

## Python modul darsligi: RegEx kutubxonasi.

RegEx yoki Regular Expression(eng: muntazam ifoda) - qidiruv sxemasini tashkil etuvchi belgilar ketma-ketligi. RegEx-dan satrda belgilangan qidiruv namunasi mavjudligini tekshirish uchun foydalanish mumkin. Bunga misol qilin **Kod\_1** ko'rsatishimiz mumkin. **Kod\_2** da siz kiritgan matn ichidan python sozi bor yoki yo'qligini qidiradi. Bu kutubxonaga misol qilib qidiuv tizimlarini aytishimiz mumkin ekan. Bular misol esa Google, Amazon,Yandex , FixeFox,DocDox lar bo'ladi, ha aytgancha Python dasturlash tili Suniy intellektda keng ko'lamda foydalanilgani sababli shu va shunga o'xshash kutubxonalardan keng ko'lamda foydalanadi. Bu bilan biz OpenAI kompaniyasining CHatGPT 3,5 hamda 4- versiyalarini aytaolamiz. BUndan tashqari Midjourney,Scribbleotar,Tersonflow, Necroset larni ham aytib otishimiz mumkin ekan.

### PIP nima?

PIP ("Pip Installs Packages" ya'ni pip o'rnatish paketlari deb tarjima qilinadi ) Python dasturlash tilida ishlatiladigan, Python paketlarini boshqarish va o'rnatish uchun ishlatiladigan standart kutubxonadir. PIP orqali Python paketlarini o'rnatishingiz, yangilashingiz, o'chirishingiz, qidirishingiz va shunga o'xshash vazifalarni bajarishingiz mumkin.

### Paket nima?

Paket modul uchun kerak bo'lgan barcha fayllarni o'z ichiga oladi. Modullar loyihangizga kiritishingiz mumkin bo'lgan Python kod funksiyalar jamlanmasi.

### PIP haqida umumiy tushuncha :

Biz python ni o'rnatish chog'imizda pip ga kamyuterimizdan ruxsat bergan edik! Biz endilikdagi darslarimizda manashu metod bilan ishlashni o'rganamiz.

### PIP versiyasini aniqlash:

Xo'sh bizda pip mavjud ammo qaysi versiada, buni bilish uchun biz terminalga **pip --version** yozish kifoya. PIP ning eng oxirgi versiyasi 23.1.2 hisoblanadi.

### PIP ni yangilash(upgrade)

Sizda kichikroq versiyasi boladigan bo'lsa **python -m pip install --upgrade pip** buyrug'i orqali eng so'ngi versiaga yangilashingiz mumkin.

### PIP orqali packet ornati:

Biz pip ni yangiladik va u eng songi versiyada bizga ishlashga tayyor. Biz pip ni o'zimizga kersakli kutubxona yuklashga ishlatishimiz mumkin ekan. Ya'ni qauyidagi kod orqali istalgan **pypi.org** omboridagi kurubxonalarni o'rnatishingiz mumkin: **pip install packet\_nomi** misol uchun rasm chizishda va graph keng kolamda ishlatilinuvcchi kutubxona bu **Pillow** kutubxonasidur **pip install pillow**

### PIP da mavjud modulni yangilash:

Biz da kutubxona o'rnatilgan ammo bu kutubxonadan foydalanmay qo'ygan edik ! Ya'ni kutubxona yangi versiaga yangilandi albatta biz ham yangilashimiz kerak aks holda biz kutubxondadan foydalanish chog'imizda xatoliklarga uchrashimiz mumkin bo'ladi. Quyidagi buyrug' orqali packetlarni yangilashingiz mumkin: **pip install --upgrade package\_name**

### PIP dan modullarni qidirish:

Biz pip dan foydalanar ekanmiz modular kopayib ketishi turgan gap yani bizda juda kop modular o'rnatilgan. Bunda biz search yani qidiruv funksiyasidan foydalanishimiz mumkin yani **pip search packet\_nomi** Bu funksiya **pypi.org** saytiga API orqali kiraoluvchilar uchun! Siz show funksiyasidan foydalanishingiz mumkin : **pip show pillow** Bizda barcha o'ranitilgan packetlarni korish uchun : **pip list** ni yozish kifoyda.

### PIP da o'rnatilgan modullarni o'chirish:

Siz o'zingiz uchun biron bir modulni o'rnatdingiz va undan foydalanib bo'ldingiz endi u sizga keragi yoq ya'bi o'chirmoqchisiz bunday holatda siz : **pip uninstall packet\_nomi** misol uchun **pip uninistall pillow** hamda y(yes) buyrug'i orqali tasdiqlash kifoya. Keling sizlar bilan ba'zi kodlarni taxlil qilaylik: **Kod\_3** berilgan sozni sher ichida qidiradi **KOD\_4** da esa necha marotaba ishlatilinganligini.

### Pythonda try,except,else,finally tushunchasi:

Xatolarni boshqarish – bu dasturni ishga tushirishda yuzaga keladigan kutilmagan yoki noto'g'ri vaziyatlarni hal qilish jarayoni. Ushbu xatolar dasturning ishlaymay qolishiga yoki noto'g'ri natijalarga olib kelishi mumkin va dastur muammosiz ishlashi va kerakli natijani berishi uchun ularni to'g'ri ishlatish muhimdir.

Python-da **try-except** bayonot xatolarni qayta ishlash uchun ishlatiladi. Blokning asosiy tuzilishi **try-except** quyidagicha: **Kod\_5** Ushbu kodimizda x qiymat yo'qligi sababli except bo'limiga o'tkazib yubordi. Try = ishga tushir yoki sinab kor kabi tarjima qilinadi. Except = bundan mustasno deya tarjima qilinadi. Else = boshqa o'tkan darslarga bunga ta'rif bergan edik. Finally = nihoyat deya tarjima qilinadi. Endi **Kod\_5** ni tarjima qiladigan bo'lsal print(x) ni sinab kor agarda ishga tushgudek bo'lamsa xatolik yuz berdi deya chop et. Agarda biz x ga qiymat bergudek bolsak shunchaki javop chiqadi sabab bu metodlar shu vazifani bajaradi: **Kod\_6** Bundan xulosa shuki **try** metodi xatolik bor yoki yo'qligini tekshirar ekan .

finally kalit so'zi Python tilida "try-except" bloklari bilan birga ishlatiladi. Uning asosiy vazifasi, dasturning barcha holatlarda ham bajarilishi ta'minlanishi va bajarish so'ngida ham yozilgan kodning bajarilishi kerak bo'lgan qismi bo'lib to'g'rilashini ta'minlashdir. **Kod\_7**

finally bloki, try blokida yoki except blokida qandaydir holat yuz berishi bilan aloqador bo'lsa ham, bajarilishi kerak bo'lgan kodni ishga tushiradi.

Quyidagi misollar orqali finally kalit so'zining ishlashini ko'rib chiqamiz: **Kod\_8**