BasicComputationDemo

January 15, 2021

1 Basic Computation Demo

[ ]:
```
import numpy as np
```

[ ]:
```
from bokeh.plotting import figure
from bokeh.io import output_notebook, show
output_notebook()
```

1.1 Simple calculations

[ ]:
```
3+(5*2)/100
```

[ ]:
```
np.sin(2)
```

[ ]:
```
np.pi
```

[ ]:
```
np.sin(np.pi/6)
```

[ ]:
```
np.sqrt(3)
```

[ ]:
```
np.log(2)
```

[ ]:
```
np.exp(1)
```

1.2 Variables

[ ]:
```
a = np.log(2)
```

[ ]:
```
a
```

[ ]:
```
2*a
```

[ ]:
```
# here I calculate the magic constant
a = (3*np.exp(2))/np.sin(3*np.pi/5)
```

[ ]:
```
a
```

1.3 Lists

[ ]:
```
a = [1,2,3,4,5]
```

[ ]:
```
a
```

[ ]:
```
a[0]
```

[ ]:
```
a[1]
```

[ ]:
```
a[2:4]
```

[ ]:
```
a[2], a[3]
```

[ ]:
```
a[-1]
```

[ ]:
```
a[-2]
```

[ ]:
```
a[1:4:2]
```

[ ]:
```
a[1]
```

[ ]:
```
a[3]
```

1.4 Strings

[ ]:
```
f = "Here is a piece of text as a string"
```

[ ]:
```
f
```

[ ]:
```
f[0]
```

[ ]:
```
f[1]
```

[ ]:
```
f[2]
```

[ ]:
```
f = "Here is a piece of string"
```

[ ]:
```
f[-1]
```

1.5 Arrays

[ ]:
```
f = np.array([-3,3,8,10])
```

[ ]:
```
f
```

[ ]:
```
3*f
```

[ ]:
```
1/f
```

[ ]:
```
np.cos(f)
```

[ ]:
```
g = np.array([-1,1,2,5])
```

[ ]:
```
f*g
```

[ ]:
```
f+g
```

[ ]:
```
f.shape
```

[ ]:
```
A = np.array([[1,2],[3,4]])
```

[ ]:
```
A
```

[ ]:
```
A.shape
```

[ ]:
```
f
```

[ ]:
```
A
```

[ ]:
```
f[0]
```

[ ]:
```
f[1]
```

[ ]:
```
A[0]
```

[ ]:
```
A[1]
```

[ ]:
```
A[0,0]
```

[ ]:
```
A[:,0]
```

[ ]:
```
A[:,1]
```

[ ]:
```
A[0,:]
```

[ ]:
```
A[1,:]
```

[ ]:
```
B = np.array([[1,-1,1],[2,5,7],[-5,1,2]])
```

[ ]:
```
B
```

[ ]:
```
B.shape
```

[ ]:
```
B[:,0]
```

[ ]:
```
B[:,2]
```

[ ]:
```
C = np.array([[2,4,5],[-1,2,3],[1,1,1]])
```

[ ]:
```
C
```

[ ]:
```
B*C
```

[ ]:
```
B
```

[ ]:
```
C
```

1.6 Linear Algebra Operations

[ ]:
```
B
```

[ ]:
```
B.shape
```

[ ]:
```
# determinant
np.linalg.det(B)
```

[ ]:
```
# matrix inverse
np.linalg.inv(B)
```

[ ]:
```
D = np.linalg.inv(B)
```

[ ]:
```
# matrix product
B @ D
```

[ ]:
```
A = np.array([3,4,5])
B = np.array([4,5,6])
```

[ ]:
```
np.dot(A,B)
```

[ ]:
```
B = np.array([[1,-1,1],[2,5,7],[-5,1,2]])
```

[ ]:
```
np.dot(B,D)
```

[ ]:
```
B
```

[ ]:
```
B[:,0]=B[:,0]-3*B[:,1]
```

[ ]:
```
B
```

[ ]:
```
B[0,:] = B[0,:]-5*B[2,:]
```

[ ]:
```
B
```

1.7 some special arrays

[ ]:
```
X = np.zeros(shape=(2,2))
```

[ ]:
```
X
```

[ ]:
```
X = np.ones(shape=(3,3))
```

[ ]:
```
X
```

[ ]:
```
X = np.ones(5)
```

[ ]:
```
X
```

[ ]:
```
X = np.diag([2,3,4,5])
```

[ ]:
```
X
```

[ ]:
```
X = np.arange(0,10,.2)
```

[ ]:
```
X
```

[ ]:
```
X.shape
```

[ ]:
```
X = np.linspace(0,10,50)
```

[ ]:
```
X = np.linspace(0,9.8,50)
```

[ ]:
```
X
```

[ ]:
```
X = np.linspace(0,10,50)
```

[ ]:
```
X**2
```

[ ]:
```
Y = X**2
```

[ ]:
```
Y
```

[ ]:
```
X
```

[ ]:
```
np.sin(X)
```

[ ]:
```
np.log(X)
```

1.8 Basic plotting using Bokeh

[ ]:
```
f = figure(title="Plotting Demo")
```

[ ]:
```
f
```

1.8.1 Scatter Plot

[ ]:
```
x = np.random.uniform(0,1,size=(20,2))
```

[ ]:
```
x
```

[ ]:
```
# x[:,0] are the x values
# x[:,1] are the y values
f.scatter(x=x[:,0],y=x[:,1])
```

[ ]:
```
show(f)
```

[ ]:
```
function_plot = figure(title="Plot of a function")
```

[ ]:
```
x = np.linspace(-3,3,100)
```

[ ]:
```
y = x**2
```

[ ]:
```
function_plot.line(x=x,y=y)
```

[ ]:
```
show(function_plot)
```

[ ]:
```
function_plot.scatter(x=x,y=y,color='red',size=10)
```

[ ]:
```
show(function_plot)
```

[ ]:
```

```