

Programmazione 1

Prima Laboratorio: ABC

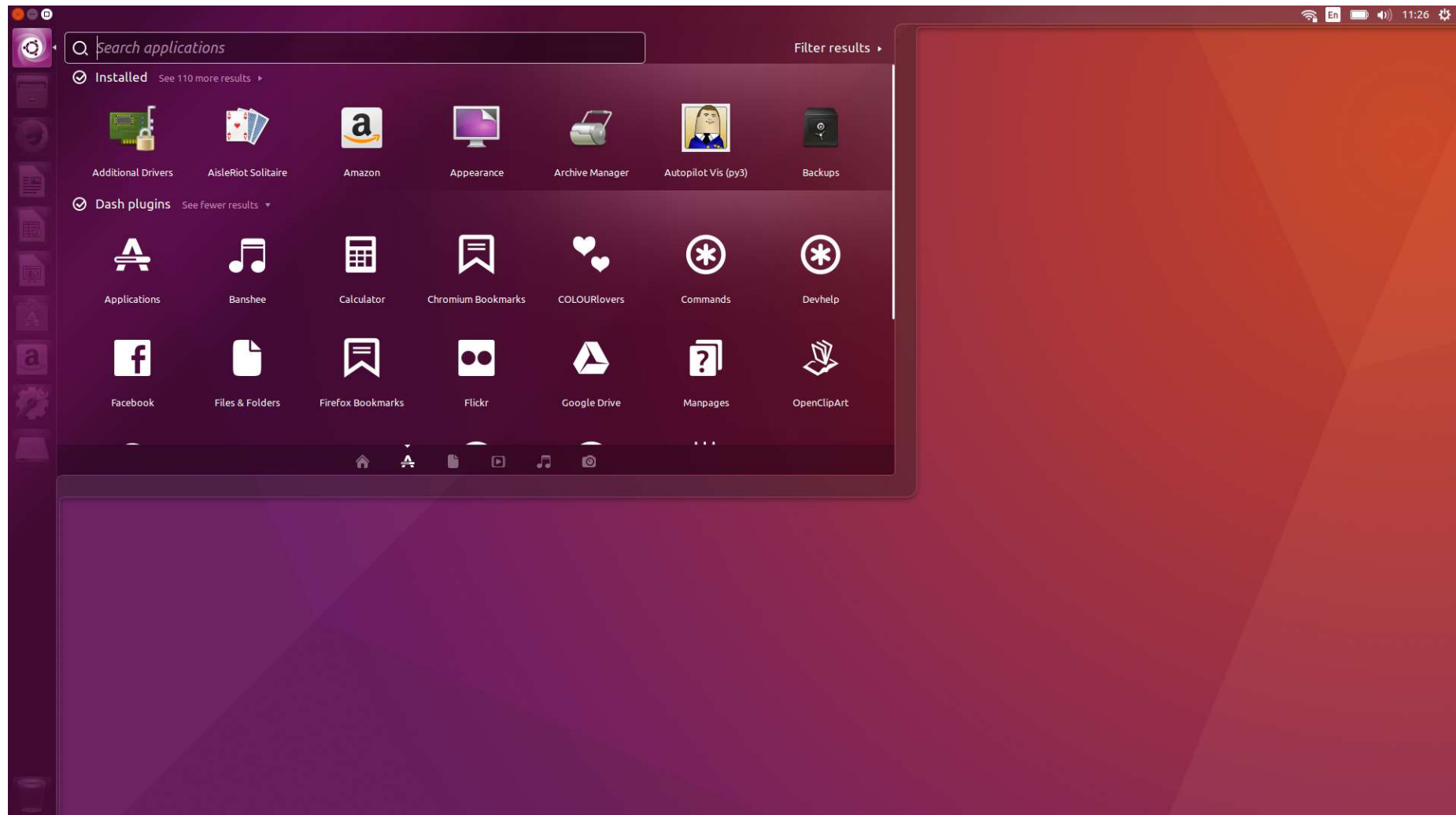
Aula C3, 05/10/2018

Scaricare i notebook dal sito

- Lavorare a coppie (si chiama **Pair Programming** https://en.wikipedia.org/wiki/Pair_programming)
- Aprire un browser (Chrome o Firefox) e andare a:
<http://mate.unipv.it/gualandi/programming>
- Cliccare su «Notebook Lab 2 (sorgente)»
- Con il tasto destro del mouse cliccare «salva con come»
- Controllare che il file sia salvato con l'estensione .ipynb (non deve essere un file .txt)

Per usare il notebook

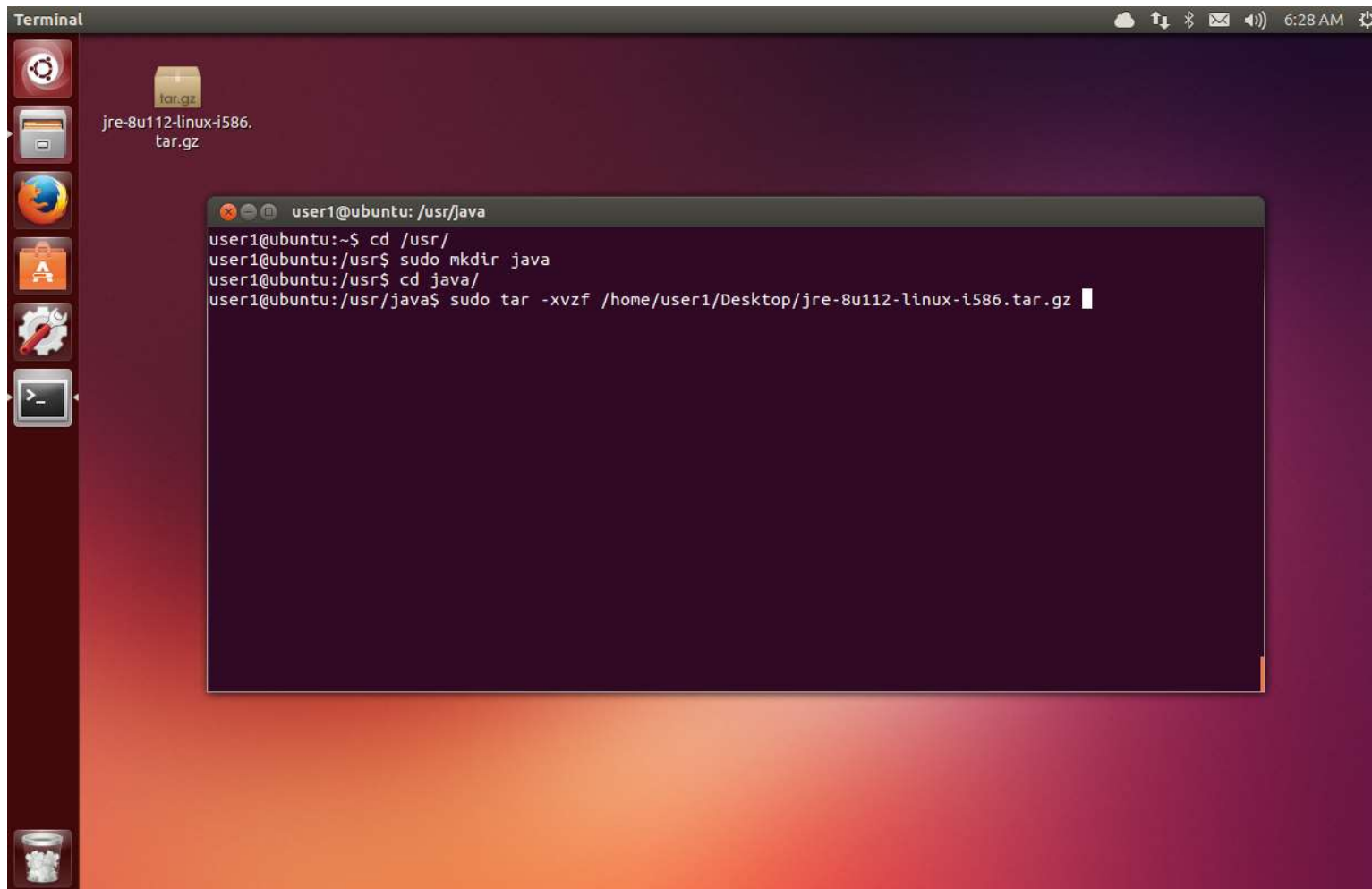
Aprire un Terminale (Search Applications -> terminal)



Per usare il notebook

Aprire un Terminale e digitare:

\$



The screenshot shows an Ubuntu desktop with a dark purple background. On the left is a vertical dock with icons for Dash, Home Folder, Firefox, LibreOffice, and the Dash icon. On the desktop, there is a folder icon labeled 'tar.gz' and a file icon labeled 'jre-8u112-linux-i586.tar.gz'. A terminal window is open in the foreground, titled 'user1@ubuntu: /usr/java'. The terminal shows the following commands and output:

```
user1@ubuntu:~$ cd /usr/  
user1@ubuntu:/usr$ sudo mkdir java  
user1@ubuntu:/usr$ cd java/  
user1@ubuntu:/usr/java$ sudo tar -xvzf /home/user1/Desktop/jre-8u112-linux-i586.tar.gz
```

Per usare il notebook

Comandi Linux utili da terminale Terminale:

```
$ ls                                # lista di file
```

```
$ cd                                # change directory
```

```
$ cd NomeCartella                 # vai nella directory ..
```

```
$ cd ..                             # Sali di una cartella
```

```
$ cp File1 File2                   # copia File1 in File2
```

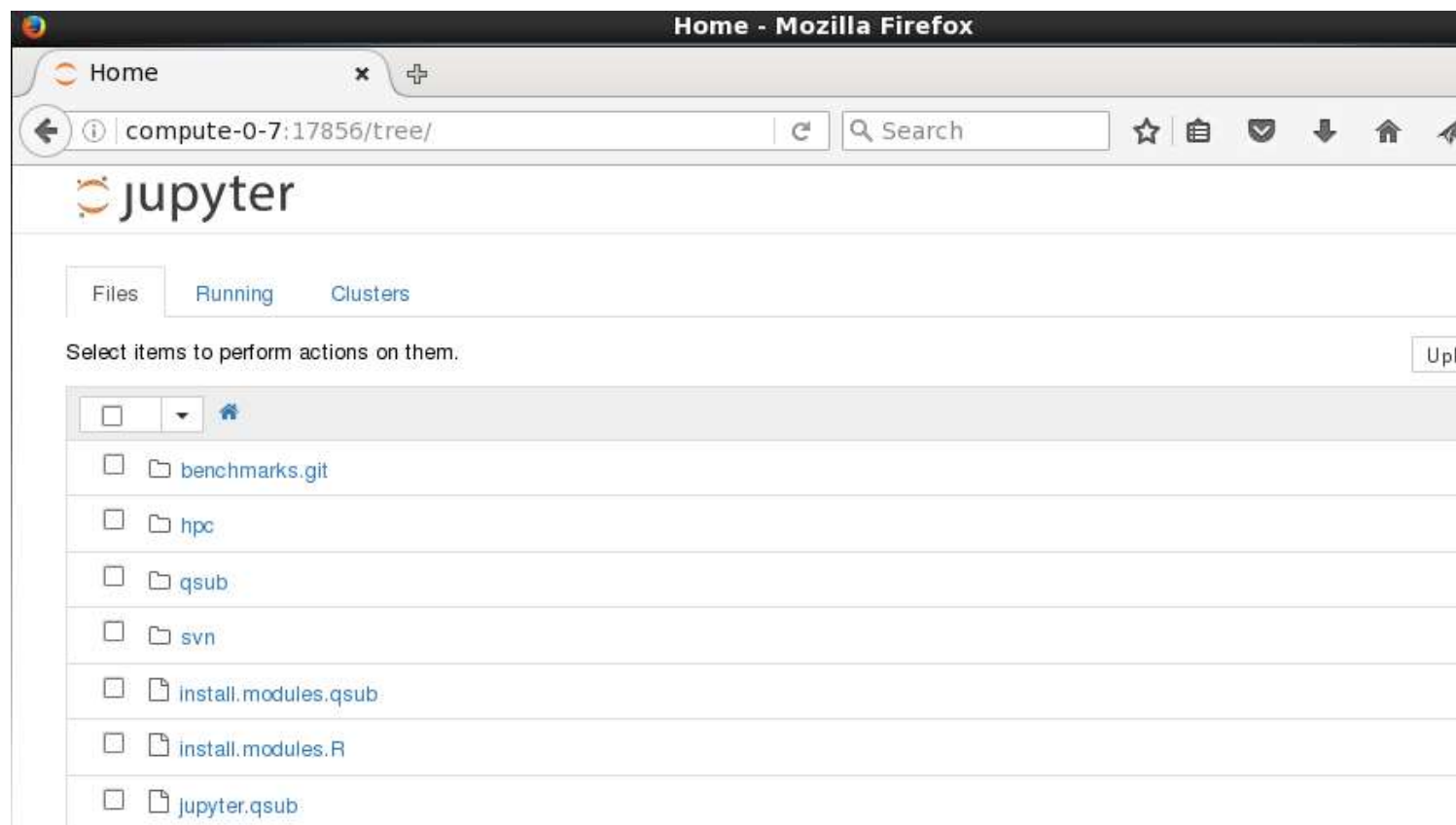
```
$ mv F.txt F.ipynb                # cambia nome del file
```

```
$ rm F.txt                         # cancella file
```

Per usare il notebook

Aprire un Terminale, andare nella cartella in cui avete salvato il notebook Lab2 e digitare:

```
$ ipython3 notebook
```



Guida di riferimento

1/2

Command Mode (press Esc to enable)

Enter	enter edit mode
Shift-Enter	run cell, select below
Ctrl-Enter	run cell
Alt-Enter	run cell, insert below
Y	to code
M	to markdown
R	to raw
1	to heading 1
2,3,4,5,6	to heading 2,3,4,5,6
Up/K	select cell above
Down/J	select cell below
A/B	insert cell above/below
X	cut selected cell
C	copy selected cell
Shift-V	paste cell above
V	paste cell below
Z	undo last cell deletion
D,D	delete selected cell

Edit Mode (press Enter to enable)

Tab	code completion or indent
Shift-Tab	tooltip
Ctrl-]	indent
Ctrl-[dedent
Ctrl-A	select all
Ctrl-Z	undo
Ctrl-Shift-Z	redo
Ctrl-Y	redo
Ctrl-Home	go to cell start
Ctrl-Up	go to cell start
Ctrl-End	go to cell end
Ctrl-Down	go to cell end
Ctrl-Left	go one word left
Ctrl-Right	go one word right
Ctrl-Backspace	delete word before
Ctrl-Delete	delete word after
Esc	command mode
Ctrl-M	command mode

Guida di riferimento

2/2

Python For Data Science Cheat Sheet Jupyter Notebook

Learn More Python for Data Science Interactively at www.DataCamp.com



Saving/Loading Notebooks

Create new notebook

Make a copy of the current notebook

Save current notebook and record checkpoint

Preview the printed notebook

Close notebook & stop running any scripts

Open an existing notebook

Rename notebook

Revert notebook to a previous checkpoint

Download notebook as

- IPython notebook
- Python
- HTML
- Markdown
- reST
- LaTeX
- PDF

Writing Code And Text

Code and text are encapsulated by 3 basic cell types: markdown cells, code cells, and raw NBConvert cells.

Edit Cells

Cut currently selected cells to clipboard

Paste cells from clipboard above current cell

Paste cells from clipboard on top of current cell

Revert "Delete Cells" invocation

Merge current cell with the one above

Adjust metadata underlying the current notebook

Remove cell attachments

Paste attachments of current cell

Copy cells from clipboard to current cursor position

Paste cells from clipboard below current cell

Delete current cells

Split up a cell from current cursor position

Merge current cell with the one below

Move current cell down

Find and replace in selected cells

Copy attachments of current cell

Insert image in selected cells

Insert Cells

Add new cell above the current one

Add new cell below the current one

Working with Different Programming Languages

Kernels provide computation and communication with front-end interfaces like the notebooks. There are three main kernels:



Installing Jupyter Notebook will automatically install the IPython kernel.

Restart kernel

Restart kernel & run all cells

Restart kernel & run all cells

Interrupt kernel

Interrupt kernel & clear all output

Connect back to a remote notebook

Run other installed kernels

Command Mode:



Edit Mode:



1. Save and checkpoint
2. Insert cell below
3. Cut cell
4. Copy cell(s)
5. Paste cell(s) below
6. Move cell up
7. Move cell down
8. Run current cell
9. Interrupt kernel
10. Restart kernel
11. Display characteristics
12. Open command palette
13. Current kernel
14. Kernel status
15. Log out from notebook server

Executing Cells

Run selected cell(s)

Run current cells down and create a new one above

Run all cells above the current cell

Change the cell type of current cell

toggle, toggle scrolling and clear all output

Run current cells down and create a new one below

Run all cells

Run all cells below the current cell

toggle, toggle scrolling and clear current outputs

View Cells

Toggle display of Jupyter logo and filename

Toggle line numbers in cells

Toggle display of toolbar

Toggle display of cell action icons:

- None
- Edit metadata
- Raw cell format
- Slideshow
- Attachments
- Tags

Widgets

Notebook widgets provide the ability to visualize and control changes in your data, often as a control like a slider, textbox, etc.

You can use them to build interactive GUIs for your notebooks or to synchronize stateful and stateless information between Python and JavaScript.

Download serialized state of all widget models in use

Save notebook with interactive widgets

Embed current widgets

Asking For Help

Walk through a UI tour

Edit the built-in keyboard shortcuts

Description of markdown available in notebook

Python help topics

NumPy help topics

Matplotlib help topics

Pandas help topics

List of built-in keyboard shortcuts

Notebook help topics

Information on unofficial Jupyter Notebook extensions

IPython help topics

SciPy help topics

SymPy help topics

About Jupyter Notebook



Esercizi da svolgere nel notebook

Svolgere il seguente esercizio in un nuovo notebook:

- Scrivere una funzione che prende in input un numero r (il raggio) e calcola il perimetro di una circonferenza. Usare la funzione con $r=13$, $r=19$, e $r=23$. Memorizzare i valori dei tre perimetri in tre variabili diverse e stampare a video i tre valori corrispondenti
- Svolgere gli esercizi 1.3, 1.4 e 1.5 del notebook “Lab 2”

Elementi del linguaggio introdotti

1. Numeri interi, variabili boolean
2. Operatori aritmetici e booleani, in notazione

1. Prefix: add, sub, mul, truediv

2. Postfix: +, -, *, /, and, or, not

3. La definizione di semplici **funzioni**

```
def <NomeProcedura>(<parametri formali>):  
    <corpo della procedura>  
    return <valore>
```

4. Il comando **who**, **print()**

5. Le espressioni condizionali

```
if <espressione logica 1>:  
    <body1>  
elif <espressione logica 2>:  
    <body2>  
else:  
    <body3>
```

Per usare Spyder

Aprire un Terminale e digitare:

\$ **spyder3**

Dovete avere >= Python 3.4 !

The screenshot displays the Spyder Python IDE interface. The top toolbar shows the 'Run' button (a green play icon) circled in red. The main window is divided into several panels:

- Code Editor:** Contains a Python script named `questionnaire.py`. The script imports `pandas` and `DataFrame`, reads a CSV file, and performs data manipulation. The code is as follows:

```
3 Created on Wed Apr 29 09:29:09 2015
4
5 @author: erik
6 """
7
8 import pandas as pd
9
10 from pandas import DataFrame
11
12 def reverseScoring(df, high, cols):
13     df[cols] = high - df[cols]
14     return df
15
16
17 #The questionnaire is coded in swedish. Changing stuff to english...
18 iv = ['Y' + str(i) for i in range(1,21)]
19 firstnames, lastnames = ['Date', 'Informed', 'Sex', 'Age'], ['EduYears', 'Sub_id']
20 names = firstnames + iv + lastnames
21 frame = pd.read_csv('acs_okt.csv', skiprows=1, names=names, encoding='latin6')
22
23 print frame.head()
24
25 #Removing strings and leaving score (1-4)
26 #Going col by col
27 for idx in range(len(iv)):
28     for i,row in enumerate(frame[iv[idx]]):
29         #Had to turn row into string (was unicode type)
30         row = str(row)
31         frame[iv[idx]][i] = int(row[0])
32
33
34 #print frame[['Y1', 'Y2']]
35 frame['Sub_id'] = range(1, len(frame['Date'])+1)
36 #grouped = frame.groupby('Sex')
37
38 #Save the frame to csv
39 #frame.to_csv(path_or_buf="ACS_DATA.csv", sep=';')
40 print "Done"
41
42 toReverse = ['Y1', 'Y2', 'Y3', 'Y6', 'Y7', 'Y8', 'Y11', 'Y12', 'Y15', 'Y16',
43             'Y20']
44
45 revFrame = reverseScoring(frame, 5, toReverse)
46
47 revFrame.to_csv(path_or_buf="ACS_DATA_rev.csv", sep=';')
```
- Variable Explorer:** Shows the variables defined in the script. The `frame` variable is a `DataFrame` with 234 rows and 26 columns. The `names` variable is a list of 26 strings. The `row` variable is a `unicode` string.
- IPython Console:** Displays the output of the script. It shows a `UnicodeEncodeError` and a `SettingWithCopyWarning`. The error message is: `UnicodeEncodeError: 'ascii' codec can't encode characters in position 11-12: ordinal not in range(128)`. The warning message is: `/home/erik/Dokument/Programming/Python/datawrangling/questionnaire.py:31: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame`.

Esercizi da svolgere in Spyder3

Si considerino le due procedure seguenti contenute nel file `IntroSpyder.py` (da scaricare dal sito del corso):

```
def Inc(x):  
    return x+1
```

```
def Dec(x):  
    return x-1
```

Si implementi una funzione **Somma(x, y)** che calcola la somma $x+y$, ma che utilizza solo chiamate ricorsive, le espressioni condizionali e le due funzioni **Inc(x)** e **Dec(x)**

Esercizio Facoltativo

- Si implementi una funzione **RadiceIntera (x)** che prende in input un numero **$x > 1$** e che, utilizzando solo gli elementi del linguaggio sin qua introdotti, calcoli:
 - se la radice intera esiste, la calcola e restituisce il valore corrispondente
 - altrimenti, restituisce il valore -1
- Implementare l'algoritmo del “*contadino russo*” visto durante la prima lezione