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

swissuniversities

> 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

In this course:

- Online environment: Jupyter Python in renkulab.io
- Jupyter Python in Local installation

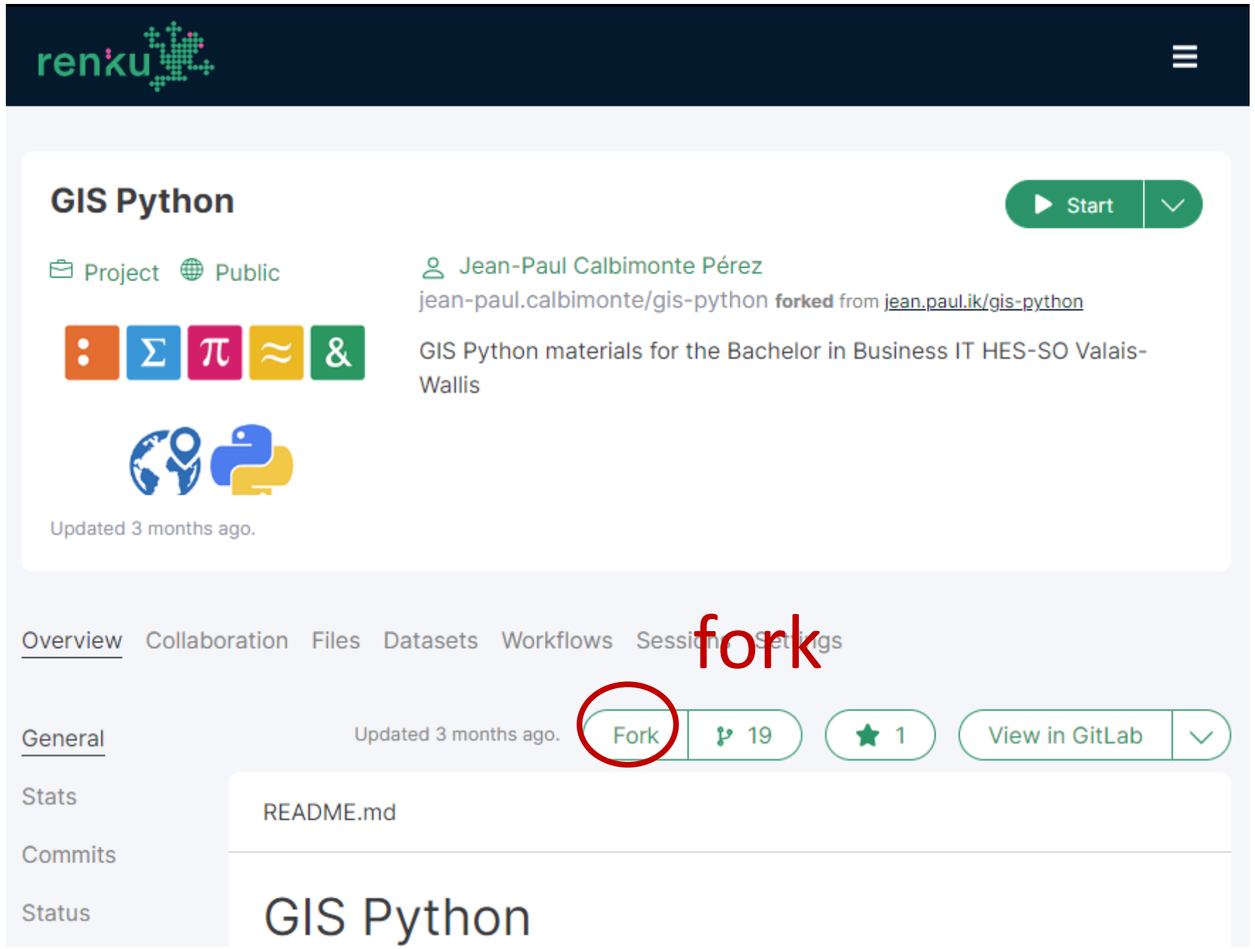
> renkulab.io

Log in with your hevs edu-ID account

> renkulab.io

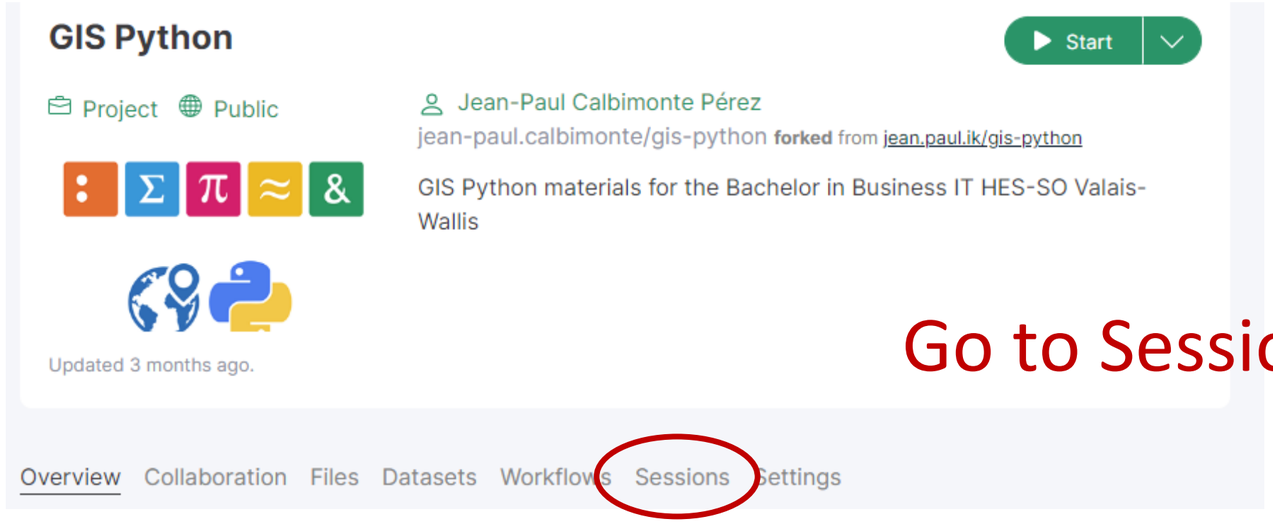
Got to this project and fork it:

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

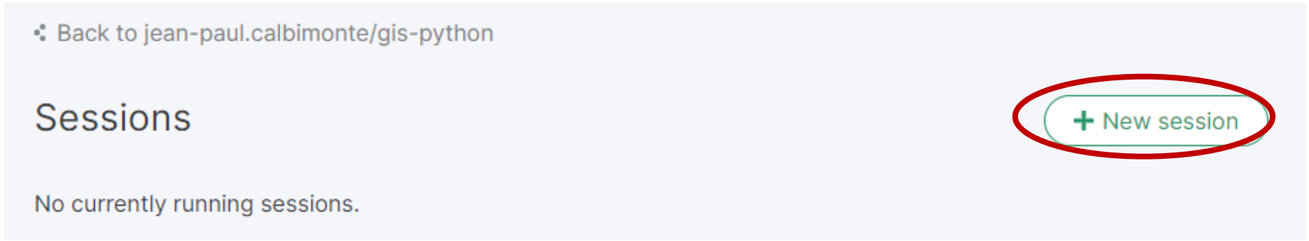


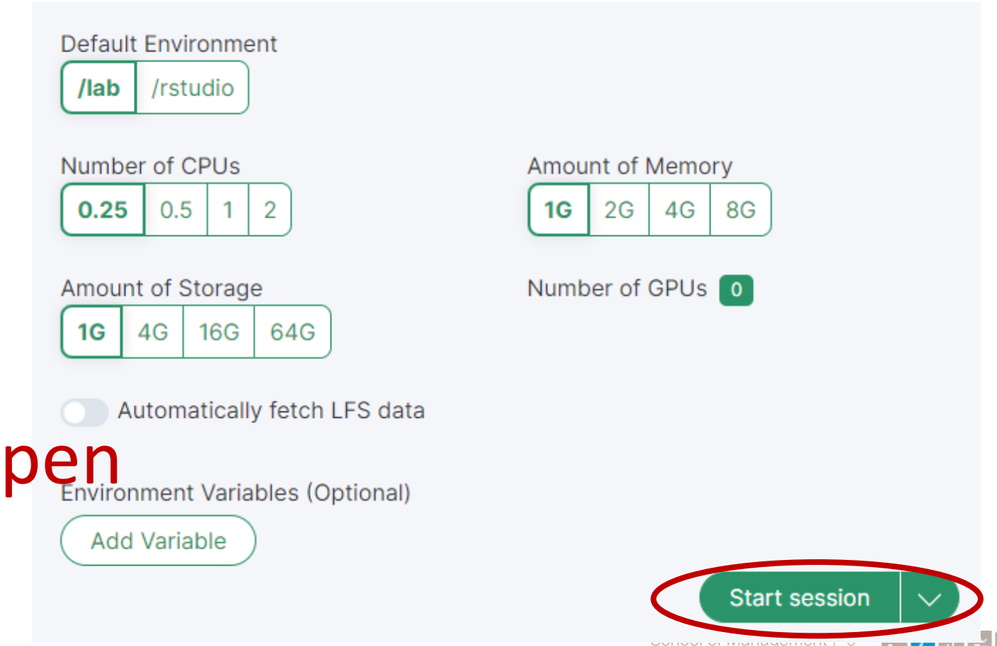
> renkulab.io

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



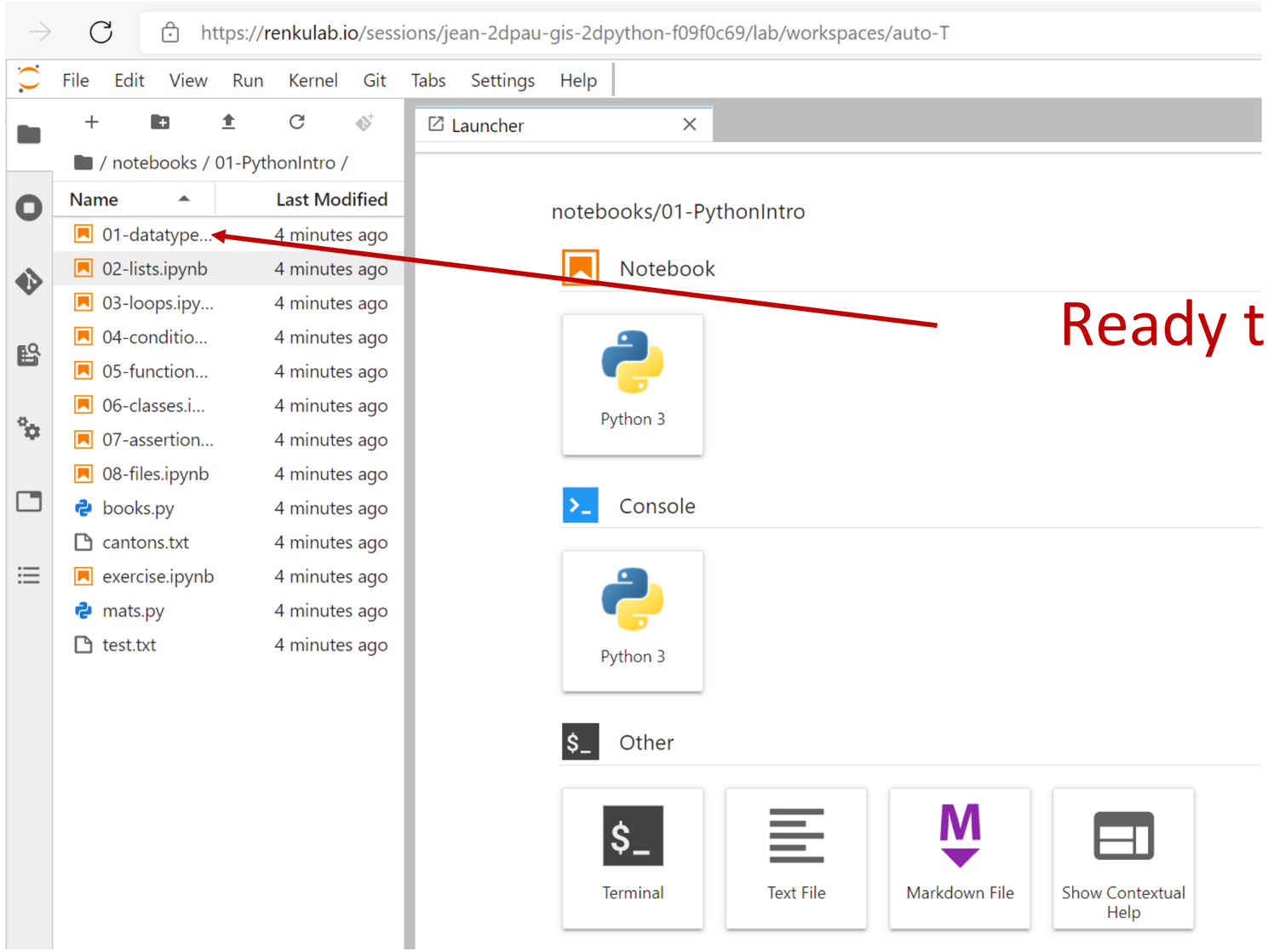
Go to Sessions





Start a new session and open it when it is ready

> renkulab.io



Ready to open the notebooks

> Running Jupyter

Basics of Python

> Basic operators

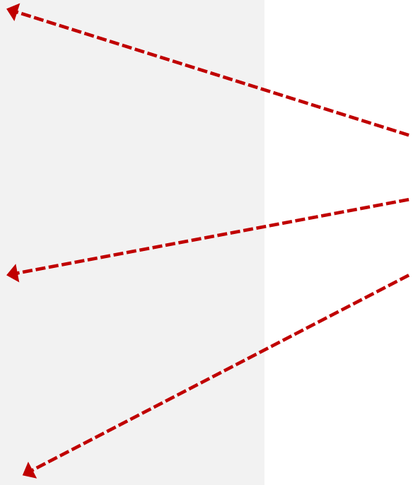
```
1+1
2


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
a+b
3.4
```

int
float

```
a==b
False
a==1
True
```

bool

```
st='we like GIS'
print(st)
we like GIS
```

str

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

unidimensional

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

multidimensional

```
list=[3,4.5,6,2,2.1]
len(list)
5

list[2]
6
list[-1]
2.1

type(list[1])
float
```

indices

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):
        self.wheels=wheels
        self.maxSpeed=maxSpeed

    def fasterThan(self,otherVehicle):
        return self.maxSpeed > otherVehicle.maxSpeed



v1=Vehicle(2)
print(v1.maxSpeed)
0
```

'constructor'

method

instantiation

> 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?

hes·
so
you.

Thank you for your attention.

School of Management
Route de la Plaine 2
3960 Sierre