FEniCS Course

Lecture 0.5: Python programming

Contributors
Anders Logg

fenics
project

What is Python?

The Python programming language is:

- General purpose
- Imperative
- Object-oriented
- High-level
- Slow
- Easy

Computing $1 + 2 + \ldots + 100$ in Python

```python
s = 0

for i in range(1, 101):
    s += i

print s
```

Running the program

Bash code

```
$ python sum.py
5050
```

Performance in Python vs C++

Let's compute $\sum_{k=1}^{N} k$ for $N = 100,000,000$.

Bash code

```
$ time python sum.py
5000000050000000

real   0m13.243s
```

Bash code

```
$ time ./sum
5000000050000000

real   0m0.287s
```

Python/FEniCS programming 101

1. Open a file with your favorite text editor (Emacs :-)) and name the file something like `test.py`

2. Write the following in the file and save it:

   ```
   from fenics import *
   ```

3. Run the file/program by typing the following in a terminal (with FEniCS setup):

   ```
   $ python test.py
   ```

Python basics

Structure of a Python program

```
import stuff

def some_function(argument):
    "Function documentation"
    return something

# This is a comment
if __name__ == "__main__":
    do_something
```

Declaring variables

```
a = 5
b = 3.5
c = "hej"
d = ’hej’
e = True
f = False
```

Illegal variable names

```
and, del, from, not, while, as, elif, global, or,
with, assert, else, if, pass, yield, break, except,
import, print, class, exec, in, raise, continue,
finally, is, return, def, for, lambda, try
```

Comparison

```
x == y
x != y
x > y
x < y
x >= y
x <= y
```

Logical operators

```
not x
x and y
x or y
```

If

```
if x > y:
    x += y
```

If / else

```
if x > y:
    x += y
    y += x
elif x < y:
    x += 1
else:
    y += 1
```

For loop

```
for variable in enumerable:
    stuff

for i in range(100):
    stuff

for i in range(100):
    stuff
    morestuff
```

While loop

```
while condition:
    stuff

i = 0
while i < 100:
    stuff
    i++

i = 0
while True:
    stuff
    if i == 99:
        break
```

Functions

```python
def myfunction(arg0, arg1, ...):
    stuff
    ...
    return something # or not, gives None

def sum(x, y):
    return x + y
```

Plotting

Matplotlib gives MATLAB-like plotting

```
from pylab import *

plot(x, y)
xlabel(’x’)
ylabel(’y’)
title(’My figure’)
grid(True)
savefig(’myfigure.png’)
savefig(’myfigure.pdf’)
```

Python classes

Class structure

```python
class Foo:

    def __init__(self, argument):
        stuff

    def foo(self):
        stuff
        return something

    def bar(self):
        stuff
        return something

f = Foo(argument)
f.foo()
f.bar()
```

Class members

```
class Foo:

    def __init__(self, argument):
        self.x = 3  # this is a member variable

    def foo(self):  # this is a member function
        stuff
```

Public and private class members

```
class Foo:

    def __init__(self, argument):
        self.x = 3   # public member variable
        self.__x = 3 # private member variable
```

Python exercises

Exercises

- Write a program that generates the sequence $(x_n)_{n=0}^{100}$ for $x_n = n$.
- Write a program that generates the odd numbers between 1 and 100.
- Write a program that computes the sum $\sum_{n=0}^{100} x_n$ for $x_n = n$.
- Write a program that computes the sum of the odd numbers between 1 and 100.
- Write a program that generates all prime numbers between 2 and 1000.
- Write a program that generates the first 1000 prime numbers.
- Write a program that computes the approximation $\sqrt{2} \approx x_{100}$ for $x_n = (x_{n-1} + 2/x_{n-1})/2$ and $x_0 = 1$.
- Write a program that computes the approximation $\sqrt{2} \approx x_N$ for $x_n = (x_{n-1} + 2/x_{n-1})/2$ and $x_0 = 1$ where N is the smallest number such that $|x_N - x_{N-1}| < 10^{-10}$.
- Write a program that generates the sequence $(x_n)_{n=0}^{N}$ for $N = 10^6$ when

  $$x_n = 4 \sum_{k=0}^{n} (-1)^k/(2k+1). \tag{1}$$

  Does it seem to converge to some particular number?