



Real Python

دوره جامع پایتون در یادگیری ماشین

قسمت سوم، عملیات ریاضی بر روی آرایه ها

Numpy3!

DataTalk.ir

Created by : Ali Arabshahi

Contact us : [Linkedin.com/in/mrAliArabshahi](https://www.linkedin.com/in/mrAliArabshahi)

انجام عملیات ریاضی در نامپای

محاسبات دستی بر روی آرایه ها

ما به راحتی می توانیم بر روی آرایه هایی که توسط کتابخانه نامپای درستشون کردیم، عملیات ریاضی انجام بدیم و اگه یادتون باشه گفته بودیم که مهم ترین خوبی آرایه هم اینه که می توانیم عملیات های مختلف رو به راحتی و با یک دستور بر روی همه عناصر اون آرایه پیاده سازی کنیم. 😊

بیاین ببینیم داستان از چه قراره؛ اما قبل از هر چیز، طبق معمول کتابخانه پایتون رو فراخونی می کنیم و یک آرایه می سازیم

```
import numpy as np
arr = np.arange(0,10)
```

حالا میایم و بر روی این آرایه عملیات های مختلف ریاضی رو انجام می دیم

```
arr + arr
```

```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
arr * arr
```

```
array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81])
```

```
arr - arr
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

در حالت تقسیم پایتون به هشدار کوچک بهمون می‌ده، نگران نباشین چون هنوز بهمون ارور نداده! دلیل این هشدارم اینه که در ریاضی صفر تقسیم بر صفر معنی نداره و در عملیاتمون به چنین حالتی برخوردیم بنابراین در اون مورد به خصوص به ما خروجی نال یا بی معنی داده 🤖

```
arr/arr
```

```
array([nan, 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

و این حالت در مواقعی که تقسیم 1 بر 0 اتفاق می‌افته هم پدیدار میشه

```
1/arr
```

```
array([ inf, 1., 0.5, 0.33333333, 0.25, 0.2, 0.16666667, 0.14285714, 0.125, 0.11111111])
```

حتما می‌دونین که دو تا ستاره در پایتون به معنی عملیات ریاضی توان محسوب میشه

```
arr**3
```

```
array([ 0, 1, 8, 27, 64, 125, 216, 343, 512, 729], dtype=int32)
```

اعمال توابع از پیش تعریف شده بر روی آرایه ها

نامپای درون خودش کلی تابع از پیش تعریف شده داره که می‌تونیم از اون ها هم برای تسریع در عملیات ریاضی مون بهره ببریم. یعنی نیاز نیست بشینیم و همه چی رو از صفر تعریف کنیم 😊

فقط با فراخونی کتابخونه نامپای، به کلی توابع خفن دسترسی پیدا می‌کنیم. ما در اینجا به چند تا از پرکاربردترین اون توابع اشاره می‌کنیم اما برای مشاهده توابع بیشتر می‌تونین در گوگل سرچ کنین، و یا در اینجا کلیک کنین:

اینجا👉

np.sqrt(arr)

```
array([0.          , 1.          , 1.41421356, 1.73205081, 2.          ,  
       2.23606798, 2.44948974, 2.64575131, 2.82842712, 3.          ])
```

np.exp(arr)

```
array([1.00000000e+00, 2.71828183e+00, 7.38905610e+00, 2.00855369e+01,  
       5.45981500e+01, 1.48413159e+02, 4.03428793e+02, 1.09663316e+03,  
       2.98095799e+03, 8.10308393e+03])
```

np.max(arr)

9



np.sin(arr)

```
array([ 0.          ,  0.84147098,  0.90929743,  0.14112001, -0.7568025 ,  
       -0.95892427, -0.2794155 ,  0.6569866 ,  0.98935825,  0.41211849])
```

np.log(arr)

```
array([-inf, 0.          ,  0.69314718,  1.09861229,  1.38629436,  
       1.60943791,  1.79175947,  1.94591015,  2.07944154,  2.19722458])
```

و خیلی توابع بیشتر اما روال کار همیشه همینه، نامپای رو فراخونی می کنیم و بعدش اسم اون تابع و درون اون تابع هم آرایه ای که می خوام تغییر ها روشن اعمال شه رو صدا می زنیم

همین! تا جلسه بعد ، فعلا

