

IMPORT MODULA

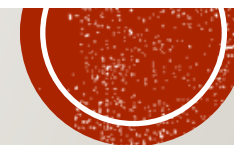
Igor Buzov, dipl. inf. viši predavač

Centar umjetne inteligencije Lipik



```
1
2 import pandas as pd
3 from sklearn.datasets import load_digits
4 from sklearn.model_selection import train_test_split
5 from sklearn.svm import SVC
6 import matplotlib.pyplot as plt
7
8
9 digits = load_digits()
10 print(dir(digits))
11
12
```

['DESCR', 'data', 'images', 'target', 'target_names']



PONAVLJANJE

- Koja je razlika između grešaka sintakse i runtime errora?
- Što je iznimka (exception)?
- S kojim naredbama upravljamo iznimkama?
- Koje vrste grešaka postoje?



PONAVLJANJE — ŠTO RADI OVAJ KOD?

```
unos_korisnika = input("Unesite cijeli broj: ")
```

```
try:
```

```
    broj = int(unos_korisnika)
```

```
    print(f"Uspješno ste unijeli broj: {broj}")
```

```
except ValueError:
```

```
    print("Greška: Niste unijeli cijeli broj. Molimo pokušajte ponovo.")
```

```
print("Program je završio s radom.")
```



PONAVLJANJE – POPRAVI KOD

```
print("--- Konverter Celzijusa u Fahrenheite ---")
```

```
temp_celsius_str = input("Unesite temperaturu u Celzijusima")
```

```
try:
```

```
temp_celsius = float(temp_celsius_str)
```

```
temp_fahrenheit = (temp_celsius * 9/5) + 32
```

```
except ValueError
```

```
print("Greska: Unijeli ste neispravan format broja. Molimo unesite samo numeričke vrijednosti.")
```

```
else:
```

```
print("Temperatura u Fahrenheitima iznosi:", temp_fahrenheit)
```

```
print("Pretvorba uspješna.")
```

```
print("Program završio.")
```



UVOD

- Što je rezultat ovog koda:

```
lista = [5, 9, 12, 4, 3]
```

```
print (sum(lista))
```

- Kako biste izračunali prosjek?



MODULI

- Python je jako moćan jezik, ali ne može imati sve funkcije ugrađene direktno u sebi jer bi to bio preogroman program
- Zato se koristi sustav modula
- Modul u Pythonu je jednostavno datoteka (.py) koja sadrži Python kod
- Svaki modul može sadržavati:
 - Funkcije: Skupove naredbi koje obavljaju određeni zadatak (npr. `statistics.mean()` koja računa prosjek).
 - Klase: "Nacrte" za stvaranje objekata (npr. ako imaš modul za rad s korisnicima, možda bi imao klasu `Korisnik`).
 - Varijable: Podatke koje želiš podijeliti.



VAŽNOST MODULA

- Organizacija: Pomažu da kod bude uredan i logički podijeljen.
- Ponovna iskoristivost: Kada se jednom napiše nešto korisno u modulu, može se to koristiti u mnogim drugim svojim programima tako što se jednostavno "uveze" (eng. import). Ne mora se pisati isti kod iznova.
- Čitljivost: Programi su lakši za čitanje i razumijevanje jer su podijeljeni na manje, smislene dijelove.
- Izbjegavanje sukoba: Funkcije iz različitih modula mogu imati isto ime (npr. dvije različite obradi_podatke() funkcije), ali Python ih razlikuje jer pripadaju različitim modulima (npr. modul_A.obradi_podatke() i modul_B.obradi_podatke()).



KORIŠTENJE MODULA

- Svaki modul se uvozi naredbom `import` nakon koje slijedi naziv modula
 - Npr `"import pandas as pd"`
 - Pd je ovdje kratica koja će se poslije koristiti
- Moduli se importaju na početku (vrhu programa)
- Importaju se samo potrebni moduli, NE SVI moduli koji postoje na svijetu pa ako zatreba, neka se nađe
- Može se koristiti i sintaksa `"from MODUL import FUNKCIJA"`



PRIMJER 1 - IZRADA VLASTITOG MODULA

- Kreirati dvije datoteke:

vjezbaModuli.py

modul_zbrajanje.py

- vjezbaModuli.py nam je glavna datoteka
- Modul_zbrajanje.py će biti naš novi modul u koji ćemo pohraniti jednu funkciju koju želimo redovito koristiti
- Oba filea neka budu spremljena u isti folder



KOD NAŠEG NOVOG MODULA MODULI_ZBRAJANJE.PY

```
# modul_zbrajanje.py
```

```
def zbroji_dva_broja(a, b):  
    return a + b
```



KOD NAŠEG NOVOG GLAVNOG PROGRAMA VJEZBAMODULI.PY

```
# glavni_program.py
import modul_zbrajanje

print("--- Primjer korištenja vlastitog modula ---")
broj1 = float(input("Unesi prvi broj: "))
broj2 = float(input("Unesi drugi broj: "))

# Pozivamo funkciju iz našeg modula
# Koristimo sintaksu: ImeModula.ImeFunkcije(argumenti)
rezultat = modul_zbrajanje.zbroji_dva_broja(broj1, broj2)

print ("Zbroj glasi: ", rezultat)
print("Program je završio s radom.")
```

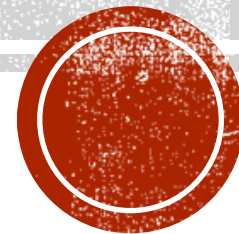


ZADATAK 1

- Napravite novi modul koji ćete nazvati `modul_mnozenje.py`
 - U njemu množite dva broja unutar funkcije i vraćate vrijednost
- Importajte taj modul u glavni program
- Unesite dva broja i kao rezultat ispišite umnožak



**NAPRAVITE NOVI FILE
POD NAZIVOM MODUL2.PY**



KORIŠTENJE POSTOJEĆIH MODULA

- Uz izradu vlastitog modula, mogu se koristiti i postojeći moduli
 - numpy
 - pandas
 - statistics
 - matplotlib
 - sklearn
- I mnogi, mnogi drugi



MODULE STATISTICS

- dio je standardne Python biblioteke i pruža funkcije za izračun **matematičke statistike** numeričkih (realnih) podataka
- Njegov je cilj osigurati brze, pouzdane i matematički ispravne metode za osnovne statističke izračune.
- Najpopularnije funkcije:
 - `mean()`
 - `median()`
 - `mode()`
 - `stdev()`



PRIMJER 2

```
import statistics
```

```
lista = [5, 7, 12, 4, -3, 56, 9]
```

```
prosjek = statistics.mean(lista)
```

```
print (prosjek)
```



ZADATAK 2

- Za ovu listu izračunajte prosjek i medijan:

place = [1000, 1200, 850, 750, 1300, 1400, 42000, 1500, 1050, 920]

- Za ovu listu izračunajte najčešći oblik plaćanja

placanje = ["kartica", "gotovina", "kartica", "kartica", "paypal", "kartica"]



ZADATAK 3 - ANALIZA PROČIŠĆENIH TEMPERATURNIH SENZORA

- Zamislite da prikupljate podatke s temperaturnih senzora. Ponekad senzori zabilježe grešku, koju mi označavamo s posebnom vrijednošću (npr. 999.0).
- Vaš je zadatak da pročistite listu sirovih očitavanja tako da uklonite greške, a zatim izračunate prosječnu vrijednost samo ispravnih očitavanja.
- Ispravna očitavanja pohranite u novu listu
- Koristite ovu listu:
- `sirova_ocitanja = [20.5, 21.0, 999.0, 19.8, 22.3, 999.0, 20.1, 25.0, 999.0, 21.5]`
- Koristite i `try/except` naredbe
- Postoji mogućnost da su vaši senzori zabilježili samo posebne vrijednosti tako da bi vaša nova lista onda bila prazna
- U tom slučaju diže se iznimka `"statistics.StatisticsError:"`



MODUL PANDAS

- Pandas je modul namijenjen uvozu, čišćenju, manipulaciji i analizi podataka
- Podaci mogu dolaziti u različitim formatima (txt, csv, xlsx, baze podataka itd)
- Ključne strukture su Series (jedan stupac) te DataFrame (tablica podataka)
- Pandas nije dio standardne biblioteke, spada u third party biblioteke
- Instalacija – u terminalu:
 - pip install pandas
 - Dovoljno jedanput instalirati



UČITAVANJE FILEA "ADVERTISING.CSV"

- Preuzeti datoteku "Advertising.csv" i snimiti je u isti folder gdje se nalazi vaš python code

```
import pandas as pd
```

```
# u zagradi je moja putanja, staviti svoju, provjeriti jesu li kose crte ispravne
```

```
podaci = pd.read_csv ("D:\\IGOR\\Posao\\LIPIK AI\\LIPIK  
Python\\python_work2\\Advertising.csv")
```

```
print (podaci.head())
```

```
print (podaci.describe())
```



MATPLOTLIB — MODUL ZA GRAFIČKI PRIKAZ

- Instalirati ga u terminalu s naredbom:

```
pip install matplotlib
```

- Matplotlib je sveobuhvatna biblioteka za izradu statičnih, animiranih i interaktivnih vizualizacija u Pythonu
- Omogućava stvaranje raznih vrsta grafova: linijskih, stupčastih, raspršenih (scatter), tortnih (pie), histograma itd.



PRIMJER 3

- Preuzmite kod za grafičku usporedbu ulaganja u oglašavanje na televiziji i prodaje
- Modificirajte grafikon i usporedite ulaganje u oglašavanje u novinama i prodaju – popravite i naslove grafova

