Bugun darsligimiz NumPy kutubxonasi qo’shimcha funksiyalari haqida:

O'z ufuncingizni qanday yaratishingiz mumkin:

O'zingizning ufuncingizni yaratish uchun siz Python'dagi oddiy funktsiyalar kabi funktsiyani belgilashingiz kerak, keyin uni usul bilan NumPy ufunc kutubxonangizga qo'shasiz **frompyfunc().**

Usul frompyfunc() quyidagi argumentlarni oladi:

1.function- funksiya nomi.

2.inputs- kiritilgan argumentlar (massivlar) soni.

3.outputs- chiqish massivlari soni.

Ufunci bizga qay xolatlarda kerak bo’ladi. Misol uchun sizda aytaylik 100 dan ortiq massiv yoki element mavjud sizga atigina ulardan bir qismi kerak. Sizga shunday hollarda ufunc yordam beradi. **Kod\_1** Biz bbiz olgan ufuncni kodimiz davomida topa olmagan holda bizga **type()** funcsiyasi yordam beradi. **Kod\_2**  bu yerda **print(type(np.add))** bizga ufunc qilib beraolishi bildirmoqda.

Agar u ufunc bo'lmasa, u ikki yoki undan ortiq massivlarni birlashtirish uchun o'rnatilgan NumPy funksiyasi kabi boshqa turni qaytaradi: **Kod\_3 print(type(np.concatenate))** bu metod o’zbek tilida aytilganda yana ham aniq qilish degan manoni bermoqda. concatenate so’zining manosi birlashtirmoq degan manoni beradi. Bu kodimizda ufunc ichida massiv yani biron bir amalni bajaruvchi funksiya bo’lganligi sababli bizga bizga **function** javobini qaytarmoqda.Bu usul bazi hollarda biron bir kutubxona shu nomli kutubxona bor yoki yoq ekanligini tekshirishda qo’llaniladi. Agar funktsiya umuman tan olinmasa yani NumPy ichida bu nomli funksiya mavjud bo’lmasa , u xatoni qaytaradi: **Kod\_4**  NumPy ichida abcdef nomli funksiya yuq shu sababdan bu kodimiz xatolikga duch keldi.

Funktsiya ufunc ekanligini yoki yo'qligini tekshirish uchun if iborasidan foydalaning:**Kod\_5**Biz bu usuldan 28 dars-

ligimizda foydalangan edik yani tushunarsiz xatoliklar olish emas o’zimizga tushunarli bo’lgan javobni olish uchun ufunc ichida arifmetik amallar ham bajarish mumkin ya’ni ikki massiv o’rtasida qo’shish, ko’paytirish,bo’lish,ayirish amallarini qo’llashimiz mumkin. **Eslatma: bu usul faqat berilgan ikki yoki undan kop massivlarda bajariladi aks holda qatolikga olib keladi, hamda berilgan massivlar ichidagi elementlar soni o’zaro teng bo’lishi kerak.**

**Add()** bu ingliz tilida tarjima qilinganda qo’shish degan ma’noni beradi hamda bu metod massivlarichidaelementlar soni teng bo’lsagina ishlaydi: **Kod\_6**

**Subtract()** bu ingliz tilida tarjima qilinganda ayirish degan ma’noni anglatadi hamda massivlar ichida elementlar soni teng bo’lsagina ishlaydi: **Kod\_7**

**Multiplication()**  bu so’z ingliz tilida tarjima qilinganda ko’paytirish degan ma’noni anglatadi hamda massivlar ichida elementlar soni o’zaro teng bo’lsagina ishlaydi : **Kod\_8**

**Division()** bu so’z ingliz tilida tarjima qilinganda bo’lish degan ma’noni anglatadi hamda massivlar ichida elementlar soni o’zaro teng bo’lsagina ishlaydi **: Kod\_9**

**Power()** bu so’z ingliz tilidatarjima qilingandaquvvat degan ma’noni anglatadi vazifasi esa birinchi massivdan ikkinchi massiv qiymatlari kuchiga ko’taradi hamda umumiy javobni yangi massivga yuklaydi: **Kod\_10**

**Mod()** bu so’z ingliz tilida tarjimasi bazi manbalarda topilmasligi mabjud ammo manosi qolgan degan ma’noni anglatadi hamda buni massivlar o’rtasida bo’ilsh amalida qolgan qoldiqni olishda ishlatishimiz mumkin: **Kod\_11**

**Remainder()** bu metod ham mod bilan qo’llaniladi yani ikki usul ham bir javobni qayd etadi: **Kod\_12**