Documentation for Python

Web Serve on Python: Django: <https://www.djangoproject.com/start/overview/>

Flask: <https://palletsprojects.com/p/flask/>

en: <https://docs.python.org/3/tutorial/>

cn: <https://www.liaoxuefeng.com/wiki/1016959663602400> (recommended)

## **Jupyter Notebook**

To check running jupyter server: jupyter notebook list

|  |  |
| --- | --- |
| Server run: | nohup jupyter notebook --no-browser --port=8890 & |
| Local Run: | ssh -N -f -L localhost:8887:localhost:8890 lxiang@shang.phys.gwu.edu |
| Local Browser: | localhost:8887 |
| Find process | ps aux | grep "ssh -N -f -L localhost:8887:localhost:8890 |
| -N | Do not execute a remote command, this is useful for just forwarding ports. |
| -f | Requests **ssh** to go to background just before command execution. This is useful if **ssh** is going to ask for passwords or passphrases, but the user wants it in the background. This implies **-n**. The recommended way to start X11 programs at a remote site is with something like **ssh -f host xterm**. |
| Man ssh | https://man.openbsd.org/ssh |
|  |  |

## **Rstudio**

R package conflict with miniconda

ln -s /usr/local/bin/R ./

<https://support.rstudio.com/hc/en-us/articles/200532327-Managing-the-Server>

ssh -N -f -L localhost:7777:localhost:8787 lxiang@shang.phys.gwu.edu

## **Jupyter Switch Kernel**

1. python -m pip install ipykernel
2. python -m ipykernel install --user --name ipykernel\_py3 --display-name "Python ipykernel\_py3" (This gives ipyker)
3. jupyter kernelspec list
4. python -c "import matplotlib as pd; print (pd.\_\_version\_\_)"h

|  |  |
| --- | --- |
| conda create -n py36 python=3.6  conda activate py36  conda install notebook ipykernel  ipython kernel install --user | jupyter kernelspec list (find available kernels)  locate “kernel.json” and replace  “argv”:[  “python\_wrong\_version”, with “correct version”  ] |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **Package Lost**

export PYTHONPATH=$PYTHONPATH:/opt/miniconda2/lib/python2.7/site-packages

Alternatively, you can add the custom module directory in PYTHONPATH environment variable, which will augment the default module search paths used by the Python interpreter

|  |  |
| --- | --- |
| import sys  sys.path | sys.path.append('/home/Xiang/miniconda2/lib/python2.7/site-packages')  ###  '/home/Xiang/miniconda2/lib/python27.zip',  '/home/Xiang/miniconda2/lib/python2.7',  '/home/Xiang/miniconda2/lib/python2.7/plat-linux2',  '/home/Xiang/miniconda2/lib/python2.7/lib-tk',  '/home/Xiang/miniconda2/lib/python2.7/lib-old',  '/home/Xiang/miniconda2/lib/python2.7/lib-dynload' |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **Fonts:**

<https://askubuntu.com/questions/651441/how-to-install-arial-font-and-other-windows-fonts-in-ubuntu>

## **Python Identifiers**

https://www.tutorialspoint.com/python/python\_basic\_syntax.htm

|  |  |
| --- | --- |
| Class = Xxxx | \_xxxx indicates that the identifier is private. |
| others = xxxx | \_\_xxx indicates that the identifier is strongly private. |
|  | \_\_xx\_\_ indicates: A languagedefined special name |
|  |  |

## **Variable Types**

https://www.tutorialspoint.com/python/python\_variable\_types.htm

|  |  |
| --- | --- |
| Numbers | Int, float, complex, long  10, 0.0, 3.14j, 51923L, (Use Uppercase L in tail) |
| String var=’0123456789’  var[1:-1] = 12345678  %c, %s, %i, %d, %u, %f | Print String Management:  <https://www.tutorialspoint.com/python/python_strings.htm>  print "My name is %s and weight is %d kg!" % ('Zara', 21) |
| List = ['a','b','c'] | List[1:2] = ['b'] |
| Tuple=('a','b','c') | Tuples can be thought of as **read-only** lists |
| Dictionary |  |
|  |  |
|  |  |
|  |  |

## **Python Operators**

https://www.tutorialspoint.com/python/python\_basic\_operators.htm

|  |  |
| --- | --- |
| Membership operators | In, not in, is, is not |
| Bitwise operators |  |
|  |  |

## **Decision Making**

|  |  |
| --- | --- |
| If ( a == b):  Statements  else:  statements | if expression1:  statement(s)  elif expression2:  statement(s)  else:  statement(s) |
| While expression:  Statement  Or  While expression: Statement | While expression:  Statement  Else:  statement |
| for name in name\_list:  statement | for index in range(len(fruits)):  else: |
| Break, continue, pass |  |
|  |  |

## **Mathematical Functions**

Import math

|  |  |
| --- | --- |
| Ceil(x)  Floor(x) | The ceiling of x, smallest integer not less than x.  The floor of x: the largest integer not greater than x |
| Cmp(x,y) | -1 if x < y, 0 if x == y, or 1 if x > y |
| Modf(100.12) | (0.12000000000000455, 100.0) |
| Pow(x,y) | The value of x\*\*y |
| Round(x,[,n]) | x rounded to n digits from the decimal point. |
|  |  |

Import random

|  |  |
| --- | --- |
| Choice() | A random item from a list, tuple, or string |
| Randrange(1,100,10) | A randomly selected element from range(start, stop, step) |
| Random() | Between 0 to 1 |
| Shuffle(list) | randomizes the items of a list in place. |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  | word = 'word'  sentence = "This is a sentence."  paragraph = """This is a paragraph. It is  made up of multiple lines and sentences.""" |
|  | raw\_input("\n\nPress the enter key to exit.") |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **Environment Variables**

https://www.tutorialspoint.com/python/python\_environment.htm

|  |  |
| --- | --- |
| Setting path at unix/ linux | **bash shell (Linux)** − type export ATH="$PATH:/usr/local/bin/python" and press Enter. |
| line continuation character (\) | single ('), double (") (''' or """) quotes |
| Python -h | provide you with usage info |
|  |  |

## **https://www.tutorialspoint.com/python/python\_variable\_types.htm**

## **Sample**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Calculation

### Numbers:

#### In interactive mode, the last printed expression is assigned to the variable \_. This means that when you are using Python as a desk calculator, it is somewhat easier to continue calculations.

### Complex numbers:

#### uses the j or J suffix to indicate the imaginary part (e.g. 3+5j)

### Lists:

#### Cubes=[1, 2, 3, 4, 5]

#### Cubes.append (6)

#### Nest lists x[0][1]

## Logic operation

### If:

#### If x < 0:

##### … print

#### Elif x == 0:

#### Else:

### For:

#### For w in words:

##### Print w, len(w)

## 

## **Color palette**

## <https://seaborn.pydata.org/generated/seaborn.diverging_palette.html#seaborn.diverging_palette>

## **Pandas**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## **Packaging and distribution**

<https://setuptools.readthedocs.io/en/latest/userguide/keywords.html>

<https://packaging.python.org/tutorials/packaging-projects/>

## [**https://python-packaging.readthedocs.io/en/latest/command-line-scripts.html**](https://python-packaging.readthedocs.io/en/latest/command-line-scripts.html)

|  |  |
| --- | --- |
| python3 -m build | python3 -m twine upload --repository pypi ./dist/\* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Conda manage environment**

<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file>

conda env export > environment.yml

conda env create -f environment.yml

conda env update --file environment\_hichub.yml –prune

conda remove --name myenv --all

## **Sample**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |