lec1_step1

November 30, 2022

```
[15]: ## Python basics for novice data scientists, supported by Wagatsuma Lab@Kyutech
      #
      # The MIT License (MIT): Copyright (c) 2020 Hiroaki Wagatsuma and Wagatsuma
       ↪Lab@Kyutech
      #
      # Permission is hereby granted, free of charge, to any person obtaining a copy
       ↪of this software and associated documentation files (the "Software"), to
       ↪deal in the Software without restriction, including without limitation the
       ↪rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
       ↪sell copies of the Software, and to permit persons to whom the Software is
       ↪furnished to do so, subject to the following conditions:
      # The above copyright notice and this permission notice shall be included in
       ↪all copies or substantial portions of the Software.
      # THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
       ↪IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
       ↪FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
       ↪AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
       ↪LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
       ↪FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
       ↪IN THE SOFTWARE. */
      #
      # # @Time    : 2022-8-10
      # # @Author  : Hiroaki Wagatsuma
      # # @Site    : https://github.com/hirowgit/2A1_python_intermediate_course
      # # @IDE     : Python 3.9.13 (main, Aug  7 2022, 01:33:23)  [Clang 13.1.6
       ↪(clang-1316.0.21.2.5)] on darwin
      # # @File    : lec1_step1.py
```

```
[7]: import numpy as np
     inData=np.array([1, 2, 3])
     NofD=len(inData)
     flagD=np.full(NofD,True)
     print(flagD)
```

```
[ True  True  True]
```

```
[6]:  inData=np.array([1, 2, 3])
      flagD=np.full(len(inData),True)
      print(flagD)
```

```
[ True   True   True]
```

```
[36]: rdat_s=np.floor(np.random.rand(NofD)*NofD)
      rdat_s=np.asarray(rdat_s, dtype = int)
      print(rdat_s)
```

```
[4 5 6 9 2 9 5 8 6 3]
```

```
[35]: NofD=10
      rdat_s=np.random.randint(NofD, size=NofD)
      print(rdat_s)
```

```
[5 8 8 6 7 0 4 3 1 8]
```

```
[122]: inData = np.linspace(0, 10, NofD, endpoint=True, dtype=int)
       print(inData)
       inData = np.linspace(0, 10, NofD, endpoint=False, dtype=int)
       print(inData)
       # array range →a_range→ arange
       inData = np.arange(0, 100, 10, dtype=int)
       print(inData)
       print(inData[1])
       print(inData[[1]])
       print(inData[[1,2,3]])
       print(inData[rdat_s])
```

```
[ 0  1  2  3  4  5  6  7  8 10]
[0 1 2 3 4 5 6 7 8 9]
[ 0 10 20 30 40 50 60 70 80 90]
10
[10]
[10 20 30]
[40 50 60 90 20 90 50 80 60 30]
```

```
[123]: inData[1]
```

```
[123]: 10
```

```
[124]: inData[[1]]
```

```
[124]: array([10])
```

```
[125]: inData[[1]]+inData[[2]]
```

2

```
[125]: array([30])
```

```
[126]: inData[1]+inData[2]
```

```
[126]: 30
```

```
[130]: inData[[1,2]]+inData[[2,3]]
```

```
[130]: array([30, 50])
```

```
[128]: inData[1]+inData[[2,3]]
```

```
[128]: array([30, 40])
```

```
[129]: inData[[1]]+inData[[2,3]]
```

```
[129]: array([30, 40])
```

```
[131]: inData[[1]]*inData[[2,3]]
```

```
[131]: array([200, 300])
```

```
[90]: r_a1 = np.random.rand()
      print(r_a1)
      r_a2 = np.random.rand(3)
      print(r_a2)
      r_a3 = np.random.rand(1, 4)
      print(r_a3)
      print(' ')
      r_b1 = np.random.randint(1, 5, size=3)
      print(r_b1)
      r_b2 = np.random.randint(1, 5, size=(1,3))
      print(r_b2)
      r_b3 = np.random.randint(1, 5, size=(2,3))
      print(r_b3)
      r_b4 = np.random.randint(1, 5, size=(1,2,3))
      print(r_b4)
```

```
0.5839855423349606
[0.0797405  0.82345808 0.71464285]
[[0.77374123 0.78811937 0.41242111 0.58940351]]

[2 4 1]
[[2 4 1]]
[[2 1 1]
 [2 4 4]]
[[[1 3 1]
  [3 3 4]]]
```

```python
[99]: c1=np.zeros(6, dtype=int)
      print(c1)
      c2=c1.reshape(2,3)
      print(c2)
      c2=c1.reshape(1,6)
      print(c2)
```

```
[0 0 0 0 0 0]
[[0 0 0]
 [0 0 0]]
[[0 0 0 0 0 0]]
```

```python
[137]: inData = np.arange(0, 10, 1, dtype=int)
       print(inData)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```python
[138]: inData[3:]
```

```
[138]: array([3, 4, 5, 6, 7, 8, 9])
```

```python
[134]: inData[-1]
```

```
[134]: 9
```

```python
[136]: inData[3:-1]
```

```
[136]: array([3, 4, 5, 6, 7, 8])
```

```python
[141]: inData[-3:-1]
```

```
[141]: array([7, 8])
```

```python
[ ]:
```