OSNOVE PROGRAMIRANJA • MATERIJAL ZA STUDENTE

Uvod u Python

Sintaksa, promenljive, operatori, tipovi podataka, uslovi, petlje i funkcije

PYTHON

```
# Prvi program
print("Hello, World!")
```

IZLAZ

```
Hello, World!
```

PREGLED

Šta ćemo danas raditi

01 Prvi program

02 Ulaz i izlaz

03 Promenljive

04 Operatori

05 Tipovi podataka

06 Uslovne naredbe

07 Petlje

08 Funkcije

UVOD

Python je jezik visokog nivoa sa čitljivom sintaksom

Čist kôd

Isti zadatak rešavate sa manje linija koda.

Više paradigmi

Objektno, funkcionalno i proceduralno programiranje.

Široka primena

Veb razvoj, automatizacija, analiza podataka i veštačka inteligencija.

Dinamički tipiziran

Tip se određuje pri dodeli, a memoriju automatski čisti sakupljač smeća.

PRVI PROGRAM

Hello World pokazuje osnovnu strukturu programa

PYTHON

```
# Ovo je komentar.
print("Hello, World!")
```

IZLAZ

```
Hello, World!
```

`#`	Komentar. Interpretator ga ignoriše, služi ljudima.
`print()`	Ugrađena funkcija koja ispisuje podatke na ekran.
`"Hello, World!"`	String u jednostrukim ili dvostrukim navodnicima.
`Uvlačenje`	Blokove koda određuje uvlačenje, a ne vitičaste zagrade {}.

ULAZ I IZLAZ

input() čita od korisnika, print() ispisuje rezultat

PYTHON

```
name = input("Enter your name: ")
print("Hello,", name)

x, y = input("Dva broja: ").split()

i = int(input("How old are you?: "))
f = float(input("Evaluate 7/2: "))
print(i, f)   # 23 3.5
```

input() uvek vraća string

Za brojeve koristite konverziju tipa: int() ili float().

Više vrednosti

print() prima više argumenata odvojenih zarezom; split() deli unos na delove.

PROMENLJIVE

Promenljiva je ime dodeljeno vrednosti

Pravila imenovanja

- Slova, cifre i donja crta _
- Prvi znak ne sme biti cifra
- Velika i mala slova se razlikuju: myVar ≠ myvar
- Ključne reči (if, for, class…) su zabranjene

ISPRAVNO

```
age = 21
_colour = "lilac"
total_score = 90
```

NEISPRAVNO → SyntaxError

```
1name = "Error"     # počinje cifrom
class = 10          # ključna reč
user-name = "Doe"   # sadrži crticu
```

KAKO RADI MEMORIJA

Promenljive su etikete na objektima, ne kutije

KORAK 1

```
x = 5
```

x → [5]

Python kreira objekat 5 i x pokazuje na njega.

KORAK 2

```
y = x
```

x → [5] ← y

Deljena referenca: obe pokazuju na isti objekat.

KORAK 3

```
x = 'Geeks'
```

x → ['Geeks']
y → [5]

x dobija novi objekat; y ostaje vezan za 5.

Objekat na koji ne pokazuje nijedna promenljiva uklanja sakupljač smeća (garbage collection).

OPERATORI

Sedam vrsta operatora rade nad operandima

Kategorija	Operatori	Primer
Aritmetički	`+ - * / // % **`	`15 // 4 → 3`
Poređenja	`== != > < >= <=`	`13 > 33 → False`
Logički	`and or not`	`True and False → False`
Dodeljivanja	`= += -= *= /=`	`b += a ≡ b = b + a`
Bitovni	`& | ^ ~ << >>`	`10 & 4 → 0`
Pripadnosti	`in not in`	`20 in [10, 20] → True`
Identiteta	`is is not`	`a is c → True`

PRIORITET I ASOCIJATIVNOST

Redosled operacija menja rezultat izraza

```
10 + 20 * 30
```

610

Množenje ima veći prioritet od sabiranja.

```
100 / 10 * 10
```

100.0

Isti prioritet: računa se sleva nadesno.

```
2 ** 3 ** 2
```

512

Stepenovanje ide zdesna nalevo: 2 ** 9.

Logički operatori: prvo **not**, zatim **and**, na kraju **or**. Zagrade uvek imaju prednost.

TIPOVI PODATAKA

Svaka vrednost u Pythonu je objekat određenog tipa

Kategorija	Tip	Primer vrednosti
Numerički	`int, float, complex`	`10  3.14  2 + 3j`
Tekstualni	`str`	`"Zdravo, svet!"`
Bulevski	`bool`	`True  False`
Sekvence	`list, tuple, range`	`[1, 2, 3]  (10, 20)`
Skupovi	`set, frozenset`	`{1, 2, 3}`
Rečnik	`dict`	`{"ime": "Ana"}`
Bez vrednosti	`NoneType`	`None`

KOLEKCIJE

Lista, torka, skup i rečnik se ponašaju različito

Svojstvo	`list`	`tuple`	`set`	`dict`
Zapis	`[1, 2]`	`(1, 2)`	`{1, 2}`	`{"k": 1}`
Indeksiranje	Da	Da	Ne	Po ključu
Izmena posle kreiranja	Da	Ne	Da	Da
Duplikati	Dozvoljeni	Dozvoljeni	Uklanjaju se	Ključevi jedinstveni

Torka sa jednim elementom mora imati zarez: **(1,)**. Bez zareza (1) je samo broj u zagradi.

USLOVNE NAREDBE

Uslov određuje koji blok koda se izvršava

PYTHON

```
poeni = 75
if poeni ⩾ 90:
    print("Ocena: 10")
elif poeni ⩾ 80:
    print("Ocena: 9")
elif poeni ⩾ 70:
    print("Ocena: 8")
else:
    print("Ocena: ispod 8")
```

IZLAZ

```
Ocena: 8
```

`if`

Izvršava blok samo kada je uslov True.

`elif`

Proverava sledeći uslov; prvi tačan pobeđuje.

`else`

Rezervni blok kada nijedan uslov nije tačan.

`x if uslov else y`

Ternarni operator: if-else u jednoj liniji.

`match-case`

Poređenje sa uzorcima, od Pythona 3.10.

PETLJE

for prolazi kroz sekvencu, while radi dok važi uslov

FOR

```
for i in range(1, 10, 2):
    print(i)   # 1 3 5 7 9

for idx, val in enumerate(a):
    print(idx, val)
```

Broj iteracija je poznat ili prolazite kroz listu, string ili range.

WHILE

```
cnt = 0
while cnt < 3:
    cnt += 1
    print("Hello Geek")

# while True: bez break nikad ne staje
```

Broj iteracija zavisi od uslova koji se proverava pre svakog prolaza.

FUNKCIJE

Funkcija je imenovani blok koda za ponovnu upotrebu

PYTHON

```
def saberi(a, b):
    """Vraća zbir dva broja."""
    rezultat = a + b
    return rezultat

zbir = saberi(5, 3)
print(zbir)   # 8
```

`def`

Ključna reč koja počinje definiciju.

`saberi(a, b)`

Ime funkcije i parametri u zagradama.

`:  + uvlačenje`

Dvotačka i uvučeno telo funkcije.

`return`

Vraća rezultat; bez njega funkcija vraća None.

ARGUMENTI FUNKCIJE

Četiri načina da prosledite vrednosti funkciji

Podrazumevani

```
def f(x, y=50):
```

y dobija 50 ako se ne prosledi.

Imenovani

```
student(lname="P", fname="G")
```

Redosled nije bitan, važi ime.

Pozicioni

```
nameAge("Olivia", 27)
```

Vrednosti idu redom; zamena daje pogrešan rezultat.

Proizvoljni

```
def f(*args, **kwargs):
```

*args je torka, **kwargs je rečnik.

PRENOS ARGUMENATA

Mutabilnost objekta određuje šta funkcija može da promeni

```python
def izmeni(x):
    x[0] = 20

b = [10, 11, 12]
izmeni(b)
print(b)        # [20, 11, 12]

def izmeni2(x):
    x = 20

a = 10
izmeni2(a)
print(a)        # 10
```

Promenljivi objekti

list, dict: izmene unutar funkcije vide se i spolja.

Nepromenljivi objekti

int, float, str, tuple: dodela pravi novi lokalni objekat, original ostaje isti.

KLJUČNA PORUKA

Uvlačenje je sintaksa, a promenljive su reference na objekte.

Za vežbu: pokrenite svaki primer, promenite vrednosti i predvidite izlaz pre izvršavanja.