

Module 61-12: Option GIS-Python

Introduction to Python

hes.
so
business.

Jean-Paul Calbimonte

School of Management

Bachelor of Science HES-SO (BSc) in Business
Information Technology



> GIS for developers

Why Python for geodata?



- **Free:** no added costs for licensing
- **For coders:** fully programmable geodata manipulation
- **Modular:** libraries adapted to different use-cases
- **Efficiency:** optimized for Big Data analytics
- **Extensibility:** possibility to extend or reuse multiple libraries
- **Flexibility:** options for lots of formats/standards/approaches
- **Open Source:** code reuse/reproducibility/open science
- **Integration:** supported by other tools as QGIS/ArcGIS etc.

> How do we run python?

Different options

- Online environment: Jupyter Python in renkulab.io

In this course:



- Jupyter Python in Local installation: (e.g. Anaconda)

> renkulab.io

Log in with your hevs edu-ID account

The image shows a composite of two screenshots. The left screenshot is the RenkuLab homepage, featuring a dark blue background with a pattern of colorful dots. The RenkuLab logo is in the top left. The main text reads "RenkuLab" in large white font, followed by "We have been improving RenkuLab. Welcome to the new version!" in green and pink. At the bottom, there are three buttons: "Feedback", "Learn more", and "Login". The right screenshot is the SWITCH edu-ID login page. It has a white background and a grey header with "SWITCH edu-ID" and "Help EN". The main content area is titled "Log in to: renkulab.io" and contains a "Service description" section. Below this is a "SWITCH edu-ID" section with an "E-mail:" field containing the letter 'j' and a "Password:" field with a placeholder "Enter your password" and an eye icon. There are "Create account" and "Login" buttons. At the bottom of the login page, there are links for "Forgot password?" and "Options for personal data protection". The SWITCH logo is in the bottom right corner of the login page.

renku

RenkuLab

We have been improving RenkuLab.
Welcome to the new version!

Feedback Learn more Login

SWITCH edu-ID

Help EN

Log in to: renkulab.io

Service description:
This is a public beta instance of the Renku platform for reproducible data science operated by the Swiss Data Science Center.

SWITCH edu-ID

E-mail: j

Password: Enter your password

Create account Login

Forgot password?
Options for personal data protection

SWITCH

About / Terms of Use / Legal Notice / Imprint

> renkulab.io

Got to this project and fork it:

<https://renkulab.io/projects/jean-paul.calbimonte/gis-python/>

The screenshot shows the project page for 'GIS Python' on renkulab.io. The page header includes the 'renku' logo and a menu icon. The main content area displays the project title 'GIS Python' with a 'Start' button. Below the title, it indicates the project is 'Public' and owned by 'Jean-Paul Calbimonte Pérez', who forked it from 'jean.paul.ik/gis-python'. The project description is 'GIS Python materials for the Bachelor in Business IT HES-SO Valais-Wallis'. There are icons for a globe and Python, and a note that the project was 'Updated 3 months ago'. A navigation bar includes 'Overview', 'Collaboration', 'Files', 'Datasets', 'Workflows', 'Sessions', and 'Settings'. The 'General' section shows the project was updated 3 months ago, with a 'Fork' button circled in red, 19 forks, 1 star, and a 'View in GitLab' button. The 'Stats' section shows 'README.md' and the 'Status' section shows 'GIS Python'.

fork

> renkulab.io

https://renkulab.io/projects/jean-paul.calbimonte/gis-python/

GIS Python ▶ Start ▼

Project Public

Jean-Paul Calbimonte Pérez
jean-paul.calbimonte/gis-python forked from [jean.paul.ik/gis-python](#)

GIS Python materials for the Bachelor in Business IT HES-SO Valais-Wallis

Updated 3 months ago.

Overview Collaboration Files Datasets Workflows **Sessions** Settings

Go to Sessions

Back to jean-paul.calbimonte/gis-python

Sessions

No currently running sessions.

+ New session

Start a new session and open
it when it is ready

Default Environment
/lab /rstudio

Number of CPUs
0.25 0.5 1 2

Amount of Memory
1G 2G 4G 8G

Amount of Storage
1G 4G 16G 64G

Number of GPUs 0

Automatically fetch LFS data

Environment Variables (Optional)
Add Variable

Start session ▼

> renkulab.io

The screenshot shows the renkulab.io web interface. On the left is a file explorer with a table of files:

Name	Last Modified
01-datatype...	4 minutes ago
02-lists.ipynb	4 minutes ago
03-loops.ipy...	4 minutes ago
04-conditio...	4 minutes ago
05-function...	4 minutes ago
06-classes.i...	4 minutes ago
07-assertion...	4 minutes ago
08-files.ipynb	4 minutes ago
books.py	4 minutes ago
cantons.txt	4 minutes ago
exercise.ipynb	4 minutes ago
mats.py	4 minutes ago
test.txt	4 minutes ago

The main area is titled 'notebooks/01-PythonIntro' and contains a 'Launcher' section with the following options:

- Notebook (with a Python 3 icon)
- Console (with a terminal icon)
- Python 3 (with a Python 3 icon)
- Other (with a terminal icon)

Under the 'Other' section, there are four buttons: Terminal, Text File, Markdown File, and Show Contextual Help.

Ready to open the notebooks

> Running Jupyter

Basics of Python

> Basic operators

$1 + 1$

2

$3 * 4.5$

13.5

$2 - 3.5$

-1.5

$10 / 3$

3.3333333333333335

$3 ** 2$

9

basic arithmetic operations

> Math functions

```
import math
```

```
math.sqrt(81)
```

```
9.0
```

```
math.pi
```

```
3.141592653589793
```

```
math.sin(math.pi/2)
```

```
1.0
```

import library

imported function

> Variables & Basic types

```
a=1
```

```
b=2.4      int
```

```
a+b      float
```

```
3.4
```

```
a==b
```

```
bool
```

```
False
```

```
a==1
```

```
True
```

```
st='we like GIS'  str
```

```
print(st)
```

```
we like GIS
```

```
b=st
```

```
print(b)
```

```
we like GIS
```

```
print('This is number',a,'followed by this string:',b)
```

```
This is number 1 followed by this string: we like GIS
```

> Basic types

```
type (a)
```

```
int
```

```
type (st)
```

```
str
```

```
type (3.42)
```

```
float
```

```
type (a==2)
```

```
bool
```

int
str
float
bool



> Lists

```
[3, 4, 6, 2, 1]
```

unidimensional

```
[3, 4, 6, 2, 1]
```

```
[[2, 3], [5, 6], [4, 3]]
```

multidimensional

```
[[2, 3], [5, 6], [4, 3]]
```

```
list=[3, 4.5, 6, 2, 2.1]
```

```
len(list)
```

```
5
```

```
list[2]
```

```
6
```

```
list[-1]
```

```
2.1
```

indices

```
type(list[1])
```

```
float
```

types

> Lists

```
list=[3,4.5,6]  
del list[2]  
print(list)  
[3, 4.5]
```

remove item

```
list.append(44)  
print(list)  
[3, 4.5, 44]
```

append item

```
list.reverse()  
print(list)  
[44, 4.5, 3]
```

reverse list

```
list.sort()  
print(list)  
[3, 4.5, 44]
```

sort list

```
list[0]=55  
print(list)  
[55, 4.5, 44]
```

replace item

> Lists & Loops

```
string='strange'  
len(string)  
7  
  
print(string[2])  
r  
  
for ch in string:  
    print(ch)  
  
s  
t  
r  
a  
n  
g  
e
```

iterate over chars

```
count=0  
for ch in string:  
    print(ch, count)  
    count=count+1  
print(count)  
s 0  
t 1  
r 2  
a 3  
n 4  
g 5  
e 6  
7
```

> Loops

```
range(4)
```

```
range(0, 4)
```

```
for i in range(4):  
    print(i)
```

```
0
```

```
1
```

```
2
```

```
3
```

iterate range

```
for i in range(2, 14, 3):  
    print(i)
```

```
2
```

```
5
```

```
8
```

```
11
```

range with step

```
name='Aladdin'
```

```
for i in range(len(name)):  
    print(name[i])
```

```
A
```

```
l
```

```
a
```

```
d
```

```
d
```

```
i
```

```
n
```


> Conditions

```
speed=80

if speed > 100:
    print('too fast')
elif speed > 80 and speed <=100 :
    print('speed ok')
else:
    print('too slow')

too slow
```

← if

← elif

← else

> Conditions

```
exam1 = 3.5  
exam2 = 'B'  
  
if (exam1 >=4 and exam2 == 'A') :  
    print('grades are great')  
elif exam1 < 4 and exam2 == 'B':  
    print('grades are poor')  
else:  
    print('grades are mixed')
```

grades are poor

> Functions

```
def  
calculateArea (length, height) :  
    return length*height
```

```
calculateArea (5, 4)  
20
```

```
calculateArea ('a', 4)  
'aaaa'
```

> Functions

```
def countLetters(strings, letter):  
    count=0  
    for str in strings:  
        count=count+str.count(letter)  
    return count
```

```
countLetters(['day', 'pasta', 'lasagna'], 'a')  
6
```

> Classes

```
class Vehicle(object):  
    wheels=0  
  
    def __init__(self, wheels, maxSpeed=0):           'constructor'  
        self.wheels=wheels  
        self.maxSpeed=maxSpeed  
  
    def fasterThan(self, otherVehicle):             method  
        return self.maxSpeed > otherVehicle.maxSpeed  
  
v1=Vehicle(2)                                       instantiation  
print(v1.maxSpeed)  
0
```

> Classes

```
class Bike(Vehicle):  
    def __init__(self, maxSpeed=0):  
        self.wheels=2  
        self.maxSpeed=maxSpeed
```

```
b1=Bike(30)  
b1.wheels  
b1.maxSpeed  
30
```

```
b2=Vehicle(4, 80)  
b1.fasterThan(b2)  
False
```

> Error

```
def calculateArea(length,height):  
    return length*height
```

```
calculateArea('a','dsd')
```

```
TypeError Traceback (most recent call last)  
<ipython-input-2-2ccc00b376b5> in <module>  
----> 1 calculateArea('a','dsd')
```

```
<ipython-input-1-bf7816c63e4e> in calculateArea(length,  
height)
```

```
    1 def calculateArea(length,height):  
----> 2 return length*height
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

> Assertions

```
def calculateArea(length,height):  
    assert length > 0 , 'length must be positive'  
    assert height > 0 , 'height must be positive'  
    assert type(length) == float  
calculateArea(-3.0,2)  
-----  
(-----  
(AssertionError Traceback (most recent call last)  
<ipython-input-15-f962da30df2f> in <module>  
----> 1 calculateArea(-3.0,2)  
  
<ipython-input-12-f5dade16329c> in calculateArea(length, height)  
      3 #assert type(height) == float  
      4  
----> 5 assert length > 0 , 'length must be positive'  
      6 return length*height  
AssertionError: length must be positive
```


> Questions?

> Install Python

<https://www.anaconda.com/distribution/>

 Windows |  macOS |  Linux

Anaconda 2019.10 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (462 MB)

32-Bit Graphical Installer (410 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)

> Install Python

Open console as admin:

```
> conda --version
```

Install libraries:

```
> conda install -c conda-forge
```

```
geopandas
```

```
> conda install -c conda-forge
```

```
geoplot
```

```
> conda install -c conda-forge osmnx
```

```
> conda install -c conda-forge pysal
```

```
> conda install -c conda-forge
```

> Running Jupyter



```
> jupyter notebook
```

jupyter Quit Logout

Files Running Clusters

Select items to perform actions on them. Upload New ▾ ↻

0 ▾ / OneDrive / pubs / tutorial / GIS2019 / python Name ↑ Last Modified File size

<input type="checkbox"/>	..		seconds ago
<input type="checkbox"/>	loops.ipynb	Running	4 hours ago 3.33 kB
<input type="checkbox"/>	lists.ipynb	Running	13 hours ago 3.98 kB
<input type="checkbox"/>	functions.ipynb	Running	3 hours ago 1.87 kB
<input type="checkbox"/>	datatypes.ipynb	Running	2 hours ago 6.5 kB
<input type="checkbox"/>	conditions.ipynb	Running	3 hours ago 1.75 kB
<input type="checkbox"/>	classes.ipynb	Running	2 hours ago 2.13 kB
<input type="checkbox"/>	assertions.ipynb	Running	3 hours ago 3.02 kB

hes.
so
you.

School of Management
Route de la Plaine 2
3960 Sierre

hevs.ch/heg



Thank you for your attention.

swissuniversities

