Terminal

python -v

python -V

python3 -V

# Setting PATH for Python 3.6

# The original version is saved in .bash\_profile.pysave

PATH="/Library/Frameworks/Python.framework/Versions/3.6/bin:${PATH}"

export PATH

# alias for sublime to open

alias subl="/Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl"

# get current branch in git repo

function parse\_git\_branch() {

BRANCH=`git branch 2> /dev/null | sed -e '/^[^\*]/d' -e 's/\* \(.\*\)/\1/'`

if [ ! "${BRANCH}" == "" ]

then

STAT=`parse\_git\_dirty`

echo "[${BRANCH}${STAT}]"

else

echo ""

fi

}

# get current status of git repo

function parse\_git\_dirty {

status=`git status 2>&1 | tee`

dirty=`echo -n "${status}" 2> /dev/null | grep "modified:" &> /dev/null; echo "$?"`

untracked=`echo -n "${status}" 2> /dev/null | grep "Untracked files" &> /dev/null; echo "$?"`

ahead=`echo -n "${status}" 2> /dev/null | grep "Your branch is ahead of" &> /dev/null; echo "$?"`

newfile=`echo -n "${status}" 2> /dev/null | grep "new file:" &> /dev/null; echo "$?"`

renamed=`echo -n "${status}" 2> /dev/null | grep "renamed:" &> /dev/null; echo "$?"`

deleted=`echo -n "${status}" 2> /dev/null | grep "deleted:" &> /dev/null; echo "$?"`

bits=''

if [ "${renamed}" == "0" ]; then

bits=">${bits}"

fi

if [ "${ahead}" == "0" ]; then

bits="\*${bits}"

fi

if [ "${newfile}" == "0" ]; then

bits="+${bits}"

fi

if [ "${untracked}" == "0" ]; then

bits="?${bits}"

fi

if [ "${deleted}" == "0" ]; then

bits="x${bits}"

fi

if [ "${dirty}" == "0" ]; then

bits="!${bits}"

fi

if [ ! "${bits}" == "" ]; then

echo " ${bits}"

else

echo ""

fi

}

# terminal look; regx statement

export PS1="\[\e[36m\]\A\[\e[m\] \[\e[32m\]\W\[\e[m\] \[\e[37;40m\]\`parse\_git\_branch\`\[\e[m\] "

mkrdir

ls -a

open /Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl

nano .bash\_profile

# alias for sublime to open alias subl="/Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl”

ctrl x

subl

subl .bash\_profile

Get git for terminal

# get current branch in git repo

# function parse\_git\_branch() {

# BRANCH=`git branch 2> /dev/null | sed -e '/^[^\*]/d' -e 's/\* \(.\*\)/\1/'`

# if [ ! "${BRANCH}" == "" ]

# then

# STAT=`parse\_git\_dirty`

# echo "[${BRANCH}${STAT}]"

# else

# echo ""

# fi

# }

# get current status of git repo

# function parse\_git\_dirty {

# status=`git status 2>&1 | tee`

# dirty=`echo -n "${status}" 2> /dev/null | grep "modified:" &> /dev/null; echo "$?"`

# untracked=`echo -n "${status}" 2> /dev/null | grep "Untracked files" &> /dev/null; echo "$?"`

# ahead=`echo -n "${status}" 2> /dev/null | grep "Your branch is ahead of" &> /dev/null; echo "$?"`

# newfile=`echo -n "${status}" 2> /dev/null | grep "new file:" &> /dev/null; echo "$?"`

# renamed=`echo -n "${status}" 2> /dev/null | grep "renamed:" &> /dev/null; echo "$?"`

# deleted=`echo -n "${status}" 2> /dev/null | grep "deleted:" &> /dev/null; echo "$?"`

# bits=''

# if [ "${renamed}" == "0" ]; then

# bits=">${bits}"

# fi

# if [ "${ahead}" == "0" ]; then

# bits="\*${bits}"

# fi

# if [ "${newfile}" == "0" ]; then

# bits="+${bits}"

# fi

# if [ "${untracked}" == "0" ]; then

# bits="?${bits}"

# fi

# if [ "${deleted}" == "0" ]; then

# bits="x${bits}"

# fi

# if [ "${dirty}" == "0" ]; then

# bits="!${bits}"

# fi

# if [ ! "${bits}" == "" ]; then

# echo " ${bits}"

# else

# echo ""

# fi

# }

Create virtual environment

separate environment python2 and python3

mkdir flash\_tutorial

cd flash

Last login: Sun Feb 18 13:23:16 on ttys000

13:24 ~  cd cjn-python/

13:24 cjn-python  mkdir flask\_tutorial

13:24 cjn-python  cd flask\_tutorial/

13:25 flask\_tutorial  python3 -m venv ENV

13:26 flask\_tutorial  ls

ENV

13:26 flask\_tutorial  cd env

13:26 env  ls

bin include lib pyvenv.cfg

13:26 env  pwd

/Users/clarkngo/cjn-python/flask\_tutorial/env

13:27 env  cd..

-bash: cd..: command not found

13:27 env  cd ..

13:27 flask\_tutorial  cd ENV/

13:27 ENV  cd bin/

13:27 bin  ls

activate easy\_install-3.6 python

activate.csh pip python3

activate.fish pip3

easy\_install pip3.6

13:28 bin  ls -a

. activate.fish pip3

.. easy\_install pip3.6

activate easy\_install-3.6 python

activate.csh pip python3

13:28 bin  source activate

(ENV) 13:28 bin  which python

/Users/clarkngo/cjn-python/flask\_tutorial/ENV/bin/python

(ENV) 13:28 bin  python -V

Python 3.6.2

(ENV) 13:29 flask\_tutorial  deactivate

13:30 flask\_tutorial  which python

/usr/bin/python

13:30 flask\_tutorial

13:30 flask\_tutorial  python -V

Python 2.7.10

13:30 flask\_tutorial

13:30 flask\_tutorial  python -V

Python 2.7.10

13:30 flask\_tutorial  source ENV/bin/activate

(ENV) 13:31 flask\_tutorial

pip python manager

<https://pypi.python.org/pypi/Flask>

pip freeze

pip install flask

(ENV) 13:35 flask\_tutorial  pip freeze

click==6.7

Flask==0.12.2

itsdangerous==0.24

Jinja2==2.10

MarkupSafe==1.0

Werkzeug==0.14.1

(ENV) 13:35 flask\_tutorial

(ENV) 13:36 flask\_tutorial  mkdir app && cd app

(ENV) 13:36 app

double underscores are important

(ENV) 13:36 app  touch \_\_init\_\_.py

(ENV) 13:37 app  ls

\_\_init\_\_.py

(ENV) 13:37 app  cd ..

(ENV) 13:37 flask\_tutorial  subl .

(ENV) 13:37 flask\_tutorial

opens up

Folders in flask\_tutorial in sublime\_text

command shift P

install

install control package

sublimeLinter

<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>

<https://www.youtube.com/playlist?list=PLVngfM2hsbi-Uo2jDso-K06VScDC1ucMy>

<https://codefellows.github.io/sea-python-401d7/index.html?lipi=urn%3Ali%3Apage%3Ad_flagship3_feed%3B1oYc77NCTHyv%2FTFmuJmw%2Bw%3D%3D>

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/>

ipython

SHA512 and SHA256

<https://globallegalhackathon.com/>

<https://angelhack.com/>

jinja2 templates

render\_template

‘’’Reloads every save’''

flask run --reload

<html>

<head>

<title>{{ title }} - Microblog</title>

</head>

<body>

<h1>Hello, {{ user.username }}!</h1>

</body>

</html>

<html>

    <head>

        {% if title %}

        <title>{{ title }} - Microblog</title>

        {% else %}

        <title>Welcome to Microblog!</title>

        {% endif %}

    </head>

    <body>

        <h1>Hello, {{ user.username }}!</h1>

    </body>

</html>

\*args \*\*kwargs

render template

looks for templates folder

does not need to have equal parameters to be passed.

uses \*\* context

Bootstrap

boostrap cdn