Matlab ⇔ *Python*

July 26, 2017

Matlab

```matlab
disp("Hello World!")
```

Python

```python
print("Hello World!")
```

Comments

Matlab

```matlab
% this is a comment

%{
   This is a long comment in Matlab 7
%}
```

Python

```python
# this is a comment

"""
   This is a long comment
"""
```

Variables

Matlab

```matlab
x = 5.71;
I = besseli(x,A);
A = [1 2 3; 4 5 6; 7 8 9];
```

Python

```python
x = 5.71

import numpy as np
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

I = besseli(x,A)
```

String Formatting

Matlab

```
A = pi*ones(1,3);
txt = sprintf('%f | %.2f | %12f', A);
```

Python

```
A = np.pi*np.ones(3)
text = "{:f} | {:.2f} | {:12f}".format(*A)
```

User Input

Matlab

```matlab
x = input(prompt)
str_input = input(prompt,'s')
```

Python

```python
str_input = input(prompt)
```

Selection Statements

Matlab

```
if x > 5
    y = 2;
elseif x < 0
    y = 8;
else
    y=24;
end
```

Python

```
if x>5:
    y=2
elif x<0:
    y=8
else
    y=24
```

For Loops

Matlab

```
x = ones(1,10);
for n = 2:6
    x(n) = 2 * x(n - 1);
end
```

Python

```
x = np.ones(10)
for n in range(2,7):
    x[n] = 2*x[n-1]
```

While Loops

Matlab

```
n = 1;
nFactorial = 1;
while nFactorial < 1e100
    n = n + 1;
    nFactorial = nFactorial *
end
```

Python

```
n = 1
nFactorial = 1
while nFactorial < 1e100:
    n+=1
    nFactorial*=n
```

Defining Functions

Matlab

```
function f = fact(n)
    f = prod(1:n);
end
```

Python

```
def fact(n):
    f = np.prod(np.arange(1,n+1))
    return f
```

Using Functions

Matlab

```
# Put file in same folder or on path and then just call

myfunction(x)
```

Python

```
from myfile import myfunction

myfunction(x)
```

Creating Classes: Matlab

```matlab
classdef BasicClass
   properties
      Value
   end
   methods
        function obj = BasicClass(val)
            if val > 0
                obj.Value = val;
            else
                error('Value must be numeric')
        end
    end
end
```

Creating Classes: Matlab cont.

```
classdef BasicClass
        ...
    methods
        ...
        function r = roundOff(obj)
            r = round([obj.Value],2);
        end
        function r = multiplyBy(obj,n)
            r = [obj.Value] * n;
        end
    end
end
```

Creating Classes: Python

```
class BasicClass(object):
    def __init__(self, value):
        if val>0:
            self.value = value
        else:
            raise ValueError("Value must be positive")

    def roundOff(self):
        return round(self.Value, 2)

    def multiplyBy(self, n):
        return self.Value * n
```

Creating Objects

Matlab

```
a = BasicClass(pi/3)
roundOff(a)
multiplyBy(a,3)
a.multiplyB(3)
```

Python

```
a = BasicClass(np.pi/3)
a.Value = np.pi/2
a.roundOff()
a.multiplyBy(3)
```

Let's mix?

Python in Matlab	`https://www.mathworks.com/help/matlab/call-python-libraries.htm`
Matlab in Python	`https://www.mathworks.com/help/matlab/matlab_external/call-matlab-functions-from-python.html`
Translate	`https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html`