

Python da funksiya:

Funksiya ma'lum bir vazifani bajarishga mo'ljallangan kodlar yig'indisi. Biz shu paytgacha bir nechta tayyor funksiyalardan foydalanib keldik. Misol uchun `print()` funksiyasi konsolga matn chiqarish uchun, `range()` funksiyasi esa ma'lum oraliqdagi sonlarni yaratish uchun ishlatiladi.

Aslida har qanday funksiyaning ortida ham bir necha qatordan iborat kod bo'ladi, lekin biz funksiyaga murojat qilganda uning nomini yozamiz xolos. Funksiya ortidagi kod esa biz uchun yashirin bo'lib qolaveradi. Funksiyalarning qulayligi ham shunda. Dastur davomida ma'lum bir kodlarni qayta-qayta yozmaslik uchun biz ularni jamlab, bitta funksiya ichiga joylashimiz va dastur davomida bu kodlarga funksiya nomi orqali murojat qilishimiz mumkin.

Funksiyalar turlicha bo'ladi, ba'zi funksiyalar sizdan qiymat qabul qilib, konsolga biror mat'umotni chiqaradi, ba'zilari esa sizdan qabul qilgan qiymat ustida turli amallar bajarib yangi qiymat qaytaradi. Foydalanuvchidan mutlaqo qiymat qabul qilmaydigan funksiyalar ham mavjud.

Ushbu mavzuda siz qanday qilib Pythonda yangi funksiya yaratish, unga murojat qilish, tekshirish va to'g'rilashni o'rganasiz. Shuningdek darsimiz yakunida dasturimizni bir nechta faullarga ajratishni va funksiyalarani alohida, module deb ataluvchi fayllarga joylashni ham o'rganamiz.

FUNKSIYA YARATAMIZ

Keling oddiy, `salom_ber` deb nomlangan funksiya yaratamiz. Bu funksiya murojat qilganimizda konsolga "Assalom alaykum!" degan xabarni chiqarsin. **Kod_1**

Kodni qatroma-qator tahlil qilaylik:

Avvalo `def` operatori yordamida Pythonga funksiya yaratayotganimizni bildirdik. `def` dan so'ng esa funksiyamizga nom berdik va qavslarni ochib, yopdik. Bizning funksiyamiz foydalanuvchidan hech qanday qiymat qabul qilmaydi, shuning uchun ham qavs ichi bo'sh. Keyingi misollarda foydalanuvchidan qiymat qabul qiluvchi funksiyalarni ham ko'ramiz.

`def` qatoridan keyin o'ngga surib yozilgan har qanday kod funksiyaning badani hisoblanadi. 2-qatorda biz uchta ketma-ket qo'shtirnoq ichida funksiya haqida ma'lumot berdik. Python mana shu ma'lumotni o'qib, dasturchi funksiya haqida bilmoqchi bo'lganda aynan shu matnни ko'rsatadi.

Oxirgi qatorimizda esa "Assalomu alaykum!" matnini konsolga chiqarishni buyurdik. Bizning sodda funksiyamizning asosiy vazifasi ham shu.

Mana funksiya tayyor. Endi bu funksiyadan foydalanish uchun uni chaqiramiz. Buning uchun funksiya nomini yozamiz va qavslarni ochib, yopamiz (esingizda bo'lsa bizning funksiyamiz qiymat qabul qilmaydi, shuning uchun qavslar ichi bo'sh). Funksiyaga nom berishda fe'l, ya'ni harakatni bildiruvchi so'zlar yoki jummalardan foydalaning. Bu bilan siz o'zgaruvchi va funksiya o'rtasini farqlashingiz oson bo'ladi. Misol uchun, yuqorida biz funksiyamizni `salom` emas `salom_ber` deb nomladik.

FUNKSIYAGA QIYMAT UZATISH

Avvalgi sodda funksiyamiz foydalanuvchidan hech qanday qiymat olmaydi va barchaga birday "Assalomu alaykum!" deb javob qiladi. Keling funksiyaga o'zgartirish kiritamiz, funksiya foydalanuvchi ismini qabul qilib, unga ismi bilan murojat qilsin. Buning uchun funksiya nomidan keyin, qavs ichida foydalanuvchi berishi kerak bo'lgan qiymatni ko'rsatamiz. **Kod_2**

Agar funksiyaga murojat qilishda, unga qiymat bermasak xatolik vujudga keladi:

```
salom_ber()
```

Natija: `TypeError: salom_ber() missing 1 required positional argument: 'ism'`

DOCSTRING

Avval aytganimizdek, funksiya yaratganda, funksiya qanday ishlashi haqida qisqacha ma'lumot berib ketish o'zimiz uchun ham, kelajakda bizni funksiyamizni ishlatadigan boshqa dasturchilar uchun ham juda foydali bo'ladi. Quyidagi funksiyaning 2-qatorda biz funksiya haqida ma'lumot berdik. Bu qator docstring deyiladi. Murakkab funksiyalar uchun docstringni bir necha qatarga bo'lib yozishingiz mumkin **kod_3**

Xo'sh, bu ma'lumot qachon va qayerda ko'rsatiladi? Dastur yozish jarayonida funksiya nomini yozishingiz bilan, docstring ko'rsatiladi:

Docstringni konsolga chiqarish uchun `print(funksiya_nomi.doc)` deb ham yozishimiz mumkin: **kod_4**

FUNKSIYAGA BIR NECHA BOR MUROJAT QILISH

Funksiya yaratishning asl maqsadlaridan biri, biz unga qayta-qayta, yangi qiymatlar bilan murojat qilishimiz mumkin.

ARGUMENT VA PARAMETER

Funksiya yaratishda, qavs ichida berilgan, funksiya to'g'ri ishlashi uchun uzatiladigan qiymat parameter deb ataladi. Yuqoridagi misolda ism bu `salom_ber` funksiyasining parametri.

Foydalanuvchi funksiyaga murojat qilishda funksiya uzatgan qiymat esa argument deb ataladi. `salom_ber('hasan')` kodida 'hasan' bu argument.

Ba'zi manbalarda yoki darslarda argument va parametr so'zlari almashtirib ishlatilishi ham kuzatiladi.

FUNKSIYAGA BIR NECHTA ARGUMENT UZATISH

Shunday funksiyalar bor, bir emas bir nechta parameter talab qilishi mumkin, foydalanuvchi esa o'z navbatida bir nechta argumentlar taqdim qilishi kerak. Funksiyaga argument uzatishning bir necha usuli bor, keling ular bilan birma bir tanishamiz. **Kod_5**

TO'G'RI TARTIBDA UZATISH

Bu usulda, funksiya parametrlari qaysi tartibda yozilgan bo'lsa, argumentlar ham aynan shu ketma-ketlikda uzatilishi kerak. Keling bitta misol ko'ramiz. **Kod_6** funksiya foydalanuvchining ismi va familiyasini parametr sifatida qabul qilib, ularni jamlab xabar chiqaradi.

Agar argumentlarni noto'g'ri ketma-ketlikda bersak, natija ham biz kutganday chiqmaydi: **kod_7**

Ko'p xolatlarda esa, argumentlarni noto'g'ri tartibda uzatish xatolikka ham olib kelishi mumkin. **Kod_8**

KALIT SO'Z BILAN UZATISH

Yuqoridagi kabi holatlarning oldini olish uchun argumentlarni parametr nomi bilan qo'shib uzatishimiz mumkin. Buning uchun funksiyaga o'zgartirish kiritish talab qilinmaydi. **Kod_9**

yosh_hisobla(tugilgan_yil=1997, ism='olim')

Natija: Olim 23 yoshda

Yuqoridagi misolda funksiyani chaqirishda biz parametrlar ketma-ketligiga rioya qilmagan bo'lsakda, argumentlarni parametr_nomi=qiymat ko'rinishida yozganimiz sababli funksiya to'g'ri ishladi.

Huddi shu kabi yuqoridagi toliq_ism funksiyasiga murojat qilishimiz mumkin: **Kod_10**

Kalit so'z usulidan foydalanganda parametr nomi to'g'ri yozilganiga ahamiyat bering.

STANDART QIYMAT

Funksiya yaratishda, istalgan parametr uchun standart qiymat ko'rsatib ketishimiz mumkin. Agar foydalanuvchi shu parametr uchun qiymat (argument) kiritmasa, funksiya bajarilishi jarayonida standart qiymat ishlatiladi. Standart qiymatni funksiya yaratish vaqtida parametr = qiymat ko'rinishida beriladi. **Kod_11**

Yuqoridagi misolda biz joriy_yil parametriga 2023 standart qiymatini berib ketdik.

Funksiya yaratishda, standart qiymatga ega parametrlar doim oxirida yozilishi kerak. Aks holda xatolik yuzaga keladi.

Keling avval funksiyani ikkala argument bilan chaqiramiz: **kod_11**

Endi esa faqat bitta argument (tugilgan_yil) bilan chaqiramiz: **kod_11**

Bu safar foydalanuvchi joriy_yil ni kiritmagani sababli, standart qiymat, 2023 ishlatildi.

FUNKSIYAGA MUROJAT QILISHDA XATOLIKLAR

Funksiyalarga murojat qilishda turli xatoliklarga yo'l qo'shimiz tabiiy. Bunday holatlarda Python qaytargan xatoni sinchiklab o'qib, xato qayrdaligini topishimiz va uni to'g'rilashimiz zarur. Quyida men avvalroq yaratgan funksiyalarimizni xato usullar bilan chaqiraman. Xato nimada ekanini topa olasizmi?

1-xatolik **Kod_12**

2-xatolik **Kod_13**

3-xatolik **Kod_14**