



Sveučilište Josipa Jurja Strossmayera u Osijeku  
**Građevinski i arhitektonski fakultet Osijek**  
Josip Juraj Strossmayer University of Osijek  
**Faculty of Civil Engineering and Architecture Osijek**



## **Računalno programiranje u građevinarstvu (254810) | Tjedan 1/15**

Akademski godina 2025./2026.

# **Uvod u Python**

**Plan i sadržaj predmeta, instalacija i paketi, *Spyder*, osnovne matematičke operacije**

## **Marin Grubišić**

**Izv. prof. dr. sc. / mag. ing. aedif.**

Sveučilište J. J. Strossmayera u Osijeku (UNIOS)  
Građevinski i arhitektonski fakultet Osijek (GrAFOS)  
Zavod za tehničku mehaniku (ZTM)  
Predsjednik Katedre za teoriju konstrukcija  
Vladimira Preloga 3, **Ured II.26**, HR–31 000 Osijek, Hrvatska

**[marin.grubisic@gfos.hr](mailto:marin.grubisic@gfos.hr) | [www.maringrubisic.com](http://www.maringrubisic.com)**

**Structural &  
Earthquake Engineering**

# Sadržaj prezentacije

- 1 Uvod, ciljevi i sadržaj**
- 2 Uvod i motivacija**
- 3 Organizacija predmeta**
- 4 Spyder IDE**
- 5 Osnovni tipovi podataka**
- 6 Aritmetičke operacije**
- 7 Praktični primjeri**
- 8 Zaključci**



## Ciljevi i sadržaj

- Cilj ovog predmeta je pripremiti studente za numerički rad u **Pythonu**.
- Pružiti studentima **osnove računalnog programiranja** s naglaskom na rješavanje problema iz područja građevinarstva.
- Naučiti studente služiti se **generativnom umjetnom inteligencijom (GenAI)** za potrebe generiranja programskog kôda u Python programskom jeziku (ChatGPT, Gemini, Grok, DeepSeek, Kimi, ...).

Mogućnost odabira predmeta **Računalno programiranje u građevinarstvu za temu završnog rada**.

Alati za **generativnu umjetnu inteligenciju (GenAI)**

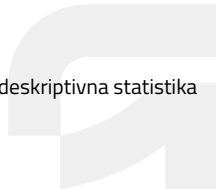
- OpenAI ChatGPT (**chatgpt.com**).
- Google Gemini (**gemini.google.com**).
- xAI Grok (**grok.com**).
- DeepSeek AI (**deepseek.com**).
- Moonshot Kimi (**kimi.com**).

# Organizacija predmeta

## Organizacija predmeta, obaveze studenata i plan nastave

Okvirni plan nastave po terminima (**ponedjeljkom, od 10 do 12 sati, predavaonica II.47, 2. kat**):

- 1 Plan i sadržaj predmeta, instalacija i paketi, *JupyterLab*, *Spyder*, osnove matematičke operacije
- 2 Vrste podataka, matematičke operacije, osnove nizova
- 3 Nizovi i matrice (*Numpy*)
- 4 Grananja i petlje 1 (*Flowgorithm*)
- 5 Grananja i petlje 2
- 6 Iscrtavanje 2D/3D grafova funkcija (*Matplotlib*)
- 7 Vlastite korisničke funkcije i metode
- 8 Samostalni rad na projektnom zadatku 1/3
- 9 Učitavanje i manipulacija vanjskim datotekama (*Pandas*)
- 10 Osnovne numeričke metode u Pythonu (*Scipy*, *Numpy*)
- 11 Simbolička matematika (*Sympy*)
- 12 Polinomi, prilagodba krivulja, interpolacija (*Scipy*, *Numpy*)
- 13 Osnove objektno orijentiranog programiranja (OOP) ili Osnovna deskriptivna statistika (*Statsmodels*, *Seaborn*)
- 14 Samostalni rad na projektnom zadatku 2/3
- 15 Samostalni rad na projektnom zadatku 3/3



# Zašto programiranje za građevinare?

## Uvod i motivacija

- Programiranje je danas **osnovna pismenost** suvremenog inženjera
- **Automatizacija repetitivnih proračuna i obrada velikih količina podataka**
- **Razvoj vlastitih alata** prilagođenih specifičnim potrebama projekata

Cilj ovog predmeta nije napraviti od vas profesionalne programere, već vam dati alate i vještine za **učinkovitije rješavanje inženjerskih problema.**



# Zašto Python?

## Uvod i motivacija

- **Besplatan** i otvorenog kôda
- Jednostavna sintaksa — lako se uči
- Bogat ekosustav biblioteka za inženjerstvo
- Održiva alternativa MATLAB-u
- Široko korišten u industriji i akademiji
- Izvrsna dokumentacija i zajednica



**Slika 1:** [www.python.org](http://www.python.org)

**Ključne biblioteke:** NumPy (matrice), Matplotlib (grafika), Pandas (tablice), SciPy (numeričke metode), SymPy (simbolička matematika), OpenSeesPy (analiza nosivih konstrukcija), i ostale.

**Python dokumentacija:** [docs.python.org/3.12](https://docs.python.org/3.12)

# Ciljevi predmeta

## Organizacija predmeta

### **Nakon uspješno položenog predmeta, studenti će moći:**

- ➊ Pisati Python programe za rješavanje proračunskih problema
- ➋ Koristiti različite Python biblioteke za različitu namjenu
- ➌ Vizualizirati podatke pomoću Matplotlib biblioteke
- ➍ Učitavati, obrađivati i analizirati podatke iz datoteka
- ➎ Primjenjivati naučeno na konkretne građevinske probleme
- ➏ Samostalno proširivati znanje i koristiti dokumentaciju



# Ocjenjivanje i obveze

## Organizacija predmeta

**Domaće zadaće su obavezne** — kontinuirana vježba gradiva. Zadaju se svaki tjedan na kraju nastave.

- 10 domaćih zadaća — **40%** ukupne ocjene
- 2 projektna zadatka — **60%** ukupne ocjene

Bodovi [%]	Ocjena
90.0 – 100	izvrstan (5)
75.0 – 89.9	vrlo dobar (4)
60.0 – 74.9	dobar (3)
50.0 – 59.9	dovoljan (2)
0.0 – 49.9	nedovoljan (1)

Komunikacija i predaja zadaća putem **Microsoft Teams** (kôd predmeta: **7k73bpb**).  
Github repozitorij: **[github.com/mgrubisic/RPuG2026](https://github.com/mgrubisic/RPuG2026)**



# Anaconda i Spyder

## Spyder IDE — integrirano razvojno okruženje

**Anaconda** je distribucija Pythona koja uključuje:

- Python interpreter
- Sve potrebne biblioteke (NumPy, Matplotlib, Pandas, SciPy...)
- Spyder IDE — razvojno okruženje
- Jednostavnu instalaciju bez dodatnog podešavanja

**Spyder** (Scientific Python Development Environment):

- Sličan MATLAB sučelju — lak prijelaz za korisnike MATLAB-a
- Editor kôda s isticanjem sintakse
- Interaktivna konzola (IPython)
- Variable Explorer — pregled svih varijabli
- Integrirani debugger i pomoć

**[repo.anaconda.com/archive](https://repo.anaconda.com/archive) — Anaconda3-2024.10-1-Windows-x86-64.exe**

# Sučelje Spydera

## Spyder IDE — integrirano razvojno okruženje

### Glavni dijelovi Spyder sučelja:

- 1 **Editor** (lijevo) — pisanje i uređivanje Python skripti (.py datoteke)
- 2 **IPython konzola** (dolje desno) — interaktivno izvršavanje naredbi
- 3 **Variable Explorer** (gore desno) — pregled vrijednosti varijabli
- 4 **Help** — dokumentacija i pomoć za funkcije
- 5 **Files** — pregledavanje datoteka u radnom direktoriju

### Pokretanje kôda:

- **F5** — pokreni cijelu skriptu
- **F9** — pokreni označeni dio kôda / trenutnu liniju
- **Ctrl+Enter** — pokreni trenutnu ćeliju (### označava ćeliju)

# Prvi program: Hello, World!

Spyder IDE — integrirano razvojno okruženje

Tradicionalno, prvi program u svakom programskom jeziku ispisuje pozdravnu poruku:

## Algoritam 4.1

```
1 print("Hello, World!")  
2 print("Pozdrav iz Pythona!")  
3 print("Računalno programiranje u građevinarstvu")
```

Funkcija `print()` ispisuje tekst (ili vrijednost) u konzolu. Tekst (string) se piše unutar navodnika ( `" ... "` ili `' ... '` ).



# Brojevi: cijeli i decimalni

## Osnovni tipovi podataka

Python razlikuje **cijele brojeve** ( `int` ) i **decimalne brojeve** ( `float` ):

### Algoritam 5.1

```
1 broj_katova = 5
2 broj_stupova = 12
3
4 ## Decimalni brojevi (float)
5 fck = 30.0           # [MPa] karakteristična čvrstoća betona
6 L = 6.5             # [m] raspon grede
7 E = 210000.0        # [MPa] modul elastičnosti čelika
8
9 ## Provjera tipa
10 print(type(broj_katova)) # <class 'int'>
11 print(type(fck))        # <class 'float'>
```

**Napomena:** Decimalni separator je **točka** ( `.` ), ne zarez!

# Tekst (stringovi)

## Osnovni tipovi podataka

**String** ( `str` ) je niz znakova — tekst:

### Algoritam 5.2

```
1  # Stringovi se pišu u navodnicima
2  ime_projekta = "Poslovna zgrada Osijek"
3  klasa_betona = 'C30/37'
4  opis = "Proračun AB grede prema EC2"
5
6  # Spajanje stringova
7  puni_naziv = ime_projekta + " - " + klasa_betona
8  print(puni_naziv)
9  # Ispisuje: Poslovna zgrada Osijek - C30/37
10
11 # Duljina stringa
12 print(len(klasa_betona)) # Ispisuje: 6
```

Mogu se koristiti i dvostruki ( `"..."` ) i jednostruki ( `'...'` ) navodnici.

# Logičke vrijednosti (Boolean)

## Osnovni tipovi podataka

**Boolean** ( `bool` ) ima samo dvije moguće vrijednosti: `True` ili `False` :

### Algoritam 5.3

```
1  # Logičke vrijednosti
2  je_armiran = True
3  potrebna_revizija = False
4
5  # Rezultat usporedbe je boolean
6  fck = 30.0
7  print(fck > 25)      # True (je li fck veći od 25?)
8  print(fck == 40)     # False (je li fck jednak 40?)
9  print(fck != 40)     # True (je li fck različit od 40?)
10
11 # Provjera tipa
12 print(type(je_armiran)) # <class 'bool'>
```

Logičke vrijednosti su ključne za **uvjetne naredbe** (Tjedan 3).

# Varijable i dodjela vrijednosti

## Osnovni tipovi podataka

**Varijabla** je simboličko ime koje pokazuje na neku vrijednost u memoriji:

### Algoritam 5.4

```
1  # Dodjela vrijednosti varijabli (znak =)
2  b = 0.30          # [m] širina presjeka
3  h = 0.50          # [m] visina presjeka
4  d = 0.45          # [m] statička visina
5
6  # Varijabla može promijeniti vrijednost
7  h = 0.60          # nova visina
8  d = h - 0.05      # d ovisi o h
9
10 # Višestruka dodjela
11 x, y, z = 1.0, 2.0, 3.0
12
13 # Zamjena vrijednosti
14 a, b = b, a        # Python omogućuje elegantnu zamjenu
```

# Pravila imenovanja varijabli

## Osnovni tipovi podataka

### Pravila:

- Ime može sadržavati slova, brojke i podvlaku (\_)
- Ne smije počinjati brojkom
- Python razlikuje velika i mala slova ( `Fcd`  $\neq$  `fcd` )
- Ne smiju se koristiti rezervirane riječi ( `if` , `for` , `print` ...)

### Algoritam 5.5

```
1  # Dobra imena (opisna, jasna)
2  sirina_grede = 0.30
3  broj_sipki = 4
4  fcd = 20.0
5
6  # Loša imena (nejasna, nečitljiva)
7  x1 = 0.30
8  a = 4
9  temp = 20.0
10
11 # Neispravna imena (greška!)
12 # 2nd_moment = 100    # počinje brojkom
13 # class = "C30/37"    # rezervirana riječ
```



# Osnovne matematičke operacije

## Aritmetičke operacije

### Algoritam 6.1

```
1 a = 15
2 b = 4
3
4 print(a + b)      # 19      (zbrajanje)
5 print(a - b)      # 11      (oduzimanje)
6 print(a * b)      # 60      (množenje)
7 print(a / b)      # 3.75    (dijeljenje - uvijek float!)
8 print(a // b)     # 3       (cjelobrojno dijeljenje)
9 print(a % b)      # 3       (ostatak dijeljenja - modulo)
10 print(a ** b)     # 50625   (potenciranje: 15^4)
```

**Važno:** Operator `/` uvijek vraća decimalni broj (`float`), čak i kad je rezultat cijeli broj: `10 / 2 → 5.0`

# Redoslijed operacija

## Aritmetičke operacije

Python poštuje standardni matematički redoslijed operacija:

- 1 Zgrade `()`
- 2 Potenciranje `**`
- 3 Množenje `*`, dijeljenje `/`, `//`, `%`
- 4 Zbrajanje `+`, oduzimanje `-`

### Algoritam 6.2

```
1  # Primjer: formula za moment tromosti pravokutnika
2  b = 0.30 # [m]
3  h = 0.50 # [m]
4
5  #  $I = b * h^3 / 12$ 
6  I = b * h**3 / 12
7  print(I) # 0.003125 m4
8
9  # S zagradama za jasnoću
10 I = (b * h**3) / 12
```

**Savjet:** Koristite zgrade za jasnoću, čak i kad nisu nužne!

# Matematičke funkcije — modul `math`

## Aritmetičke operacije

Za naprednije matematičke funkcije koristimo modul `math` :

### Algoritam 6.3

```
1  import math # učitavanje modula
2
3  # Konstante
4  print(math.pi)      # 3.141592653589793
5  print(math.e)       # 2.718281828459045
6
7  # Funkcije
8  print(math.sqrt(16)) # 4.0 (korijen)
9  print(math.sin(math.pi/2)) # 1.0 (sinus, argument u radijanima!)
10 print(math.cos(0))   # 1.0 (kosinus)
11 print(math.log(10))  # 2.302... (prirodni logaritam)
12 print(math.log10(100)) # 2.0 (logaritam baze 10)
13 print(math.exp(1))   # 2.718... (e^1)
14 print(math.degrees(math.pi)) # 180.0 (radijani u stupnjeve)
15 print(math.radians(90)) # 1.5707... (stupnjevi u radijane)
```

# Primjer 1: Površina i opseg pravokutnika

## Praktični primjeri iz građevinarstva

### Algoritam 7.1

```
1  # Proračun površine i opsega pravokutnog presjeka
2
3  # Ulazni podaci
4  b = 0.30    # [m] širina
5  h = 0.50    # [m] visina
6
7  # Proračun
8  A = b * h    # površina
9  O = 2 * (b + h) # opseg
10
11 # Ispis rezultata
12 print("Širina b =", b, "m")
13 print("Visina h =", h, "m")
14 print("Površina A =", A, "m2")
15 print("Opseg O =", O, "m")
```

## Primjer 2: Moment tromosti i otpora

### Praktični primjeri iz građevinarstva

#### Algoritam 7.2

```
1  # Moment tromosti i moment otpora pravokutnog presjeka
2
3  # Dimenzije presjeka
4  b = 0.30    # [m] širina
5  h = 0.50    # [m] visina
6
7  # Moment tromosti oko težišne osi
8  #  $I_y = b * h^3 / 12$ 
9   $I_y = (b * h**3) / 12$ 
10
11 # Moment otpora (za savijanje)
12 #  $W_y = I_y / (h/2) = b * h^2 / 6$ 
13  $W_y = (b * h**2) / 6$ 
14
15 # Ispis s formatiranjem
16 print("Moment tromosti  $I_y =$ ",  $I_y$ , "m4")
17 print("Moment otpora  $W_y =$ ",  $W_y$ , "m3")
18
19 # U praktičnijim jedinicama
20 print("Moment tromosti  $I_y =$ ",  $I_y * 1e8$ , "cm4")
21 print("Moment otpora  $W_y =$ ",  $W_y * 1e6$ , "cm3")
```

## Primjer 3: Normalno naprezanje u gredi

### Praktični primjeri iz građevinarstva

#### Algoritam 7.3

```
1  # Proračun normalnog naprezanja od savijanja
2
3  # Podaci
4  M = 150.0      # [kNm] moment savijanja
5  b = 0.30      # [m] širina presjeka
6  h = 0.50      # [m] visina presjeka
7
8  # Moment otpora
9  Wy = (b * h**2) / 6    # [m³]
10
11 # Maksimalno naprezanje na rubu presjeka
12 # sigma = M / Wy
13 sigma = M / Wy          # [kN/m²] = [kPa]
14
15 # Pretvorba u MPa (1 MPa = 1000 kPa)
16 sigma_MPa = sigma / 1000
17
18 print("Moment savijanja M =", M, "kNm")
19 print("Moment otpora Wy =", Wy * 1e6, "cm³")
20 print("Naprezanje sigma =", sigma, "kPa")
21 print("Naprezanje sigma =", sigma_MPa, "MPa")
```

## Primjer 4: Progib proste grede

### Praktični primjeri iz građevinarstva

#### Algoritam 7.4

```
1  # Maksimalni progib proste grede pod jednoliko
2  # raspoređenim opterećenjem:  $w_{max} = 5*q*L^4 / (384*E*I)$ 
3
4  # Podaci
5  q = 10.0          # [kN/m] opterećenje
6  L = 6.0           # [m] raspon
7  E = 30000000.0    # [kN/m²] modul elast. betona (30 GPa)
8  b = 0.30          # [m] širina
9  h = 0.50          # [m] visina
10
11 # Moment tromosti
12 I = (b * h**3) / 12
13
14 # Maksimalni progib na sredini raspona
15 w_max = (5 * q * L**4) / (384 * E * I)
16
17 # Ispis
18 print("Progib w_max =", w_max * 1000, "mm")
19 print("Granični progib L/250 =", L/250 * 1000, "mm")
```

# Zaključci

## Pregled najvažnijih aspekata ovog tjedna

### Što smo naučili:

- Zašto je programiranje važno za suvremene inženjere
- Osnovne značajke Pythona i Spyder okruženja
- Osnovni tipovi podataka: `int`, `float`, `str`, `bool`
- Varijable i pravila imenovanja
- Aritmetičke operacije i redoslijed izvođenja
- Matematičke funkcije iz modula `math`
- Praktični kratki primjeri iz područja građevinarstva

**Sljedeći tjedan:** Varijable, izrazi, unos podataka ( `input()` ), formatiranje ispisa (f-stringovi)



# Domaća zadaća #1

Rok predaje: do početka sljedećeg termina nastave

**Zadatak:** Napišite Python program koji računa sljedeće karakteristike **kružnog presjeka** promjera  $D$ :

- 1 Površinu poprečnog presjeka:  $A = \frac{\pi D^2}{4}$
- 2 Opseg:  $O = \pi D$
- 3 Moment tromosti:  $I = \frac{\pi D^4}{64}$
- 4 Moment otpora:  $W = \frac{\pi D^3}{32}$

## Zahtjevi:

- Koristite `math.pi` za vrijednost  $\pi$
- Definirajte promjer  $D = 0.40$  m
- Ispišite sve rezultate s odgovarajućim jedinicama
- Dodajte komentare koji objašnjavaju svoj Python kôd
- Rješenja je poželjno provjeriti kalkulatorom



Sveučilište Josipa Jurja Strossmayera u Osijeku  
**Građevinski i arhitektonski fakultet Osijek**  
Josip Juraj Strossmayer University of Osijek  
**Faculty of Civil Engineering and Architecture Osijek**



# Hvala na pozornosti! Pitanja?

## Marin Grubišić

**Izv. prof. dr. sc. / mag. ing. aedif.**

Sveučilište J. J. Strossmayera u Osijeku (UNIOS)

Građevinski i arhitektonski fakultet Osijek (GrAFOS)

Zavod za tehničku mehaniku (ZTM)

Predsjednik Katedre za teoriju konstrukcija

Vladimira Preloga 3, **Ured II.26**, HR–31 000 Osijek, Hrvatska

**[marin.grubisic@gfos.hr](mailto:marin.grubisic@gfos.hr) | [www.maringrubisic.com](http://www.maringrubisic.com)**

Structural &  
Earthquake Engineering