**Dasturchi xato qiladi. Yaxshi dasturchi esa ko'p xato qiladi**

**XATOLAR**

Har qanday dasturchi kod yozishda xato qiladi. Ko'p yozgan odam esa ko'p xato qiladi va bu tabiiy. Ba'zi xatolarimiz Python tomonidan dastur bajarilishdan avvaloq aniqlanadi. Ba'zilari esa dastur bajarish jarayonida aniqlanib, dasturimiz to'xtab qoladi. Keling, bugun dasturlashni yangi boshlaganlar eng ko'p yo'l qo'yadigan xatolar bilan tanishamiz.

**SyntaxError - SINTEKS XATOLIK**

Bu eng ko'p uchraydigan xato bo'lib, odatda dasturlash tili qoidalariga amal qilmaslik natijasida kelib chiqadi. Aksar dasturlash muhitlari sintaks xatolikni dastur bajarilishidan avvaloq aniqlab, dasturchiga ishora beradi. Sintaks xatolik bor dasturni Python bajarmaydi.

**print "Hello World!"**

*Natija: SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Hello World!")?*

Odatda dasturlash muhiti xatoning turi bilan birga (SyntaxError), xato haqida qo'shimcha ma'lumot ham beradi *(Missing parentheses in call to 'print'. Did you mean print("Hello World!")*?). Agar ingliz tilini tushunmasangiz, Google Translate sahifasi yordamida matnni rus yoki o'zbek tiliga tarjima qilib olishingiz mumkin.

Agar rus tilini bilsangiz, xato matnini rus tilga tarjima qiling. O'zbek tilidagi tarjimalar hali biroz tushunarsiz.

**EOL va EOF xatolik**

EOL (End of Line - qator yakuni) xatoligi sintaks xatolikning bir turi bo'lib, odatda qator oxirida qo'shtirnoq (birtirnoq) ni yopish esdan chiqqanda yuzaga keladi.

**print("Hello World!**

*Natija: SyntaxError: EOL while scanning string literal*

EOF (End of function - funktsiya yakuni) xatoligi esa funktsiya oxirida qavsni yopish esdan chiqqanda yuzaga keladi.

print("Hello World!"

Natija: SyntaxError: unexpected EOF while parsing

EOF xatoligining muammoli tarafi, Python aynan qaysi funktsiya yopilmay qolganini ko'rsata olmaydi, ya'ni kodni sinchiklab ko'zdan kechirib chiqish talab qilinadi.

IndentationError - JOY TASHLASHDA XATOLIK

Python tilida qator boshidan yoki joy tashlab yozish muhim ahamiyatga ega. Qator boshidan asossiz joy qoldirish **IndentationError** ga olib keladi.Misol uchun aytaylik qator boshida 1 dona bo'sh joy qolgani uchunoq VS Code muhiti xatolikni aniqlab, qizil bilan belgilab qo'yadi.

Ba'zi joylarda esa aksincha, bo'sh joy tashlab yozish talab qilinadi. Masalan, **for** tsiklida yoki **if-elif-els**e shartlarining ichida va hokazo.

**print("O'ngacha sanaymiz")**

**for n in range(10):**

**print(n+1)**

*Natija: IndentationError: expected an indented block*

**son = 50**

**if son>=0:**

**print("Musbat son")**

**else:**

**print("Manfiy son")**

*Natija: IndentationError: expected an indented block*

**QANCHA JOY TASHLAYMIZ?**

Yuqoridagi misollarda **IndentationError** oldini olish uchun joy tashlash talab qilindi. Xo'sh, qancha joy tashlash kerak va qanday qilib?

Aslida, hech bo'lmaganda 1 harflik bo'sh joy qoldirish ham bizni **IndentationError** dan xalos qiladi. LEKIN, biz dastur davomida bir hil joy tashlashga odatlanishimiz kerak.

Qoida sifatida kamida 4 ta harflik joy yoki 1 ta TAB (klaviaturadagi tab tugmasi) joy tashlashni odat qilishimiz kerak. Va eng muhimi ikkalasini aralashtirmasligimiz lozim. Ya'ni agar siz joy tashlash uchun Space (probel) ishlatsangiz, oxirigacha shunday qiling, agar Tab ishlatsangiz oxirigacha tab ishlating. Ikkalasini aralashtirmang!

**RUN TIME ERROR - DASTURNI BAJARISHDA XATOLIK**

Run time error — dastur bajarish jarayonida kelib chiqadi va dasturning ishlashini to'xtatadi. Sintaks xatolikdan farqli ravishda Python bunday xatolarni dasturni bajarishdan avval aniqlay olmaydi. Run time error ning bir necha turi bor. Keling, ulardan ba'zilari bilan tanishamiz.

**TypeError**

Biror amalni (funktsiya, metod) noto'g'ri ma'lumot turi ustida bajarish.

**son = input("Istalgan son kiriting: ")**

**print(f"{son} ning kvadrati {son\*\*2} ga teng")**

*Natija: TypeError: unsupported operand type(s) for \*\* or pow(): 'str' and 'int'*

Yuqoridagi kodda biz foydalanuvchi kiritgan qiymatni matndan songa o'tkazib olishni unutdik, natijada sonning kavdratini hisoblashda Python xato berdi.

**NameError**

O'zgaruvchi, funktsiya, obyekt nomini noto'g'ri yozish natijasida kelib chiquvchi xatolik.

**prit("Hello World!")**

*Natija: NameError: name 'prit' is not defined*

**mevalar = ['olma','uzum','nok','anor','anjir']**

**for meva in mvealar:**

**print(meva)**

*Natija: NameError: name 'mvealar' is not defined*

*ValueError*

Funktsiyaga noto'g'ri qiymatni yuborish natijasidagi xatolik

**son = int(input("Istalgan son kiriting: "))**

**if son>=0:**

**print("Musbat son")**

**else:**

**print("Manfiy son")**

Yuqoridagi dasturning 1-qatorida foydalanuvchidan istalgan son kiritishni so'rayabmiz, va foydalanuvchi kiritgan qiymatni int ya'ni butun songa o'tkazyabmiz. Kodning o'zida xato yo'q, lekin dastur bajarish jarayonida foydalanuvchi butun son emas, o'nlik son kiritgani uchun **ValueError** xatosi chiqdi. Sababi int() funktsiyasi faqatgina butun sonlar ko'rinishidagi matn bilan ishlaydi.

Dastur xato bermasligi uchun yoki int() funktsiyasini float() ga almashtrishimiz kerak, yoki foydalanuvchidan butun son kiritishni talab qilishimiz kerak.

**IndexError**

Yangi dasturchilar yo'l qo'yadigan yana bir xato bu indeks xatolik. Ya'ni ro'yxat elementlariga murojat qilishda indeksni noto'g'ri kiritish.

**mevalar = ['olma','anor','uzum']**

**print(mevalar[3])**

*Natija: IndexError: list index out of range*

Bizda mevalar degan ro'yxat bor va ro'yxatda uchta meva bor. Biz 3-elementni konsolga chiqarmoqchimiz va **print(mevalar[3])** deb yozdik va **IndexError** natijasini oldik. Sababi, dasturlashda indeks 0 dan boshlanadi va 3-elementga murojat qilish uchun 2-indeksni tanlaymiz. Demak, to'g'ri kod:

**mevalar = ['olma','anor','uzum']**

**print(mevalar[2])**

*Natija: uzum*

**ZeroDivisionError**

Dastur jarayonida 0 ga bo'lish yuzaga kelgandagi xatolik

**x, y = 50, 50**

**z = 250/(x-y)**

Natija: ZeroDivisionError: division by zero

**MANTIQIY XATOLAR**

Mantiqiy xatolar - dasturchi tomonidan yo'l qo'yilgan va kutilgan natijani berishda to'sqinlik qiluvchi xatolar. Bunday xatolar eng ko'p uchraydigan va aniqlash eng qiyin bo'lgan xatolar hisoblanadi. Aksar holatlarda Python mantiqiy xatolarni aniqlamaydi va dastur bajarilaveradi (lekin kutilgan natija chiqmaydi).

Mantiqiy xatolar turli ko'rinishda bo'lishi mumkin, masalan sonlar bilan ishlashda:

**radius = 5**

**pi = 4.14**

**aylana\_yuzi = pi\*radius\*\*2**

**print(aylana\_yuzi)**

*Natija: 103.49999999999999*

Yuqoridagi kod bajarildi, va natija ham chiqdi. Lekin natija xato. Nima uchun?

Yana bir misol ko'raylik:

**son = float(input("Istalgan son kiriting: "))**

**ildiz = son\*\*1/2**

**print(f"{son} ning ildizi {ildiz} ga teng")**

Yuqoridagi natijaga e'tibor bersangiz, 9 sonining ildizi 4.5 deb chiqdi. Sababi, 2-qatorda ildizni hisoblashda foydalanuvchi kiritgan son avval 1-darajaga oshirildi va undan keyin 2 ga bo'lindi. Kodni to'g'rilaymiz:

**son = float(input("Istalgan son kiriting: "))**

**ildiz = son\*\*(1/2)**

**print(f"{son} ning ildizi {ildiz} ga teng")**

Noo'rin bo'sh joy qoldirish (yoki qoldirmaslik) ham mantiqiy xatoga olib kelishi mumkin:

**mevalar = ['olma','uzum','nok','anor','anjir']**

**for meva in mevalar:**

**print(meva)**

**print("Dastur tugadi")**

Yuqorida "Dastur tugadi" matni bir marta, dastur tugaganidan so'ng chiqishi kerak edi. Lekin o'ngga suriib qolgani uchun bir necha bor qaytarildi.

Bundan boshqa ham mantiqiy xatoliklar juda ko'p uchraydi.

Mantiqiy xatoliklar mutlaqo topilmasdan ham qolib ketishi, va dastur bozorga chiqqanidan so'ng aniqlanishi tabiiy hol. Shuning uchun ham aksar dasturlar tez-tez yangilanib turadi.

Dastur jarayonida bundan boshqa xatoliklar ham ko'p uchraydi. Biz ulardan ba'zilari bilan tanishdik xolos. Keyingi darslarimizda Runtime xatoliklarni qanday qilib dastur davomida aniqlash, va dastur to'xtab qolishining oldini olishni o'rganamiz.