Scientific Programming in Python

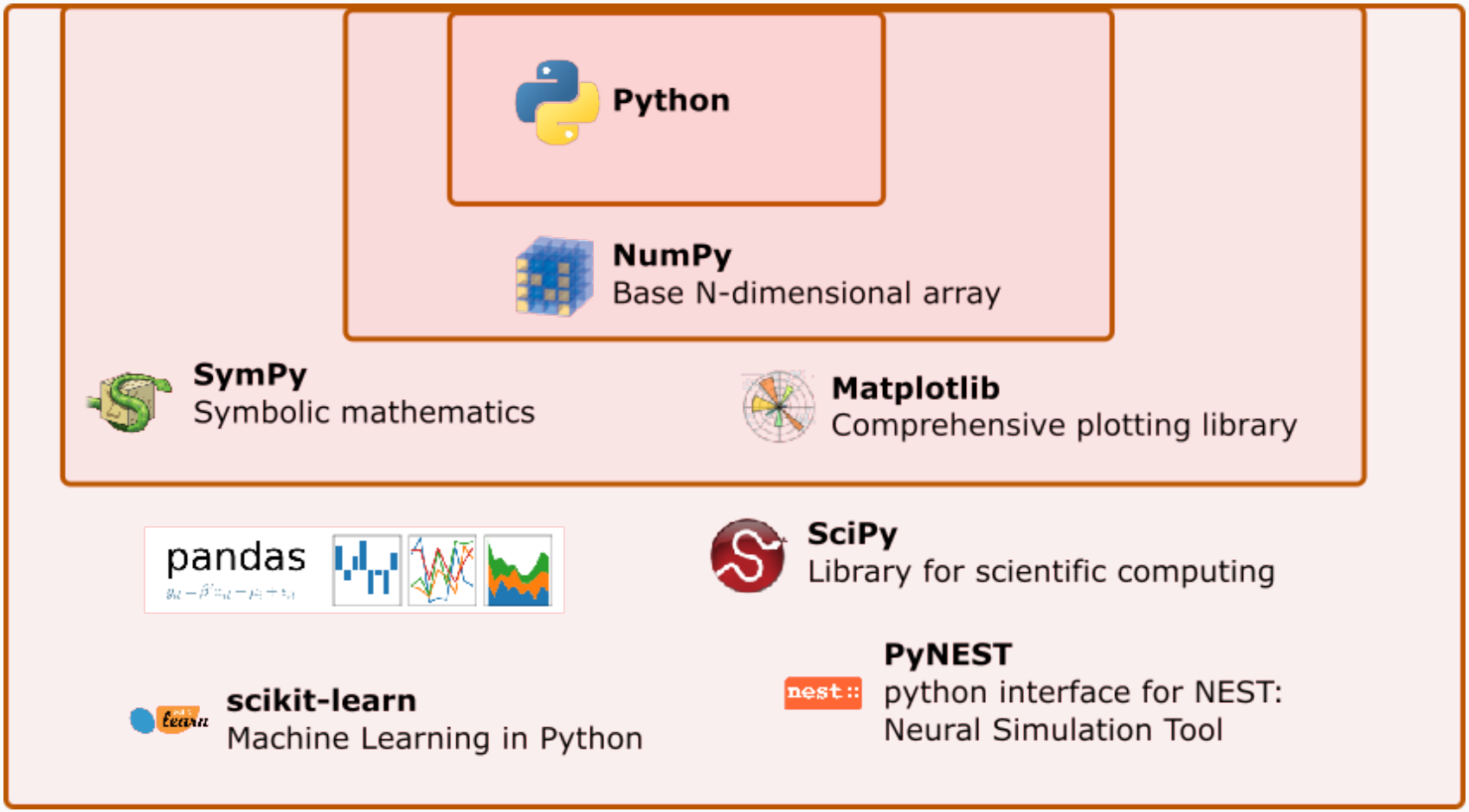
Francesco Mannella

Institute of Cognitive Sciences and technologies - CNR

Outline

Outline

Python basics - Python's scientific ecosystem



Python basics - the interpreter

Just type "python" in the command line to open:

```
$ python
Python 2.7.12 (default, Nov 19 2016,
      06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "
      license"
for more information.
>>>
```

It can be used as a calculator:

```
>>> 4 + 5
9
>>> _
```

You can execute single lines of code

```
>>> a = 5
>>> b = 3
>>> a + b
8
>>>
```

You can also run a control statement:

```
>>> for i in range(10):
...     print i**2
d..
0
1
4
9
16
25
36
49
64
81
>>>
```

Outline

Numeric types - hands-on code

Code: find if a number is a prime number

```
1  # Python program to check if the input number
2  # is prime or not
3
4  from __future__ import print_function
5  from __future__ import division
6
7  num = 1
8
9  # take input from the user
10 # num = int(input("Enter a number: "))
11
12 # prime numbers are greater than 1
13 if num > 1:
14    # check for factors
15    for i in range(2,num):
16        if (num % i) == 0:
17            print(num,"is not a prime number")
18            print(i,"times",num//i,"is",num)
19            break
20    else:
21        print(num,"is a prime number")
22
23 # if input number is less than
24 # or equal to 1, it is not prime
25 else:
26    print(num,"is not a prime number")
```

```
$ python prime.py
407 is not a prime number
11 times 37 is 407
```

Numeric types

```
# Boolean

a = True
b = False

# Integer

a = 34
b = 45//3
c = int(34/2)
d = int(3.14)
e = a + b//d
```

```
# Float

a = 2.3
b = 45/3
c = float(34)
d = b/a

# Complex

a = complex(1, 2)
b = 3 -4.5j
c = a**2 + b
d = c.real() # float
e = c.imag() # float
```

Outline

Operators - arithmetic operators

`a = 10; b = 20`

`a + b == 30`	**Addiction**	Adds values on either side of the operator.
`b - a == 10`	**Subtraction**	Subtracts right hand operand from left hand operand.
`a * b == 200`	**Multiplication**	Multiplies values on either side of the operator.
`b / a == 2` `a / 4.0`	**Division**	Divides left hand operand by right hand operand.
`b % a == 0` `a / 4.0`	**Modulus**	Divides left hand operand by right hand operand and returns remainder.
`a ** b`	**Exponent**	Performs exponential (power) calculation on operators.
`a // 4.0`	**Floor Division**	Division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity)

Operators - assignement operators

`a += b`	**Addiction**	Adds values on either side of the operator and assigns the result to left operand.
`b -= a`	**Subtraction**	Subtracts right hand operand from left hand operand and assigns the result to left operand.
`a *= b`	**Multiplication**	Multiplies values on either side of the operator and assigns the result to left operand.
`b /= a`	**Division**	Divides left hand operand by right hand operand and assigns the result to left operand.
`b %= a`	**Modulus**	Divides left hand operand by right hand operand, assigns the result to left operand and returns remainder.
`a **= b`	**Exponent**	Performs exponential (power) calculation on operators and assigns the result to left operand.
`a //= 4.0`	**Floor Division**	Perform floor Division and assigns the result to left operand.

Operators - comparison operators

`a == b`	**False**	Is *a* equal to *b*?
`a != b` `a <> b`	**True**	Is *a* different from *b*?
`a < b`	**True**	Is *a* less than *b*?
`a > b`	**False**	Is *a* greater than *b*?
`a <= b`	**True**	Is *a* less than or equal to *b*?
`a >= b`	**False**	Is *a* greater than or equal to *b*?

Operators - logical operators

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

`a and b`	**False**	are both *a* and *b* true?
`a or b`	**True**	is *a* or *b* true?
`not a`	**False**	is not *a* true?

Outline

Strings

```
name = "Federico"
surname = "Rossi"
street = 'via' + 'A. De Sanctis'
number = 6
zipcode = 100
city = "Roma"

address = """
Federico Rossi
via A. De Sanctis, 6
00100 Roma
"""
```

```
address0 = "%s %s\n%s, %d\n%05d %s" % (name,
       surname, street, number, zipcode, city)

address1 = "{} {}\n{}, {}\n{:05d} {}".format(
       name, surname, street, number, zipcode,
       city)

address2 = """
{} {}
{} {}
{:05d} {}
""".format(name, surname, street, number,
       zipcode, city )

address3 = r"{} {}\n{}, {}\n{:05d} {}\".format(
       name, surname,  street, number, zipcode,
        city)
```

```
>>> print(address0)
Federico Rossi
via A. De Sanctis, 6
00100 Roma
>>>
>>> print(address1)
Federico Rossi
via A. De Sanctis, 6
00100 Roma
>>>
```

```
>>> print(address2)
Federico Rossi
via A. De Sanctis, 6
00100 Roma
>>> print(address3)
Federico Rossi\nvia A. De Sanctis, 6\n00100
       Roma
>>> _
```

Outline

Control statements

IF statement

```
print_on = True
if print_on :
    print('print this!')

if print_on is True:
    print('print this!')
```

```
if a > 0.5:
    b += a
```

```
a = 7
if a < 5:
    print('less then 5')
elif 5 <= a < 10:
    print('between 5 and 10')
elif 10 <= a <= 15:
    print('between 10 and 15')
else:
    print('greater than 15')
```

FOR loops

```
for i in range(10):
    print(i)
```

```
for i in range(10):
    if i == 8:
        continue
    print(i)
```

```
a = []
for i in range(5):
    a += i

a == [0, 1, 2, 3, 4]
```

WHILE loops

```
i = 1
while i < 6:
    print(i)
    i += 1
```

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```