# Python და ორობითი სისტემა

Beautiful is better than ugly.
Explicit is better than implicit. Simple is better than complex. Complex is better than complicated. Flat is better than nested. Sparse is better than dense.
Readability counts. Special cases aren't

special enough to break the rules.

Although **practicality** beats purity. *Errors* should never pass silently. Unless **explicitly** silenced. In the face of ambiguity, **refuse** the temptation to guess. There should be **one** — and preferably only one — obvious way to do it. Although that way may not be obvious at first unless you're Dutch. **Now** is better than never. Although never is **often** better than right now. If the implementation is hard to explain, it's a **bad** 

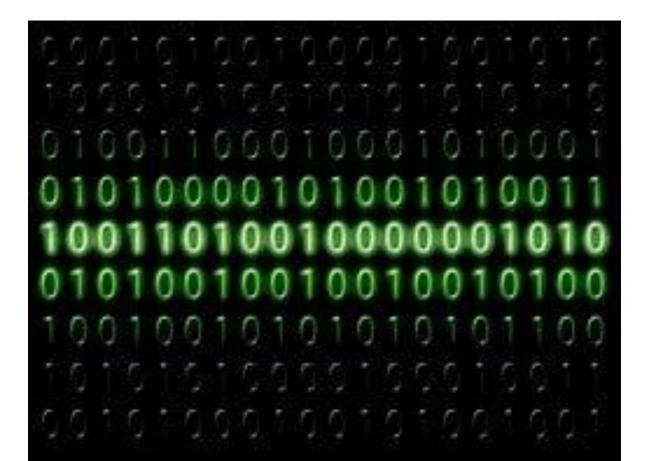
is easy to explain, it may be a good idea. Namespaces are one honking great idea — let's do more of those! idea. If the implementation is easy to explain, it may be a **good** idea. Namespaces are one honking great and idea—let's do more of those!

Although practicality beats purity. Evrors should never pass silently. Unless explicitly silenced. In the face of ambiguity, refuse the temptation to guess. There should be one — and preferably only one — obvious way to do it. Although that way may not be obvious at first unless you're Dutch. Now is better than never. Although never is often better than right now. If the implementation is hard to explain, it's a bad

Explicit is better than ugly.

Explicit is better than implicit. Simple is better than complex. Complex is better than than complicated. Flat is better than nested. Sparse is better than dense. Readability counts. Special cases aren't special cases.

python



## რა არის პროგრამირების ენა?

• პროგრამირების ენა არის ინსტრუქციათა ნაკრები, რომელიც გამოიყენება პროგრამების დასაწერად. პროგრამირების ენა აღიწერება სინტაქსისა და სემანტიკის მეშვეობით, რაც ჩვეულებრივ განისაზღვრება ფორმალური ენით. როგორც წესი ენები უზრუნველყოფენ მახასიათებლებს როგორიცაა ტიპის სისტემა, ცვლადები, და შეცდომათა დასამუშავებელი მექანიზმები. პროგრამირების ენის იმპლემენტაცია აუცილებელია პროგრამის გასაშვებად, კერძოდ ინტერპრეტატორის ან კომპილატორის გამოყენებით. ინტერპრეტატორი პირდაპირ ასრულებს წყაროს კოდს, ხოლო კომპილატორი აწარმოებს შესრულებად პროგრამას. ②წორედ ასეთი ენაა Python.

- Python წარმოადგენს <u>მაღალი დონის</u> პროგრამირების დინამიურ <u>ენას</u> შესაძლებლობათა ფართო სპექტრით, რომლის დიზაინი ყურადღებას ამახვილებს კოდის ადვილად წაკითხვადობაზე.პაითონის <u>სტანდარტული ბიბლიოთეკა</u> დიდია და ყოვლელმხრივი. ის იყენებს ძალზე იშვიათ გამომყოფებს ბლოკებისთვის აბზაცებს.
- პაითონი <u>მულტიპარადიგმული</u> ენაა და მოიცავს როგორც პროცედურული, იმპერატიული, ისე ფუნქციონალური პროგრამირების ელემენტებს. მას ხშირად ადარებენ ისეთ ენებს, როგორებიცაა Tcl, <u>Perl, Ruby, Scheme</u> და <u>Java</u>. პაითონის მახასიათებელთა შორის აღსანიშნავია:
- უაღრესად ცხადი, გასაგები და კითხვადი სინტაქსი
- ინტროსპექცია
- ობიექტ ორიენტირებული მიდგომის მულტიპარადიგმული დიზაინი
- სრული მოდულარულობა
- მონაცემთა უაღრესად მაღალი დონის ტიპირება
- მეტაპროგრამირება
- ფართო სტანდარტული ბიბლიოთეკა

### ორობითი სისტემა

- ორობითი თვლის სისტემა ან თვლის ორობითი სისტემა პოზიციური თვლის სისტემა, რომლის ფუმეა 2. ამ სისტემაში იყენებენ მხოლოდ ორ ნიშანს ციფრებს 0 და 1. აქაც, ისევე როგორც ყოველ პოზიციურ სისტემაში, ციფრის მნიშვნელობა დამატებით დამოკიდებულია მის მიერ დაკავებულ ადგილზე. რიცხვი 2 ითვლება მეორე თანრიგის ერთეულად და ჩაიწერება ასე: 10 (იკითხება: "ერთი, ნული"). შემდეგი თანრიგის ყოველი ერთული ორჯერ მეტია წინაზე, ე. ი. ეს ერთეულები ადგენენ რიცხვთა მიმდევრობას: 2, 4, 8, 16, ..., 2n, ... იმისათვის, რომ ათობით სისტემაში ჩაწერილი რიცხვი ჩაიწეროს ორობით სისტემაში, მას მიმდევრობით ყოფენ 2-ზე და მიღებულ ნაშთებს (0 და 1) ჩაწერენ რიგით ბოლოდან პირველისაკენ. მაგ., 27=13·2+1; 13=6·2+1; 6=3·2+0; 3=1·2+1; 1=0·2+1; ამრიგად 27-ის ორობითი ჩაწერა იქნება 11011.
- ამ სისტემაში განსაკუთრებით მარტივად სრულდება ყველა არითმეტიკული მოქმედება: მაგ., გამრავლების ტაბულა დაიყვანება ერთ ტოლობამდე 1·1=1, მაგრამ რიცხვების ჩაწერა მოითხოვს ციფრების დიდ რაოდენობას. მაგ., რიცხვი 7000 იქნება 13-ნიშნა, მაგრამ იმის გამო, რომ ეს სისტემა იყენებს მხოლოდ ორ ციფრს, იგი ხშირ შემთხვევაში სასარგებლოა თეორიული საკითხების განხილვისას და ეგმ-ზე გამოთვლებისას.

# დეკლარაცია

• 🛚 ვლადის დეკლარაცია არის ის მოქმედება როდრსაც კომპიუტერს ვაცნობებთ რომ რაღაც გარკვეული ცვლადი არსებობს.

#### რა არის print()

• print() არის რაიმეს დაბეჭვდა, იგივე ტერმინალზე გამოსახვა.

## ? ონაცემთა ტიპები

• მონაცემთა ტიპები — ეს არის კლასიფიკაციები, რომლებიც მიუთითებენ, თუ რა სახის მონაცემები შეიძლება შემონახოს ცვლადმა და რა ოპერაციები შეიძლება იმ მონაცემებზე შესრულდეს.