

بنام خدا

موضوع پروژه: کتابخانه numpy در پایتون

کتابخانه numpy به چه معنی و کاربرد آن چیست؟

کتابخانه NumPy یکی از کتابخانه‌های پرکاربرد و قدرتمند در زمینه محاسبات علمی و عملیات ریاضی در زبان برنامه‌نویسی پایتون است. این کتابخانه به عنوان پایه‌ای برای بسیاری از کتابخانه‌ها و ابزارهای علمی در پایتون مانند SciPy، Pandas و Matplotlib استفاده می‌شود.

کتابخانه numpy از اهمیت زیادی در پایتون برخوردار است و لازم است کسی که برای برنامه نویسی از پایتون استفاده میکند این کتابخانه را بصورت کاربردی کاملاً بتواند استفاده کند.

کتابخانه numpy ویژگی‌های زیادی دارد که میتوان گفت اصلی ترین ویژگی آن آرایه‌های چند بعدی است که قادر به ذخیره و مدیریت داده‌های عددی در ابعاد مختلف هستند و سرعت و کارایی بالایی در انجام عملیات ریاضی و آماری روی داده‌ها دارند. همچنین NumPy امکاناتی را برای مدیریت حافظه و نیز ایجاد آرایه‌های بزرگ و پیچیده فراهم می‌کند.

در این کتابخانه محاسبات علمی صورت می‌گیرد که انواع آن شامل انواع روش‌های ساختن آرایه‌ها در NumPy، آرایه‌های مقادیر ثابت، آرایه‌ها با مقادیر افزایشی و کاهشی، آرایه‌ها با مقادیر لگاریتمی، آرایه‌های مقداردهی نشده، آرایه‌ها با مشخصاتی از دیگر آرایه‌ها، Matrix Arrays می‌شود.

ما برای نصب کتابخانه numpy در پایتون در بخش cmd سیستم گرامر زیر را وارد میکنیم و بعد کلید enter را می‌زنیم:

Pip install numpy

و برای شروع کد نویسی در python از گرامر زیر استفاده میکنیم:

Import numpy as np

Import (فراخوانی)...numpy (کتابخانه مورد استفاده) as np یک اسم

آرایه ها در numpy

آرایه ها در numpy مانند لیست ها در پایتون هستند با تفاوت کمی و امکان کار کردن با اعداد را بطور گسترده برای ما فراهم میکند.

Numpy برگرفته از دو کلمه numeric python است پس برای همین اگر بخواهیم در این محیط کار کنیم باید داده ها را تبدیل به آرایه کنیم.

برای ایجاد یک آرایه ساده در numpy با استفاده از لیست پایتون از گرامر زیر استفاده میکنیم:

```
My pythonlist = [2,5,8,1]
```

برای اینکه بتوانیم این لیست ایجاد شده در پایتون را بصورت آرایه در numpy در بیاوریم از قطعه کد زیر استفاده میکنیم:

```
Import numpy as np
```

```
Numpy_array_from_list = np.array(my pythonlist)
```

```
Numpy_array_from_list
```

```
array([2,5,8,1])
```

محتوای چاپ شده به این شکل است<<<

ما برای اینکه بخواهیم به این کار سهولت و زمان کمتری اختصاص دهیم میتوانیم از تعریف لیست پایتون بصورت مجزا خودداری کنیم و با یک خط کد آرایه را بسازیم. گرامر به شکل زیر است:

```
X = np.array([2,5,8,1])
```

اعمال عملیات ریاضی روی آرایه ها

همانطور که در ریاضی با عملگر ها آشنا شدیم میدانیم عملگرها به چهار دسته اصلی و ابتدایی ریاضی جمع، تفریق، ضرب و تقسیم (+, -, *, /) هستند.

اگر بخواهیم از عملگرها در آرایه استفاده کنیم دستورکار به این میباشد: استفاده از نام آرایه <عملگر مورد نظر> عملوند . گرامر آن به شکل زیر است:

```
Numpy_array_from_list + 2
```

```
array([4,7,10,3])
```

قطعه کد چاپ شده با این شکل نمایش داده میشود<<<

ما در این کدی که نوشتیم عدد 2 را در تمام عنصر های آرایه جمع کردیم.

شکل ارایه ها

انواع ارایه ها عبارت اند از : ارایه یک بعدی . ارایه دوبعدی و ارایه سه بعدی:

ارایه یک بعدی : همان ارایه ساده است که در بالا نمونه ان را مشاهده میکنیم.

ارایه دوبعدی: ما با استفاده از یک کارکتر کاما (",") بین داده هایی در هم در براکت باز و بسته [] و هم در پرانتز باز و بسته () قرار میدهم مثل گرامر زیر:

```
X=np.array([(2,4,6),(1,3,5)])
```

```
Print (x.shape)
```

کد ساخته شده برابر با مقدار روبه رو است<<< (2,3)

ارایه سه بعدی: دقیقاً مثل ارایه دوبعدی نوشته میشود با این تفاوت که دوباره یک کاما و براکت باز و بسته اضافه میشود

```
X=np.array([ [2,4,6],[1,3,5],[7,8,9] ])
```

```
Print(x.shape)
```

کد تولید شده به شکل روبه رو است<<< (2,3)

اتصال ارایه ها در numpy:

Np.vstack: ارایه بصورت عمودی به هم متصل میشوند

Np.hstack: ارایه ها بصورت افقی به هم متصل میشوند

Np.column_stack: اتصال ارایه یک بعدی به صورت یه ستون به سطر ارایه دو بعدی.

Np.concatenate: اتصال ارایه ها به یکدیگر در یک محور مشخص

گرامر اتصال ارایه ها به شکل زیر میباشد:

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

```
Y=np.array( [ [6,7],[5,8] ] )
```

```
Print("VSTACK:",np.vstack((x,y)) )
```

```
Print("HSTACK:",np.hstack((x,y)))
```

```
Print("COLUMN_STACK:",np.column_stack((x,y)))
```

```
Print("CONCATENATE:",np.concatenate((x,y)1))
```

حاصل قطعه کد رو به رو << داخل قطعه کد در پروژه ارسال شد

تابع `as array` در پایتون

این تابع برای تبدیل داده به آرایه استفاده میشود و ورودی میتواند به شکل تاپل یا لیست باشد. گرامر آن به شکل زیر میباشد:

```
Numpy.asarray(data,dtype=None,order=None)
```

Data: داده ای که قصد داریم آن را به آرایه تبدیل کنیم

Dtype: یک گزینه اختیاری است که اگر توسط کاربر مقدار دهی نشود مقدار آن بصورت پیشفرض از نوع داده ورودی گرفته میشود

Order: یک گزینه اختیاری است و مقدار پیشفرض آن C است و استایل پایه برای سطر هاست و مقدار دیگری که میتواند بگیرد f است

تابع `numpy.arange` در پایتون

این تابع در `numpy` فاصله بین اعداد را برمیگرداند و نتیجه آن بازه ای از عددها میباشد گرامر این تابع به شکل زیر است:

```
Numpy.arange(start,stop,step)
```

Start: نقطه شروع بازه.

Stop: نقطه اتمام بازه که خود عدد شامل بازه نمیشود

Step: فاصله بین اعداد که در بازه بصورت پیشفرض یک است

مثال:

```
Import numpy as np
```

```
Np.arange(5,20,2)
```

کد تولید شده <<< `array(7,9,11,13,15,17,19)`

<https://7learn.com/blog/complete-numpy-library-tutorial>