




PROGRAMIRANJE
Predavanje 11

Ishod 5

1

**PROSLJEĐIVANJE OBJEKATA
U FUNKCIJU**

Strana • 2



2

Uvod

- Obično se kaže da je u Pythonu sve objekt
 - Svaki pojedini `int` je objekt
 - Svaki pojedini `string` je objekt
 - Svaki rječnik je objekt
 - Svaka lista je objekt
 - Svaka instanca klase je objekt
 - Svaka funkcija je objekt

Strana • 3



3

Identifikatori

- Važno je razlikovati identifikatore i objekte
- Identifikatori su samo nazivi za objekte
 - Svaki objekt može imati nula, jedan ili više identifikatora
 - Nazivi varijabli, parametara, funkcija, klasa i slično su identifikatori
- Ako želimo znati na koji objekt identifikator pokazuje, možemo koristiti funkciju `id()`
 - Vraća identitet objekta koji je nepromjenjiv

```
a = "Siječanj 2022."
print(id(a))
```

Strana • 4



4

Stog i hrpa

- Identifikatori se čuvaju u dijelu memorije zvanom stog (engl. *stack*)
- Objekti se kreiraju u dijelu memorije zvanom hrpa (engl. *heap*)

Strana * 5



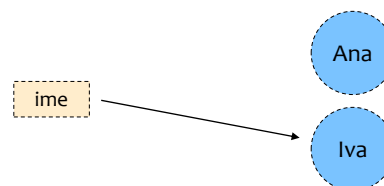
5

Identifikator i objekti: primjer

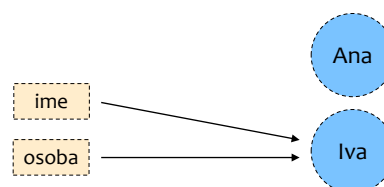
```
ime = "Ana"
print(id(ime))
```



```
ime = "Iva"
print(id(ime))
```



```
osoba = ime
print(id(osoba))
```



Strana * 6



6

Životni vijek objekta

- Python automatski vodi računa o životnom vijeku objekta:
 - Kad se objekt kreira, Python mu dodjeljuje brojač
 - Svaki put kad referenciramo objekt, brojač se povećava za 1
 - Dodijelimo objekt varijabli
 - Dodamo objekt u listu
 - Pošaljemo objekt kao parametar u funkciju
 - Svaki put kad dereferenciramo objekt, brojač se smanjuje za 1
 - Kad brojač dođe do 0, Python ga uništava i oslobađa memoriju
 - U stvari, *Garbage Collector* to učini u nekom trenutku

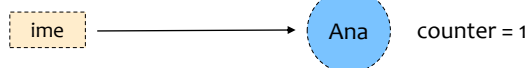
Strana • 7



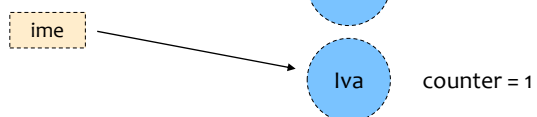
7

Identifikator, objekti i brojači: primjer

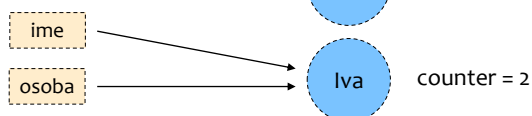
```
ime = "Ana"
```



```
ime = "Iva"
```



```
osoba = ime
```



Strana • 8



8

(De)referenciranje objekata: primjer

```
def foo(param1):
    pass

x = 10

brojevi = [32, 10, 52]

foo(x)

<Nakon poziva funkcije>
```

Strana * 9

ALGEBRA

9

IMENSKI PROSTORI

Strana * 10

ALGEBRA

10

Uvod

- Imenski prostor (engl. *namespace*) je rječnik koji povezuje identifikatore s objektima
 - Ključ je identifikator
 - Vrijednost je objekt
- U svakom trenutku u Python programu postoji više imenskih prostora koji su potpuno odvojeni jedan od drugih
- Imenski prostori definiraju što nam je dostupno za korištenje
- Funkcija `dir()` vraća sadržaj imenskog prostora
- Pokušajmo: `import this`
 - Pročitajmo zadnju rečenicu

Strana • 11



11

Tipovi imenskih prostora

- U Python programu, postoje tri osnovna tipa imenskih prostora:
 - Ugrađeni (engl. *built-in*)
 - Globalni (engl. *global*)
 - Lokalni (engl. *local*)
- Kako Python izvršava naš program, prema potrebi kreira i briše imenske prostore

Strana • 12



12

Ugrađeni imenski prostor

- Sadrži sve ugrađene Python objekte
- Kreira se pokretanjem programa, briše se završetkom programa
 - U Jupyteru to nije do kraja transparentno
 - Primjerice, zato su nam funkcije `print()`, `input()`, `type()` itd. dostupne bez potrebe importanja
- Sadržaj dostupan ovako:


```
print(dir(__builtins__))
```

Strana • 13



13

Globalni imenski prostor

- Svaki modul kreira svoj vlastiti globalni imenski prostor
 - Sadrži sve identifikatore definirane u nekom modulu
- Kreira se početkom izvršavanja modula, briše se završetkom izvršavanja modula
- Vrijedi sljedeće:
 - Postoji jedan globalni imenski prostor za trenutni modul (*notebook*)
 - Za svaki `import`-ani modul kreira se novi globalni imenski prostor
- Sadržaj dostupan ovako:

```
print(dir()) # Globalni za trenutni notebook
print(dir(random)) # Globalni za importani modul
```

Strana • 14



14

Lokalni imenski prostor

- Lokalni imenski prostor se kreira početkom izvršavanja funkcije i briše se na kraju izvršavanja funkcije
- Sadržaj dostupan ovako:

```
def test():
    a = 10
    b = 20
    print(dir()) # Lokalni za trenutnu funkciju

test()
```

Strana • 15



15

import

- Ako želimo učitati cijeli modul u njegov vlastiti imenski prostor:


```
import math
print(math.pi)
```
- Ako želimo učitati cijeli modul u globalni imenski prostor:


```
from math import *
print(pi)
```
- Ako želimo učitati dio modula u globalni imenski prostor:


```
from math import pi
print(pi)
```

Strana • 16



16

DATUM I VRIJEME

Strana • 17



17

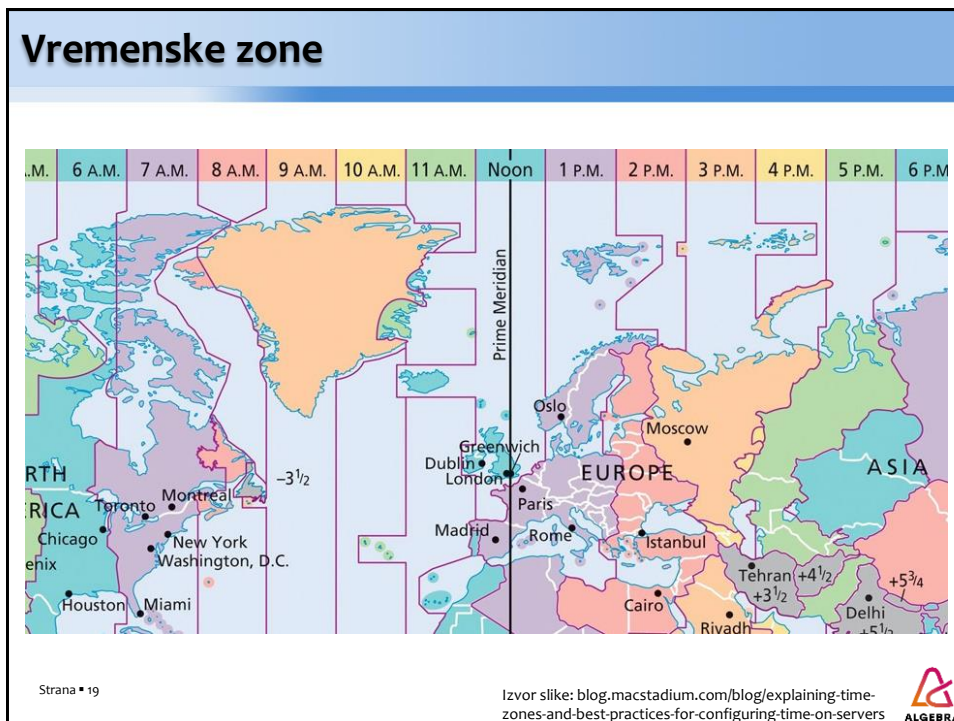
Uvod

- Svjesni (engl. *aware*) objekt iz modula datetime je onaj koji predstavlja nedvosmisleni trenutak u vremenu
 - Sadrži vremensku zonu i informaciju o ljetnom/zimskom računanju vremena
 - Primjerice, ako sazovemo sastanak 17.7.2021. u 09:00 po srednjeeuropskom vremenu, znajući da je tad ljetno računanje vremena, svi će se pojaviti na vrijeme
- Naivni (engl. *naive*) objekt je onaj ne sadrži dovoljno informacija da bi predstavljao nedvosmisleni trenutak u vremenu
 - Kako će završiti sastanak ako ga sazovemo 17.7.2021. u 09:00 bez dodatnih informacija? Kad će na sastanak doći Amerikanac, a kad Japanac?

Strana • 18



18



19

Mjerenje vremena

- UTC (engl. *Coordinated Universal Time*) predstavlja vrijeme na nultom (Griničkom) meridijanu
 - Naziva se i GMT (engl. *Greenwich Mean Time*)
 - Ne mijenja se ljeti/zimi
 - Predstavlja referencu za sve ostale vremenske zone
- Primjeri:
 - 23.12.2021. 17:00 u Zagrebu je 16:00 po GMT-u (Zagreb je +1)
 - 22.07.2022. 17:00 u Zagrebu je 15:00 po GMT-u (Zagreb je +2)

Strana * 20

ALGEBRA

20

Kako računala broje vrijeme

- Današnja računala broje vrijeme od trenutka zvanog Unix epoha
 - Radi se o 1.1.1970. u 00:00 UTC-a
- Doslovno, svako računalo broji koliko sekundi je prošlo od tog trenutka
 - Potencijalni problem Year 2038 za 32-bitne sustave
- Primjer u Pythonu:


```
import time
print(time.time())
print(time.time() / 60 / 60 / 24 / 365)
```
- Čovjeku ta informacija ne znači puno pa je pretvaramo u lokalni datum i vrijeme

Strana * 21



21

Zbrka oko lokalnog datuma i vremena

- Primjerice, ako piše da je koncert 7/5/2021 u 21:00, kad će se održati:
 - 7. svibnja 2021, ili
 - 5. srpnja 2021?
- Različite kulture vrijeme zapisuju na razne načine – potencijalni izvor nesporazuma
- ISO 8601 je standardizirani način zapisivanja vremena u računalima i ima format:


```
YYYY-MM-DD HH:MM:SS
```
- Ovako točno znamo kad će se koncert održati:


```
2021-07-05 21:00:00
```


Strana * 22



22

Modul datetime

- Cilj modula `datetime` je olakšati rad s datumom i vremenom
 - Nemojmo miješati s kompliciranijim modulom `time`
- Osnovni elementi modula su četiri klase:
 - `date` – naivni datum, sadrži varijable `year`, `month` i `day`
 - `time` – predstavlja vrijeme i sadrži varijable `hour`, `minute`, `second` i `tzinfo`
 - `datetime` – predstavlja datum i vrijeme i sadrži sve varijable kao dvije prethodne klase
 - `timedelta` – predstavlja razliku vremena između dva `datetime` objekta

Strana • 23  Detalji malo kasnije



23

Izrada objekata pomoću konstruktora

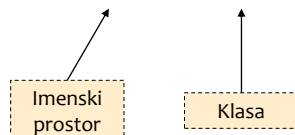
- Konstruktori su intuitivni i sadrže parametre jednake varijablama navedenim na prethodnom slajdu
- Primjer izrade objekata:

```
import datetime

datum_rodjenja = datetime.date(day=23, month=12, year=1998)

pocetak_predavanja = datetime.time(hour=18, minute=45, second=0)

koncert = datetime.datetime(day=7, month=6, year=2021, hour=19,
                             minute=30, second=0)
```



Strana • 24



24

Izrada objekata s trenutnim datumom i vremenom

- Ako želimo napraviti objekte koji sadrže trenutni datum i vrijeme, možemo pisati:

```
danasnji_dan = datetime.date(day=20, month=12, year=2021)
```

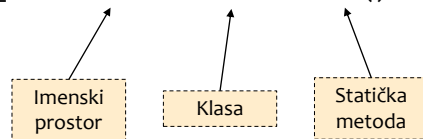
```
bas_sad = datetime.datetime(day=20, month=12, year=2021, hour=19,
                             minute=30, second=45)
```

- Postoji jednostavniji način

- Koristimo posebnu statičku metodu (engl. *classmethod*)

```
danasnji_dan = datetime.date.today()
```

```
bas_sad = datetime.datetime.now()
```



Strana • 25



25

Izrada objekata iz stringa

- Metoda `strptime()` prima string s datumom i format te kreira i vraća objekt tipa `datetime`

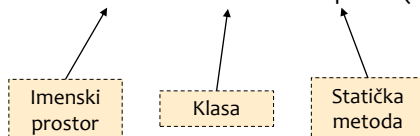
- Postoji kao statička metoda samo na klasi `datetime`

- Primjer:

```
t1 = datetime.datetime.strptime('7.5.2020', '%m.%d.%Y')
```

```
t2 = datetime.datetime.strptime('7.5.2020', '%d.%m.%Y')
```

```
t3 = datetime.datetime.strptime('7.5.2020 14:00', '%d.%m.%Y %H:%M')
```



Strana • 26



26

Osnovni elementi formata

- Osnovni elementi formata:

- Detaljnije na

<https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior>

Element	Značenje	Primjer
%d	Dan u mjesecu s vodećom nulom	01, 02, ..., 31
%m	Mjesec s vodećom nulom	01, 02, ..., 12
%Y	Četveroznamenkasta godina	..., 2020, 2021, ...
%H	Sati 0-24 s vodećom nulom	00, 01, ..., 23
%M	Minute s vodećom nulom	00, 01, ..., 59
%S	Sekunde s vodećom nulom	00, 01, ..., 59
%f	Mikrosekunde	000000, 000001, ..., 999999
%Z	Naziv vremenske zone	(empty), UTC, GMT

Strana • 27



27

Ispis datuma i vremena

- Umjesto *defaultnog* ispisa, možemo sami definirati prikaz kakav želimo

- Metoda `strftime()` radi vrlo slično metodi `strptime()`

- Dostupna na `date`, `datetime` i `time` objektima

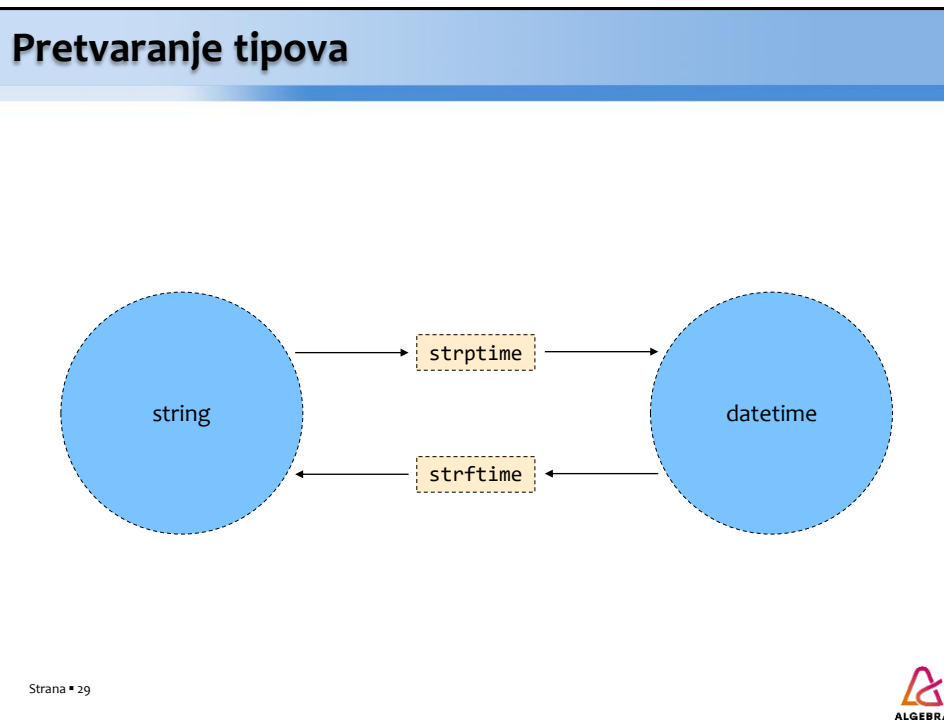
```
t1 = datetime.date.today()
t2 = datetime.datetime.now()

print(t1.strftime('%m.%d.%Y'))
print(t1.strftime('%d.%m.%Y'))
print(t2.strftime('%d.%m.%Y %H:%M'))
```

Strana • 28



28




29

Dani u tjednu

- Kad imamo objekt tipa `date` ili `datetime`, možemo na njemu pozvati metodu `isoweekday()` kako bismo dobili podataka o kojem danu je riječ:
 - 1 = ponedjeljak
 - 2 = utorak
 - 3 = srijeda
 - 4 = četvrtak
 - 5 = petak
 - 6 = subota
 - 7 = nedjelja

Strana • 30



30

Primjeri

1. Napišite program koji od korisnika učitava posebno dan, mjesec i godinu, izrađuje objekt tipa `date` te ga ispisuje u ISO i u hrvatskom formatu.
2. Napišite program koji od korisnika učitava posebno sate i minute, izrađuje objekt tipa `time` te ga ispisuje bez sekundi.
3. Napišite program koji od korisnika učitava posebno dan, mjesec, godinu, sate i minute, izrađuje `datetime` objekt te ga ispisuje u formatu: **Sat: 18:45, Datum: 22.04.2021.**
4. Napišite program koji uzima današnji datum i vrijeme i ispisuje ga u hrvatskom formatu.
5. Napišite funkciju koja prima tri datuma i vraća najmanji. Pozovite funkciju s tri datuma i ispišite najmanji.

Strana * 31



31

Primjeri

6. Napišite program koji od korisnika učitava kompletan datum sastanka u ISO formatu. Nakon toga, omogućite korisniku da unese troslovnju oznaku države te mu ispišite datum u formatu te države. Omogućite rad samo sa formatima HRV, USA i VNM (<https://gist.github.com/mlconnor/1887156>).
7. Učitajte od korisnika godinu i mjesec te napravite i ispišite prvih 15 datuma iz tog mjeseca u hrvatskom formatu.
8. Napišite program koji od korisnika unosi datume u hrvatskom formatu dok to korisnik želi, stavlja ih u listu te ih na kraju ispisuje u ISO formatu.

Strana * 32



32

Izrada objekta tipa `timedelta`

- Objekt tipa `timedelta` predstavlja razliku između dva `time`, `date` ili `datetime` objekta
- Možemo ga izraditi pomoću konstruktora koji prima sljedeće parametre (svi su opcionalni):
 - `days`, `seconds`, `minutes`, `hours`, `weeks`
- Na objektu je protek vremena dostupan samo u varijablama: `days` (punih dana) i `seconds` (sekundi u sljedećem danu)
- Svaki parametar može biti pozitivan ili negativan broj
- Primjer:


```
x = datetime.timedelta(days=2, minutes=-5)
print(x.days)
print(x.seconds)
```

Strana * 33



33

Izrada objekta tipa `timedelta`

- Drugi način izrade objekta tipa `timedelta` je na temelju razlike dva objekta
- Primjer:


```
pocetak_go = datetime.date(day=23, month=12, year=2020)
kraj_go = datetime.date(day=7, month=1, year=2021)
trajanje_go = kraj_go - pocetak_go
```

Razlika dva `date`-a
je `timedelta`

Ne možemo oduzeti
dva `time`-a

```
pocetak_predavanja = datetime.datetime(day=4, month=1, year=2021,
                                         hour=18, minute=45)
kraj_predavanja = datetime.datetime(day=4, month=1, year=2021,
                                     hour=20, minute=15)
trajanje_predavanja = kraj_predavanja - pocetak_predavanja
```

Razlika dva `datetime`-a
je `timedelta`

Strana * 34



34

Računanje s objektom tipa `timedelta`

- Kad imamo `timedelta` objekt, možemo ga dodati ili oduzeti od `date` ili `datetime` objekta:

```
dv1 = datetime.datetime(day=23, month=12, year=2020,
                        hour=17, minute=39, second=15)
d = datetime.timedelta(days=3, hours=-4)
```

```
dv2 = dv1 + d
dv3 = dv1 - d
```

Strana * 35



35

Primjeri

9. Napišite program koji od korisnika učitava datum. Nakon toga, ispišite datum koji je bio prije 100 dana, prije 1.000 dana te prije 10.000 dana.
10. Napišite program koji od korisnika učitava datum rođenja i mjeru (dani, sati ili minute) te ispisuje koliko je prošlo od rođenja do danas, u zadanoj mjeri.
11. Napišite program koji od korisnika učitava datum i vrijeme početka prvog predavanja te ispisuje termine svih 15 predavanja u hrvatskom formatu, ako znamo da se svako predavanje odvija u istom terminu svakog tjedna.

Strana * 36



36

Primjeri

12. Napišite program koji simulira budilicu. Omogućite korisniku da upiše za koliko vremena želi buđenje (omogućite mu unos u minutama ili sekundama) te kad istekne željena količina vremena, ispišite poruku BUĐENJE. Hint: koristite funkciju `sleep()` iz modula `time`.
13. Koristeći podatke iz tablice na sljedećem slajdu, napišite program koji omogućuje korisniku da upiše datum početka i kraja godišnjeg odmora pa mu izračunajte i ispišite koliko će radnih dana potrošiti (smatrajte da su radni dani ponedjeljak-petak kad nije praznik).

Strana • 37



37

Državni praznici RH u 2021.

Praznik	Datum
Nova godina	01.01.2021.
Sveta tri kralja	06.01.2021.
Uskrs	04.04.2021.
Uskršnji ponedjeljak	05.04.2021.
Praznik rada	01.05.2021.
Dan državnosti	30.05.2021.
Tijelovo	03.06.2021.
Dan antifašističke borbe	22.06.2021.
Dan pobjede i domovinske zahvalnosti i Dan hrvatskih branitelja	05.08.2021.
Velika Gospa	15.08.2021.
Dan svih svetih	01.11.2021.
Dan sjećanja na žrtve Domovinskog rata i Dan sjećanja na žrtvu Vukovara i Škabrnje	18.11.2021.
Božić	25.12.2021.
Sveti Stjepan	26.12.2021.

Strana • 38



38

Za sljedeće predavanje

- ✓ Ponoviti sve iz ovog predavanja
- ✓ Pročitati i isprobati primjere:
 - programiz.com/python-programming/namespace
 - code.tutsplus.com/tutorials/what-are-python-namespaces-and-why-are-they-needed--cms-28598
 - realpython.com/python-datetime/

Strana * 39

