#Python -da nəyi niyə öyrənim part 9

#Module anlayışına pythonda spesifik yanaşma – burada kiçik nüanslara toxunacağıq

Belə ki, biz modulları daxil edərək birbaşa işləyirik çox vaxt və əsas yerinə yetirdiyimiz proseslərə fokuslanırıq.

Lakin, zaman keçdikcə professional və ya geniş proektlərdə bizim hər şeyi yadda saxlamaq kimi bacarığımız olmadığı kimi, bizim komanda yoldaşlarımız ilə iş fəaliyyətində bölgüdə biz class – sinifləri yazdıqca metodlar dövrlər deyərkən artıq hər bir şeyi bir .py sonluqlu faylda saxlasaq düzgün ifadə edərsək, çox səmərəsiz və keyfiyyətsiz oxunaqsız bir iş görmüş oluruq.Buna görə də yazdığımız obyekt – class -ları eləcə də funksiyaları xarakterik olaraq adlandırılmış .py sonluqlu fayllara yığırıq. Bunlar da modullar olur. Gəlin anlayışı başa düşdükdən sonra öz modullarımızı və 3 -cü tərəflər (özümüz tərəfimizdən yazılmamış və Python Foundation tərəfindən yaradılmamış) modulları necə səmərəli istifadə edərik ona nəzər yetirək. ( Numpy modulunu burada nümunə üçün işlədəcəm). Bir list şəklində matrisimiz var:

my\_matris\_one = [[4, 8, 9],

[7, 12, 24],

[16, 3, 5]]

Bunu numpy modulunun array metodu ilə massivə çevirmək üçün modulumuzu 3 yolla import edə bilərik:

1. import numpy
2. from numpy import \*
3. from numpy import array

Bu 3 üsuldan mənim şəxsi qənaətim və öyrəndiklərim üzəridən deyə bilərəm ki, 3 cü üsul daha səmərəlidir. Belə ki, 1 ci və 2-ci üsulda bütün modul mövcud çalışmaya daxil edilir və bu #executetime -yi artırır və #performance -ni aşağı salır. Kiçik əməliyyatlarda hiss edilməyən bu fərqlər #data ilə işlərkən, #machinelearning #deeplearning yarışmalarında, eləcə də orta və irimiqyaslı proektlərdə sizin işlərinizin sürətində böyük fərqlər yarada bilər. 3-cü üsulun kiçik bir çaşqınlıq yarada biləcək xüsusiyyəti onu (array metodunu) artıq modul adsız çağırdığımız üçün dəyişən təyin edərkən bunu nəzərə almalıyıq. (Part 1 – də clean code haqqında mənbələr var.)