Numpy Fundamentals

- **Date:** November 14, 2023
- **Author:** Engr Muhammad Saqlain
- **Institute:** Department of Computer System Engineering, IUB

```
# import numpy
import numpy as np
```

Universal Functions

A Universal Function or ufunc is a function that performs element-wise operations on data in arrays.

Inplace Operation

```
arr = np.random.randn(1, 8)
arr

np.sqrt(arr)

arr

#use inplace to replace the original arr with output
np.sqrt(arr,arr)

arr

a = np.arange(0.1,6,1)
a

np.exp(a)

a

np.exp(a,a)

a

np.abs(-0.2)

# absolute of floating values
np.fabs(-0.2)

np.fabs(3+4j)

np.abs(3+4j)

np.square(2)
```

```
np.log(10)

np.log10(10)

np.log2(10)

# above...
np.log10(10)/np.log10(2)

np.log1p(9)

np.sign(-0.2)

np.sign(0)

np.sign(5)

np.ceil(0.95)

np.floor(0.95)

np.round(1.5)

np.reciprocal(0.2)

# separate fractional and integer parts
np.modf(0.6)

np.modf(np.arange(0.4,8,0.5))

arr = 5*np.random.randn(6)
arr

remainder, whole_part = np.modf(arr)
remainder

whole_part

b = np.sqrt(np.random.randn(1,6))
b

np.isnan(b)
```

Trigonometric functions

```
x = np.array([1,0])
x

y = np.array([0,1])
y

np.arctan2(y,x)

np.arctan(1)
```

```
np.pi/4

# find hypotenuse of right angle triangle
np.hypot(3,4)

np.sin(np.pi/3)

(np.cos(0.2))**2 + (np.sin(0.2))**2

x = np.random.randn(1,6)
y = np.random.randn(1,6)

x

y

np.add(x,y)

np.subtract(y,2*x)

np.multiply(y, x)

np.divide(x,y)

np.power(x,2)

np.maximum(x,y)

x

y

np.minimum(x,y)

np.sqrt(x)

np.sqrt(y)

np.maximum(np.sqrt(x), np.sqrt(y))

# ignoring nan
np.fmax(np.sqrt(x), np.sqrt(y))

0.1 + 0.2 == 0.3

np.allclose(0.1+0.2, 0.3)

0.1+0.2
```

Inplace arithmetic operators

```
x = np.arange(1,10)
x
```

```
y= np.arange(11,20)
y

x+=y
x

x-=y
x

x*=y

x

x//=y
x

x**=y
x

y
```

Mathematical and statistical methods

```
arr = np.random.randn(5,4)
arr

arr.mean()

arr.mean(axis=0) # column wise mean

arr.mean(axis=1) # row wise mean

arr.sum()

arr.sum(axis=0) # column sum

arr.sum(axis=1) # row sum

a = np.array([[5,2,3], [1,5,1], [9,2,3]])
a

a.prod()

a.prod(axis=0)

a.prod(axis=1)

a.min()

a.max()

a.min(axis=0)

a.max(axis=1)
```

```
a
```

```
a.argmin()
```

```
a.argmin(axis=0)
```

```
a.argmin(axis=1)
```

```
a.argmax()
```

```
a.argmax(axis=0)
```

```
a.std()
```

```
a.std(axis=0)
```

```
a.std(axis=1)
```

```
a.var()
```

```
a.var(axis=0)
```

```
a
```

Methods for boolean arrays

```
arr = np.random.randn(2,5)
arr
```

```
arr>0
```

```
(arr>0).sum()
```

```
((arr>0) & (arr<1)).sum()
```

```
(arr==0).any() # is any element 0?
```

```
np.any(arr>1)
```

```
arr
```

```
np.any(arr>2)
```

```
bool = np.array([True, True, True, False])
bool
```

```
bool.all()  # Are all true?
```

Conditional Functions

```
x = np.linspace(-5,5,11)
x
```

```
np.where(x<0, x**2, x**3)
```

```
np.select([x<-1, x<2, x>=2],[x, x**2, np.sqrt(x)])
```