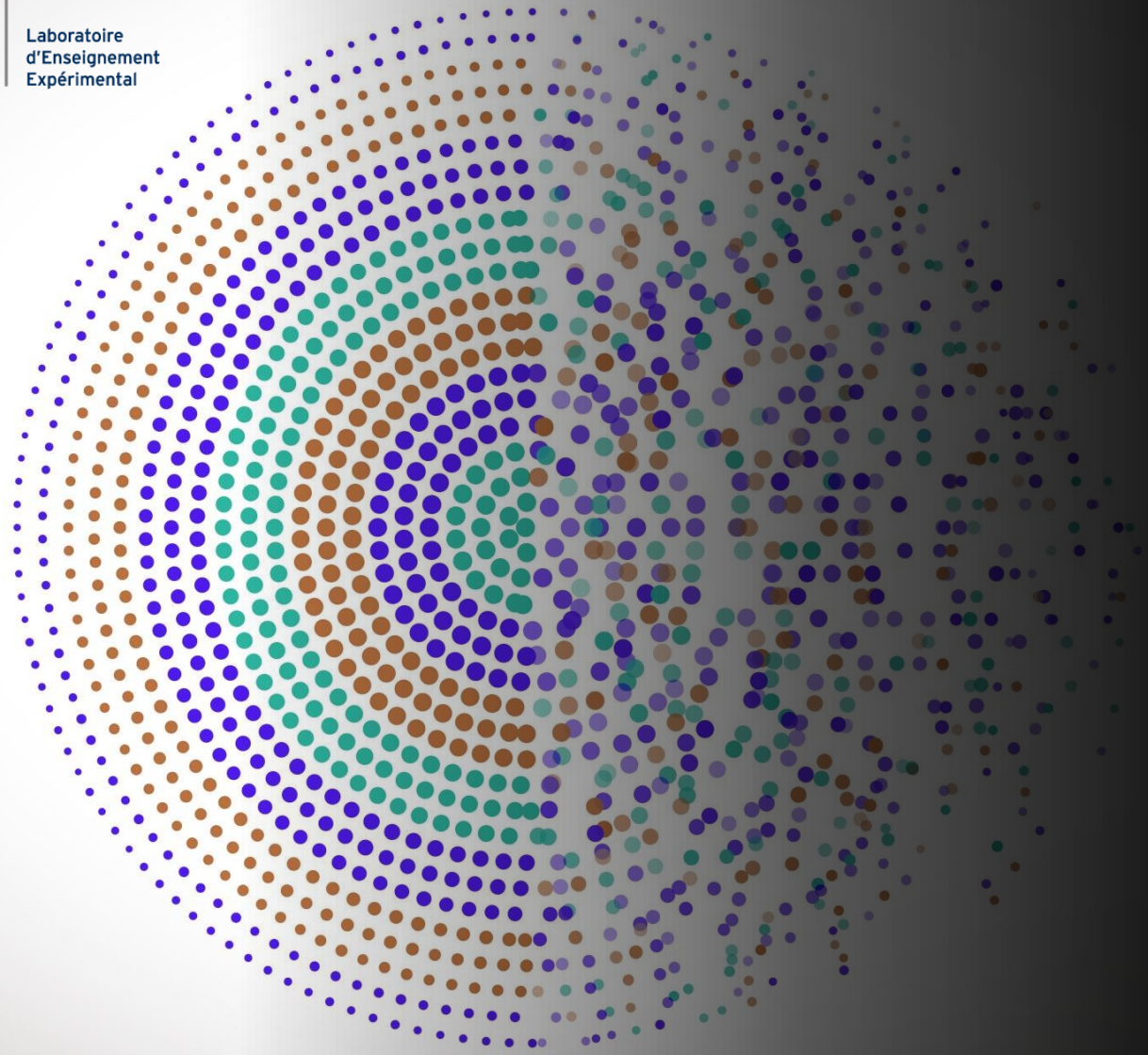
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental



Python / Numpy

Array

Digital Methods
Institut d'Optique / Notions

How to create an array ?

For scientific purpose

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

• Bad method (using List)

```
N = 10
vect = []
for i in range(N):
    vect.append(0)
```

```
print(type(vect))
```

```
<class 'list'>
```

Temps exécution :
N=10 ~1 us
N=1000 ~100 us

• Good method (using Array)

```
import numpy as np
N = 10
vect = np.zeros(N)
```

```
print(type(vect))
```

```
<class 'numpy.ndarray'>
```

Temps exécution :
N=10 ~0,25 us
N=1000 ~0,6 us

Exécution : Core(TM) i7-9750H CPU @ 2.60GHz / 32 Go RAM

How to create an array ?

For scientific purpose

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

- **1D array**
  - Ones or zeros array

```
import numpy as np
N = 10
vect0 = np.zeros(N)
vect1 = np.ones(N)
```

  - Linear/Logarithmic distribution

```
vect_lin1 = np.linspace(start, stop, nb)
```

```
vect_log = np.logspace(dec_start,
dec_stop, nb)
```

```
vect_lin2 = np.arange(start, stop, step)
```

- **2D array**
  - Ones or zeros array

```
import numpy as np
N, M = 10, 20
vect = np.zeros((N,M))
```

  - Meshgrid

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
XX, YY = np.meshgrid(x, y)
```

How to create an array ?

For scientific purpose

- **Random array**
  - Uniform distribution

```
import numpy as np
data = np.random.rand(1000,5)
print(data.shape)
```

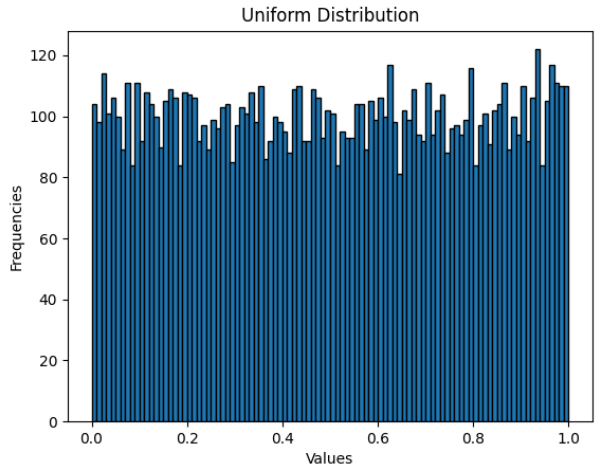
(1000, 5)

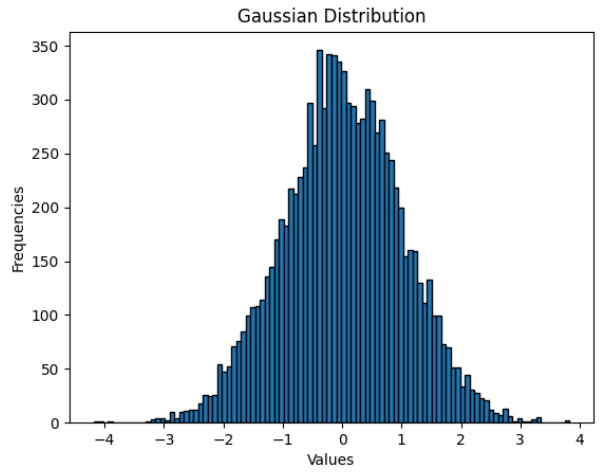
  - Integer (uniform distribution)

```
array = np.random.randint(1, 100,
size=(5, 5))
```

  - Gaussian/normal distribution

```
data = np.random.randn(1000,5)
```





How to fill an array ?

For scientific purpose

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

```
data = np.random.randn(100,5)
```

• Bad method (using List)

```
v1 = []
v2 = []
for i in range(len(data)):
    v1.append(data[i, 1])
    v2.append(data[i, 2])
```

or

```
v1_b = [data[i, 1] for i in range(len(data))]
```

Temps exécution :
~400 us

• Good method (using Array)

```
v1 = data(:, 1)
v2 = data(:, 2)
```

An array is not a simple list !

Temps exécution :
~0,5 us

Exécution : Core(TM) i7-9750H CPU @ 2.60GHz / 32 Go RAM

Conditionals on array

For scientific purpose

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

```
data = np.random.randint(0, 10, size=(10))
```

- Finding elements

```
k = ((data >= 4) & (data < 8))
print(data)
print(k)
```

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

```
[ True False False False  True  True
True False False False]
```

```
data2 = k * data
```

```
[6 0 0 0 7 4 6 0 0 0]
```

- Filtering elements

```
arr = np.array([1, 2, 3, 4, 5, 6])
filtered_arr = arr[arr > 3]
print(filtered_arr)
```

```
[4 5 6]
```