements.txt’. (D is the directory where the requirements.txt is located).
- Check again with ‘preferences’ that all the necessary packages are installed to your virtualenv.

PyCharm: Copy the wiki flask source

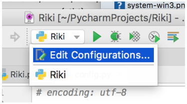
- Download the wiki_flask.zip from Canvas/Files/project.
- Unzip the wiki_flask.zip and copy the contents into the '`~/PycharmProjects/Riki`'.



```python
#!/Users/smcho/virtualenv/riki/bin/python

# -*- coding: utf-8 -*-
import ...

directory = os.getcwd()
app = create_app(directory)

if __name__ == '__main__':
    app.run(host='localhost', debug=True)
```

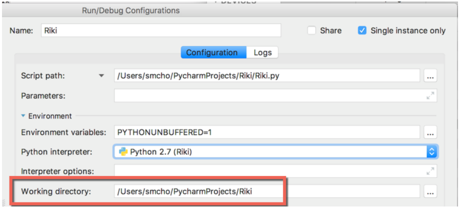
PyCharm: Wiki configuration

- Open ‘config.py’. Check CONTENT_DIR points to the ‘content’ directory. It is in your ‘PycharmProject/Riki’. Also update the USER_DIR directory to point to your user directory.
- ⚠For windows users, don’t use single \, always use ‘/’ or double \\ because Python interprets ‘\’ as a special word.
- Click the Edit Configuration.



```
# encoding: utf-8
```

- Set the default directory.



Run/Debug Configurations

Name: Riki ☐ Share ☑ Single instance only

Configuration | Logs

Script path: /Users/smcho/PycharmProjects/Riki/Riki.py

Parameters:

Environment

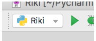
Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Python 2.7 (Riki)

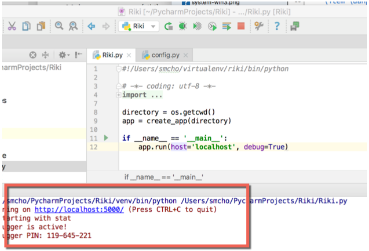
Interpreter options:

Working directory: /Users/smcho/PycharmProjects/Riki

PyCharm: Run the app

- Click the right arrow .

- Check you see ‘Running on htt://localhost:5000/’



system-win3.png
Riki [~/PycharmProjects/Riki] - .../Riki.py [Riki]
Riki
Riki.py
config.py
armProjects/Riki
if __name__ == '__main__'

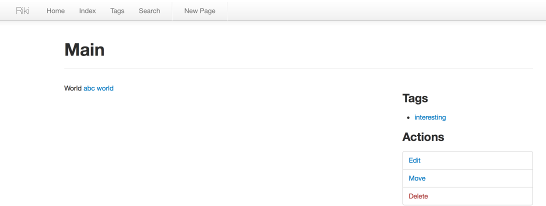
```
#!/Users/smcho/virtualenv/riki/bin/python

# -*- coding: utf-8 -*-
import ...

directory = os.getcwd()
app = create_app(directory)

if __name__ == '__main__':
    app.run(host='localhost', debug=True)
```

```
/smcho/PycharmProjects/Riki/venv/bin/python /Users/smcho/PycharmProjects/Riki/Riki.py
ring on http://localhost:5000/ (Press CTRL+C to quit)
tarting with stat
ugger is active!
ugger PIN: 119-645-221
```

.

- You are requests to login use ‘name/1234’. You can add other name/password by updating ‘user/users.json’ file.

- Now, you have your own wiki system on your computer. Create any file and check the created file is in your content directory.
- Browse the source code with PyCharm. Check how the wiki is operating with RESTFul APIs.



Main

World abc world

Tags

- interesting

Actions