

# Python

## III. Pokročilý



# Ako Začneme?

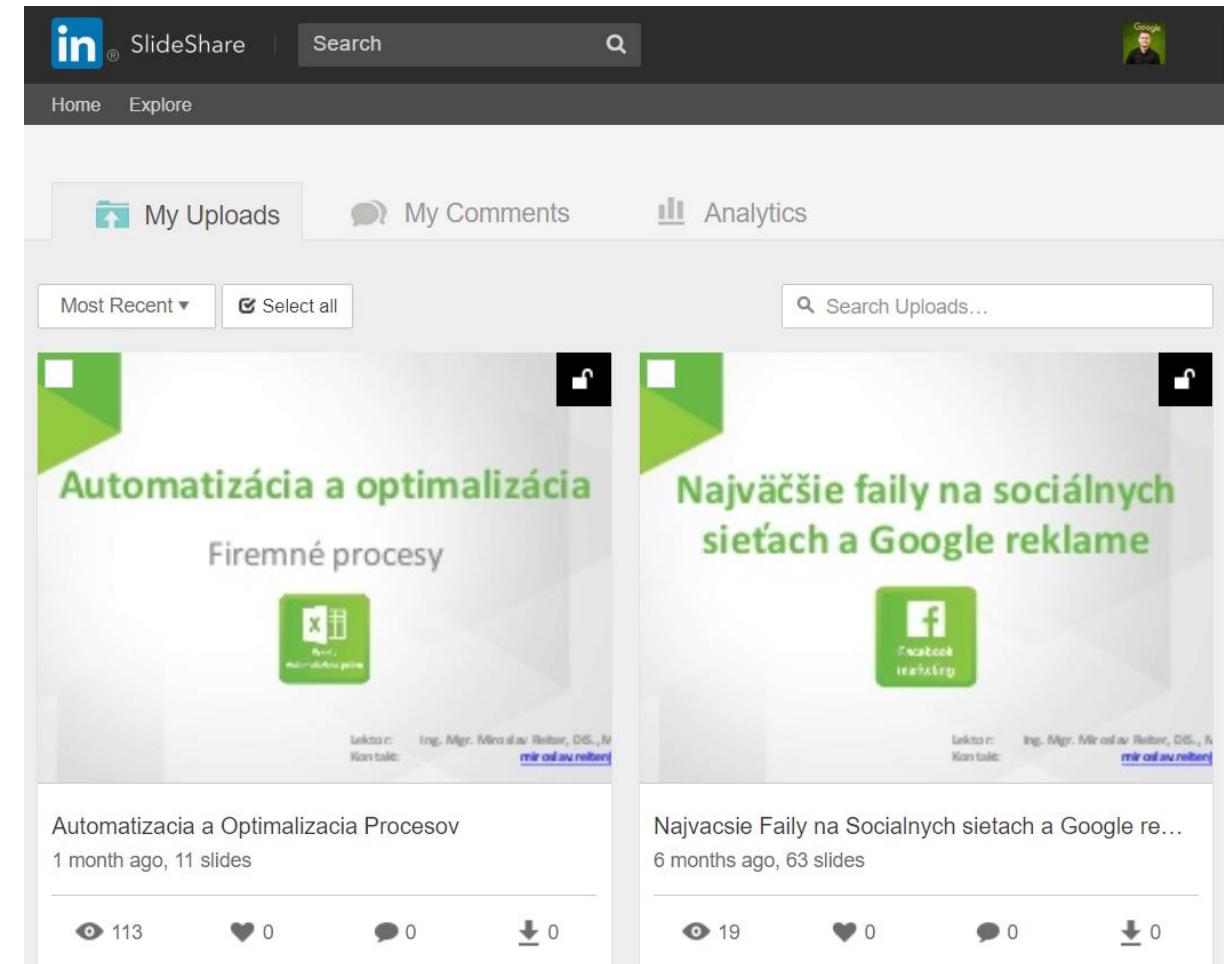
## 1. Stiahnite si Cvičný Súbor

- [https://github.com/miroslav-reiter/Kurzy\\_SAV\\_Analytika\\_Python\\_R](https://github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R)

## 2. Pridajte si ma na LinkedIn

- [www.linkedin.com/in/miroslav-reiter/](http://www.linkedin.com/in/miroslav-reiter/)

## 3. Prezentácia a materiály po prednáške



[Domov](#) » Základy programovacieho jazyku Python

## Základy programovacieho jazyku Python

Python je open-source, objektovo-orientovaný, vysoko úrovňový programovací jazyk. Python beží na mnohých variantách Unixu, na Macu a Windows (súčasťou kurzu bude inštalácia na vašom systéme). Na kurz je potrebné prísť s vlastným notebookom (s ľubovoľným operačným systémom podporujúcim Python).

### 1. Základné informácie

- História a vlastnosti Pythonu
- Python v2 vs v3
- Inštalácia Pythonu
- Python IDLE

### 2. Python syntax

### 3. Premenné

### 4. Typy dát

- String
- List
- Dictionary, Tuple, Set

### 5. Operátory

### 6. Podmienky

### 7. Cykly

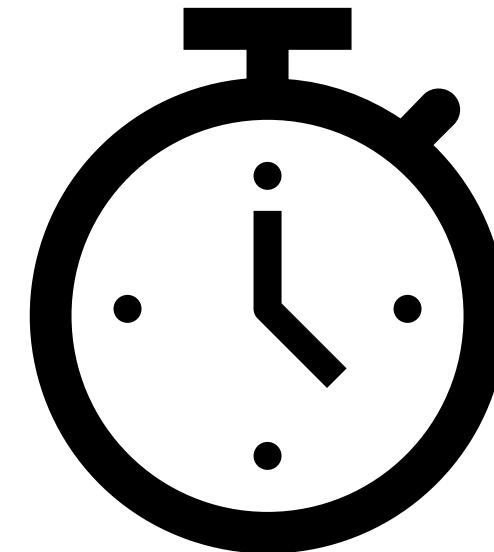
- For
- While, break, continue

### 8. Python objekty

- Class
- Module

# Úvodné informácie

- Časový rozvrh (9:00-13:30)
  - Prestávky
  - Mobilné telefóny a zariadenia
- 
- Priprav si otázky a rovno sa pýtaj
  - Interaktívna forma



# O lektorovi - Miroslav Reiter

10000+  
klientov a  
500+ firiem

Programátor  
Analytik  
Manažér

Google,  
Microsoft  
ISTQB tréner

122  
certifikácií

83 príručiek a  
publikácií

13 škôl

50+  
projektov

Vlastná firma



---

**MOTIVÁCIA**

# Študuje 5 odborov a absolvoval už 12 univerzít. Ako zvláda stres a manažovanie času?



Foto: Jakub Kovalík pre FMK UCM | Miroslav Reiter na prednáške Grow with Google na FMK UCM.

**Nikola Kotláriková**

19. júl 2022 · 8 min. čítania





# Miroslav Reiter

1. PhDr. VŠM (Podnikovný manažment)
2. Ing. STU FEI (**Aplikovaná informatika**)
3. Mgr. UK FM (**Strategický manažment a marketing**)
4. Mgr. VŠM (**Manažérstvo kvality**)
5. Mgr. VŠEMVŠ (Verejná správa)
6. Mgr. DTI (Učiteľstvo ekonomických predmetov)
7. DiS. AMOS (Cestovný ruch)
8. MBA LIGS (Executive management)
9. DBA Humanum (**IT manažment**)
10. MPA IES (Verejná správa a samospráva krajov)
11. MSc. Humanum (**Bezpečnosť informačných systémov**)
12. Ing. Paed. IGIP
13. Mgr. PEVŠ (**Bezpečnosť informačných systémov**)



DIGITÁLNA  
UNIVERZITA



FAKULTA MANAGEMENTU  
Univerzity Komenského  
v Bratislave



# Vyberte si online kurz

Naučte sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zamerajte na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia domova.

Ročné predplatné na všetky online kurzy

2299,99€

399,99€

Prístup pre Vás do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

Kúpiť teraz



407 kurzov v ponuke



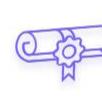
Zábavné online lekcie



Akreditované kurzy



11 rokov skúseností



Certifikovaní profesionálni lektori

Odporučame Kurzy špeciálne pre vás



Online kurz SAP I.  
Začiatočník  
224,00€ 292,50€



Online kurz Testovanie  
Softvéru I. Začiatočník  
140,00€ 183,30€



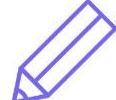
Online kurz Java I.  
Začiatočník  
67,00€ 88,40€



Online kurz PRINCE2  
Foundation  
224,00€ 308,70€



Online kurz Lektor  
(Akreditovaný Kurz Lektor)  
127,00€ 167,50€



Online kurz Copywriting I.  
Začiatočník  
50,00€ 66,30€

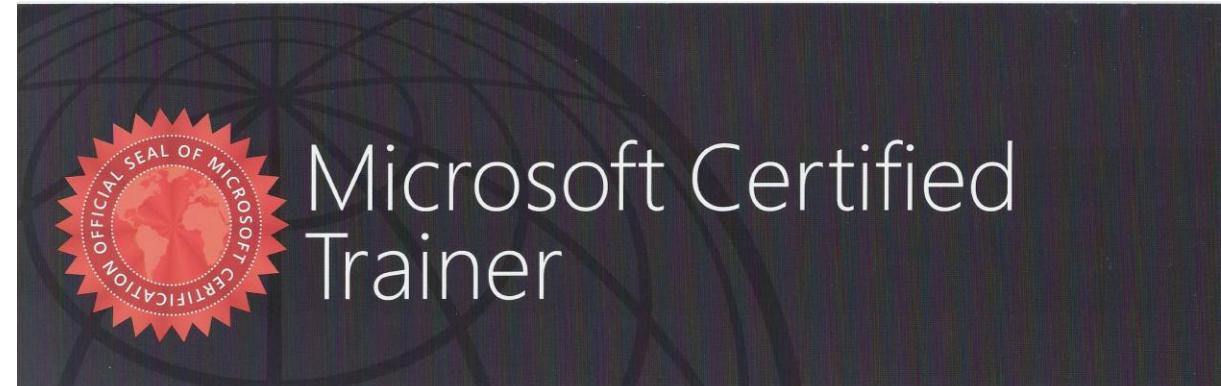
# Miroslav Reiter

získava status

## Google Certified Trainer

Automation

Google



Microsoft Certified  
Trainer

MIROSLAV REITER

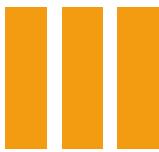
Has successfully completed the requirements to be recognized as a Microsoft Certified Trainer

N. S.  
Satya Nadella  
Chief Executive Officer

Microsoft  
CERTIFIED  
Trainer



Luigi, Mário  
a Yoshi



# Čo robíte?

1. Študent/učiteľ

2. Zamestnanec

3. Podnikateľ

4. Nezamestnaný/materská

5. Dievča pre všetko



National competence centre for high performance computing  
SLOVAKIA

C  
EURO



**EuroHPC**  
Joint Undertaking





### Vzdelávanie

Kurzy:  
[itkurzy.sav.sk](https://itkurzy.sav.sk)



### Propagácia

Prednášky:  
[https://eurocc.nscc.sk  
/news/prednasky/](https://eurocc.nscc.sk/news/prednasky/)



### HPC služby

Prístup k  
výpočtovým  
prostriedkom



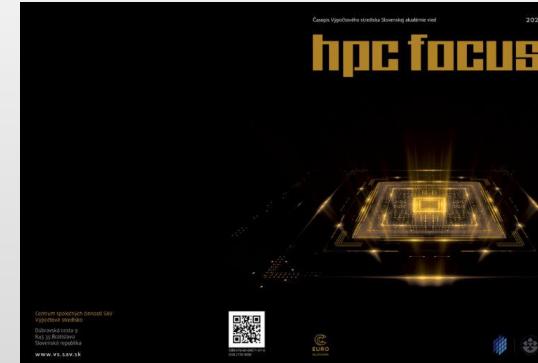
### Mapovanie HPC prostredia

Prieskum:  
[https://eurocc.nscc.sk  
/mapping-survey/](https://eurocc.nscc.sk/mapping-survey/)



### Spolupráca

Pilotné projekty  
Dlhodobá  
spolupráca



**Qubit**  
Conference

robíme it

Slovenská  
obchodná  
a priemyselná  
komora

### S kým spolupracujeme:

- Akademické inštitúcie, univerzity,  
ústavy SAV,...
- Verejná správa
- Súkromné firmy, tretí sektor

### Naučte sa pracovať v prostredí HPC systémov:

Najbližší kurz:

[HPC infraštruktúra](#) / 18. máj 2022

### Hľadáme nových kolegov do tímu!

<https://eurocc.nscc.sk/career/>



**Sledujte nás na sociálnych sietiach:**



# Interaktívna prednáška

## Aktívne používanie a zapájanie sa

Participants (20)

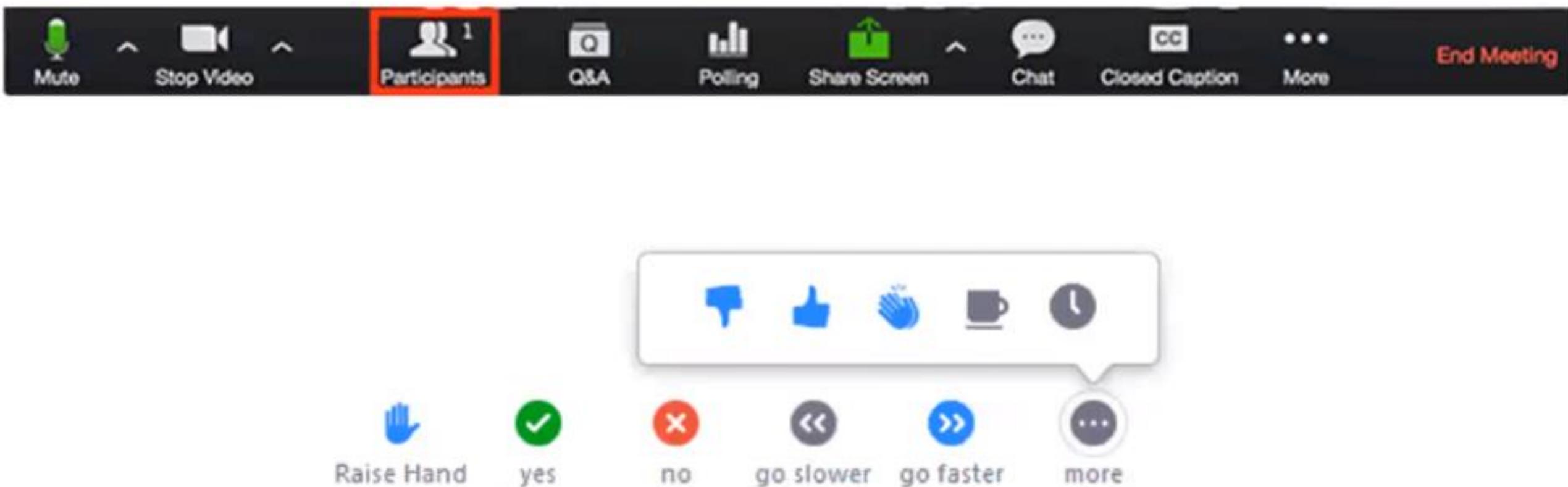
Find a participant

Participant	Microphone	Video	Hand Raised	Feedback
Miroslav Reiter (Me)	✓	🔇	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ
	ⓘ	ⓘ	ⓘ	ⓘ

Raise Hand      yes      no      go slower      go faster      more

Invite      Mute Me

# Používame Zoom



# Vaše Ciele a Očakávania

1. Doplniť si znalosti z jazyka Python

2. Zlepšiť objektové myslenie

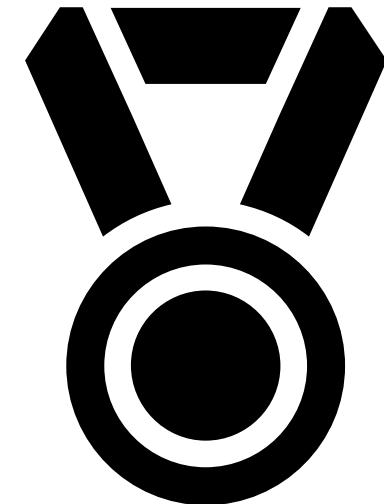
3. Zlepšiť OO programovanie

4. Doplniť si znalosti z programovania

5. Doplniť si znalosti z vývojových prostredí (IDE)

6. Využitie AI a RE pri programovaní

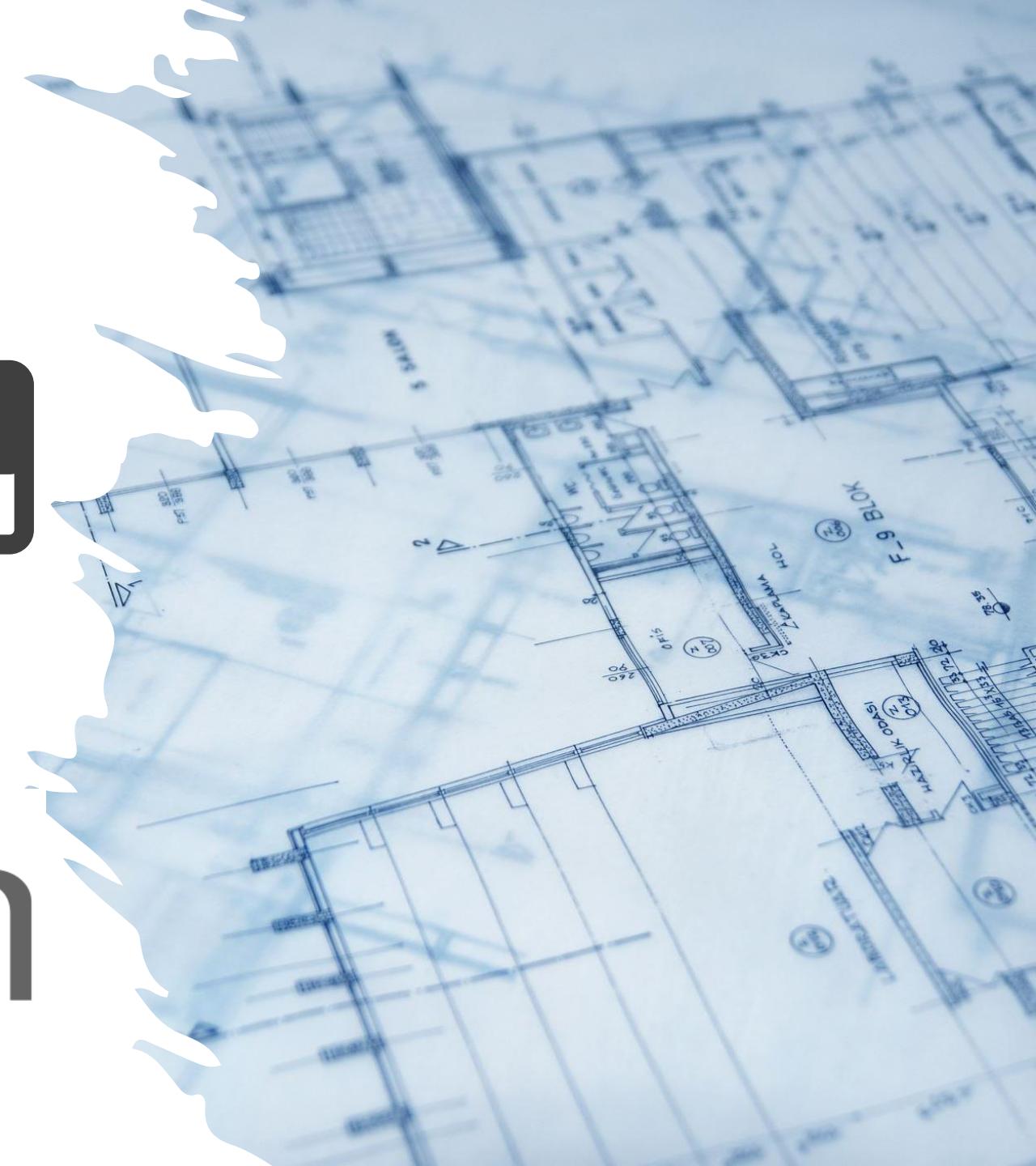
Zábava je v zaručená v každom bode :-)



# Triedy (Classes)



# python

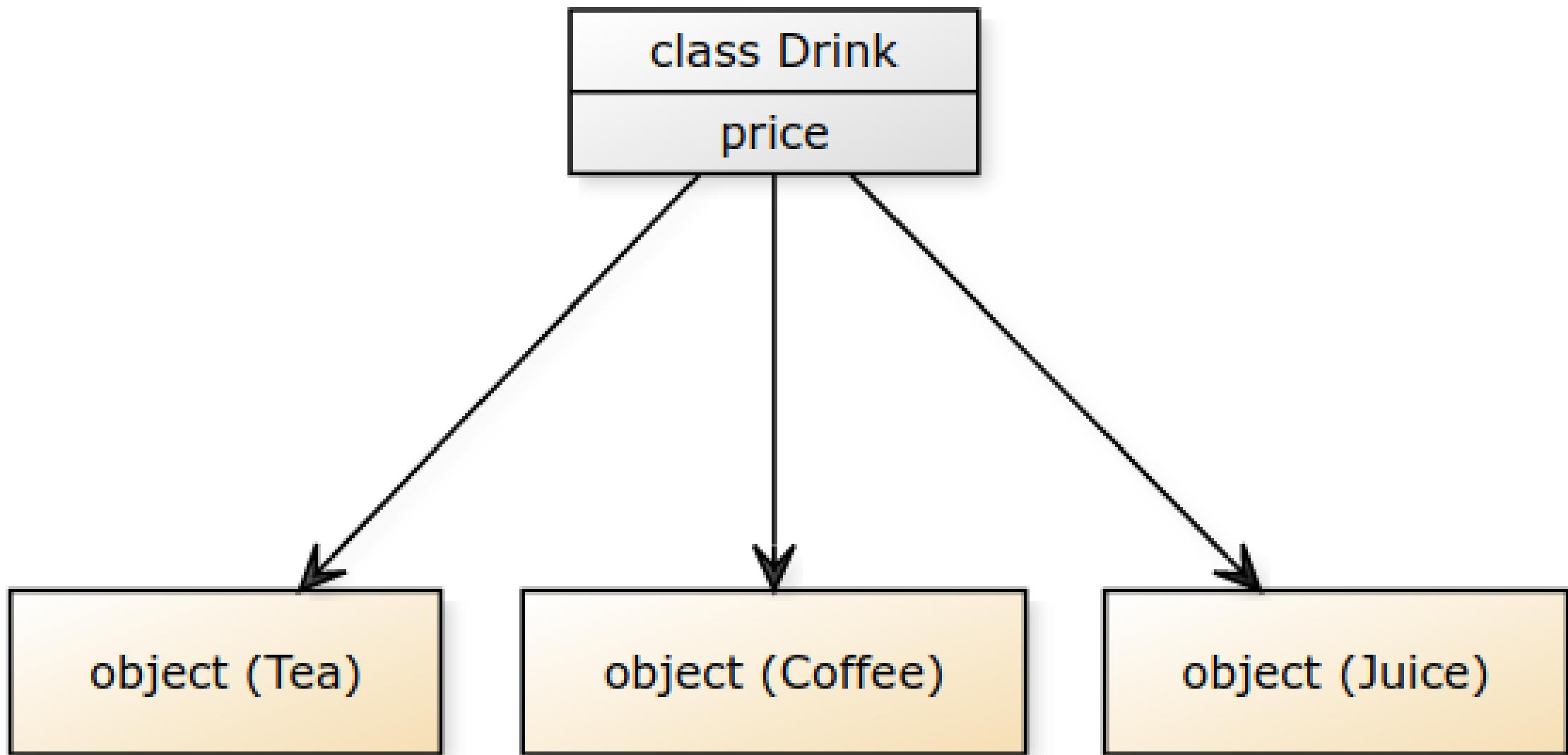


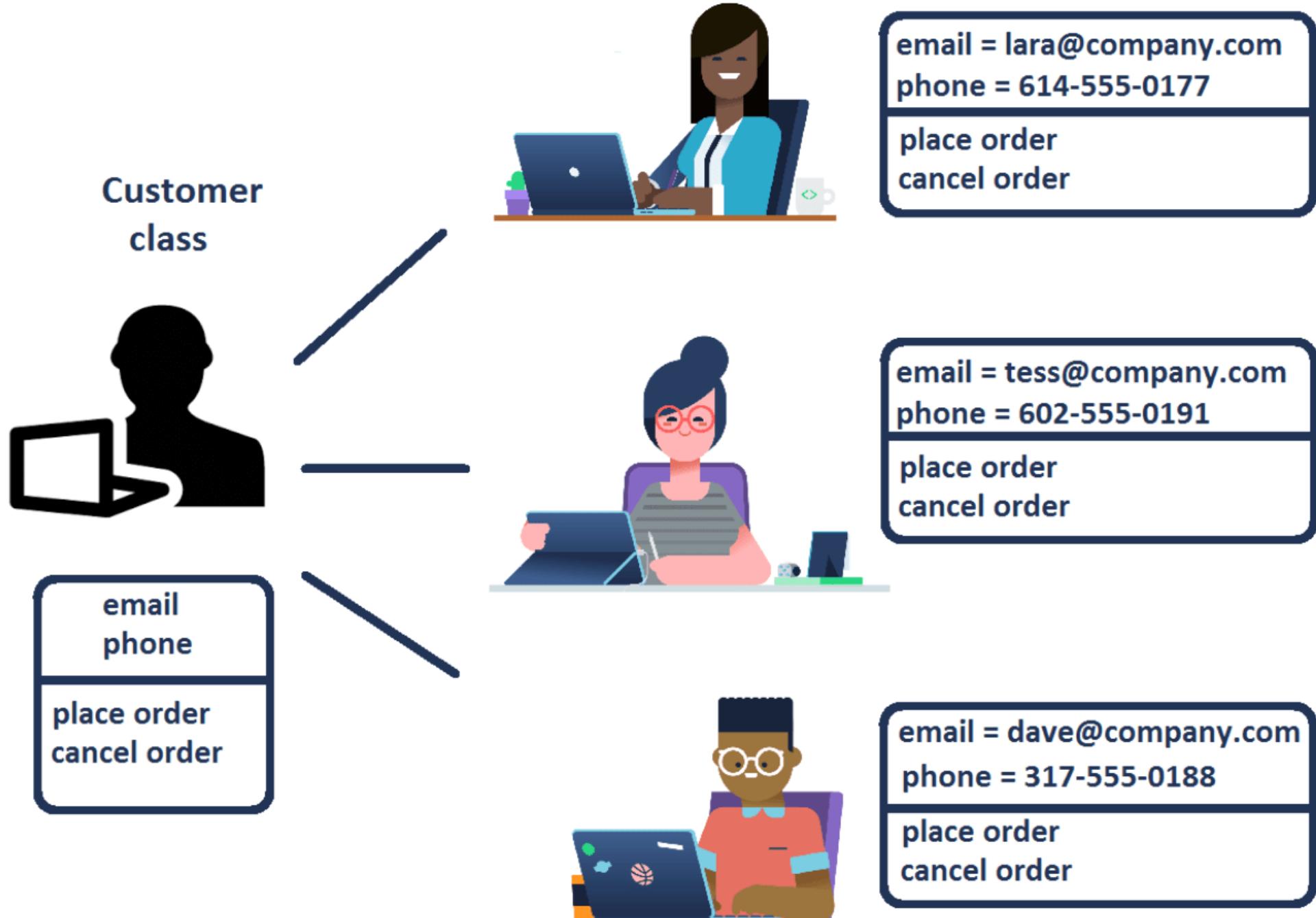
# Metódy triedy

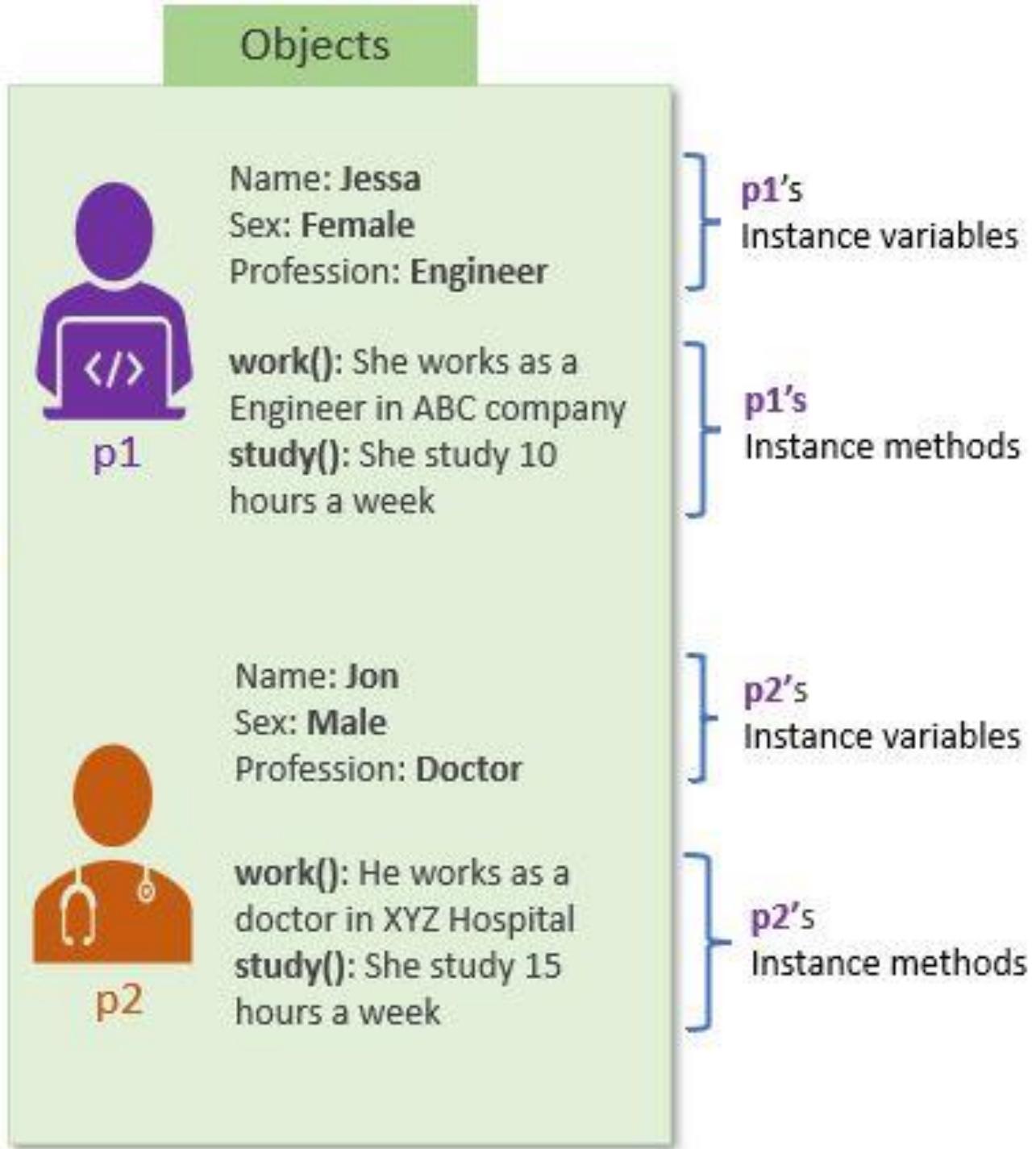
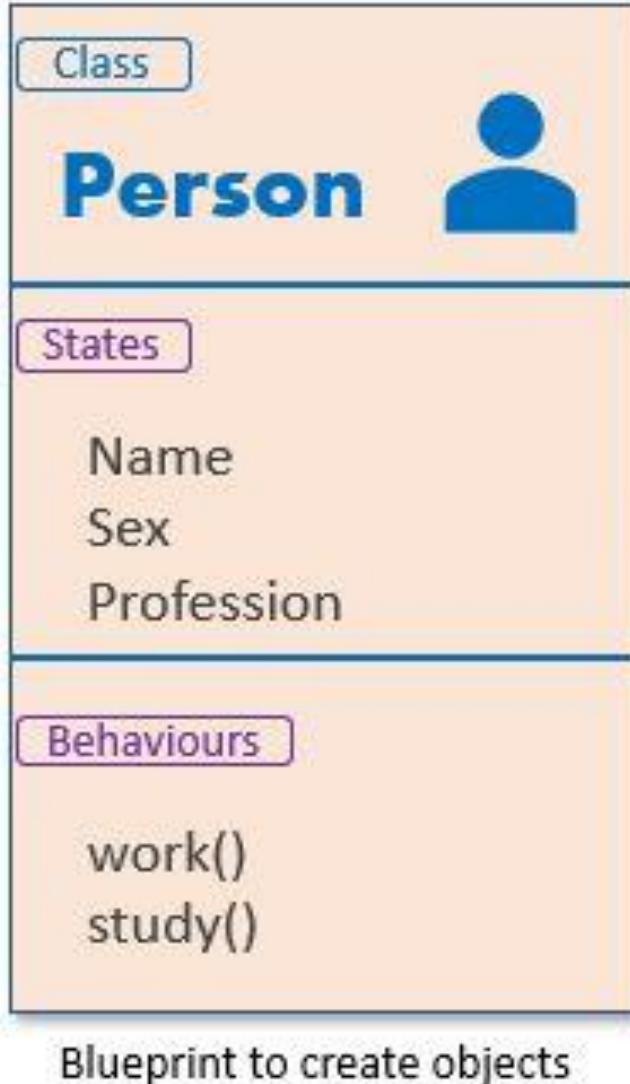
- Verejné, chránené (\_), súkromné (\_)
- Funkcie definované v triede
- Definícia:

```
def metoda(self, parametre ):  
    self.atribut1 = ...  
  
    self.metoda(parametre)  
...
```
- Každá metóda musí mať pri definovaní ako prvý parameter self
- Self je premenná, ktorý reprezentuje samotnú inštanciu (objekt) danej triedy

1. # Vytvorenie instance triedy
2. Objekt = Nazov\_Triedy(...)
3. # Vytvorenie noveho atributu/zmena hodnoty
4. Objekt.atribut = hodnota
5. # Zavolanie metoda
6. Objekt.metoda(parametre)
7. # Zrušenie/vymazanie danej triedy
8. Del objekt







# Vyber čo je trieda a čo je objekt?

1. Adam Šangala
2. Softvér na tvorbu hmatových orientačných máp
3. Formulár
4. Príručka žiadateľa o grant
5. Trieda
6. Projekt
7. Seat Ibiza
8. Objekt
9. Operačný program Efektívna verejná správa
10. Žena

# Objektovo Orientované Programovanie

1. **Zapúzdrenie (encapsulation)** - Privilegovaný selektívny prístup k dátam daného objektu triedy, čím umožňuje **ochranu dát objektu**
2. **Dedičnosť (inheritance)** - Umožňuje vytvárať **nové triedy z už existujúcich tried** s prípadnou modifikáciou ich vlastností
3. **Mnohotvárnosť (polymorphisms)** - Umožňuje **rôzne správanie sa metód** s rovnakým názvom triedy alebo pri dedení triedy



# Kedy OOP a Kedy Nie?

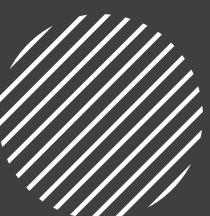


python





# Vyber čo je trieda a čo je atribút?



Výška

Tvar

Celková  
zazmluvnená  
suma

Poskytovateľ  
pomoci

Popis  
projektu

Miesto  
realizácie

IČO

Merateľné  
ukazovatele

Dátum  
platnosti  
zmluvy

Operačný  
program

DPH

Formulár

# Triedy a Objekty

## Trieda



```
class Pes:  
    meno = ""
```

```
def stekaj(self):  
    print("Haf")
```

## Objekty (inštancie triedy Pes)



```
pes1 = Pes()  
pes1.meno = "Luigi"
```



```
pes2 = Pes()  
pes2.meno = "Mario"
```

OBJECT	CLASS
Object is an instance of a class.	Class is a blue print from which objects are created
Object is a real world entity such as chair, pen, table, laptop etc.	Class is a group of similar objects.
Object is a physical entity.	Class is a logical entity.
Object is created many times as per requirement.	Class is declared once.
Object allocates memory when it is created.	Class doesn't allocate memory when it is created.
Object is created through new keyword. ob = Employee()	Class is declared using class keyword. class Employee

# Objekt

- Vyjadritelný podstatným meno
- Všetko v Pythone je objekt nejakého typu
- Má svoje idčko
- Zabudované metódy  
method

```
>>> type("Adam Sangala")
<type 'str'>
>>> dir(str)
['__add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
 '__getslice__', '__gt__', '__hash__', '__init__', '__le__',
 '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__',
 '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
 '__sizeof__', '__str__', '__subclasshook__', '__formatter_file_name_split',
 '__formatter_parser__', 'capitalize', 'center', 'count',
 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index',
 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle',
 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace',
 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split',
 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate',
 'upper', 'zfill']

>>> dir("Adam Sangala")
['__add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
 '__getslice__', '__gt__', '__hash__', '__init__', '__le__',
 '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__',
 '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
 '__sizeof__', '__str__', '__subclasshook__', '__formatter_file_name_split',
 '__formatter_parser__', 'capitalize', 'center', 'count',
 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index',
 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle',
 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace',
 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split',
 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate',
 'upper', 'zfill']
```



r/Showerthoughts

u/Stormfly • 1h

The sentence "Don't objectify women" has "women" as the object of the sentence.

Funny

Mindblowing



Vote



33



Share



BEST COMMENTS ▾

Azzarel • 1h

Woman w = new Woman(); where is your god now?



Reply



58



Super Class (Parent)



Student

"Is a" Relationship

[Student is a kind of Person]

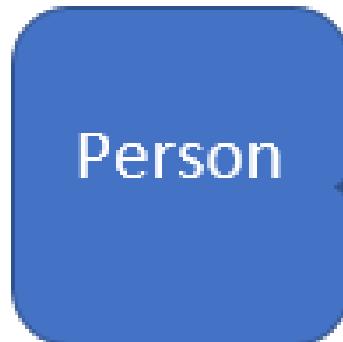
Sub Class (Child)

Inheritance

"Has a"  
Relationship



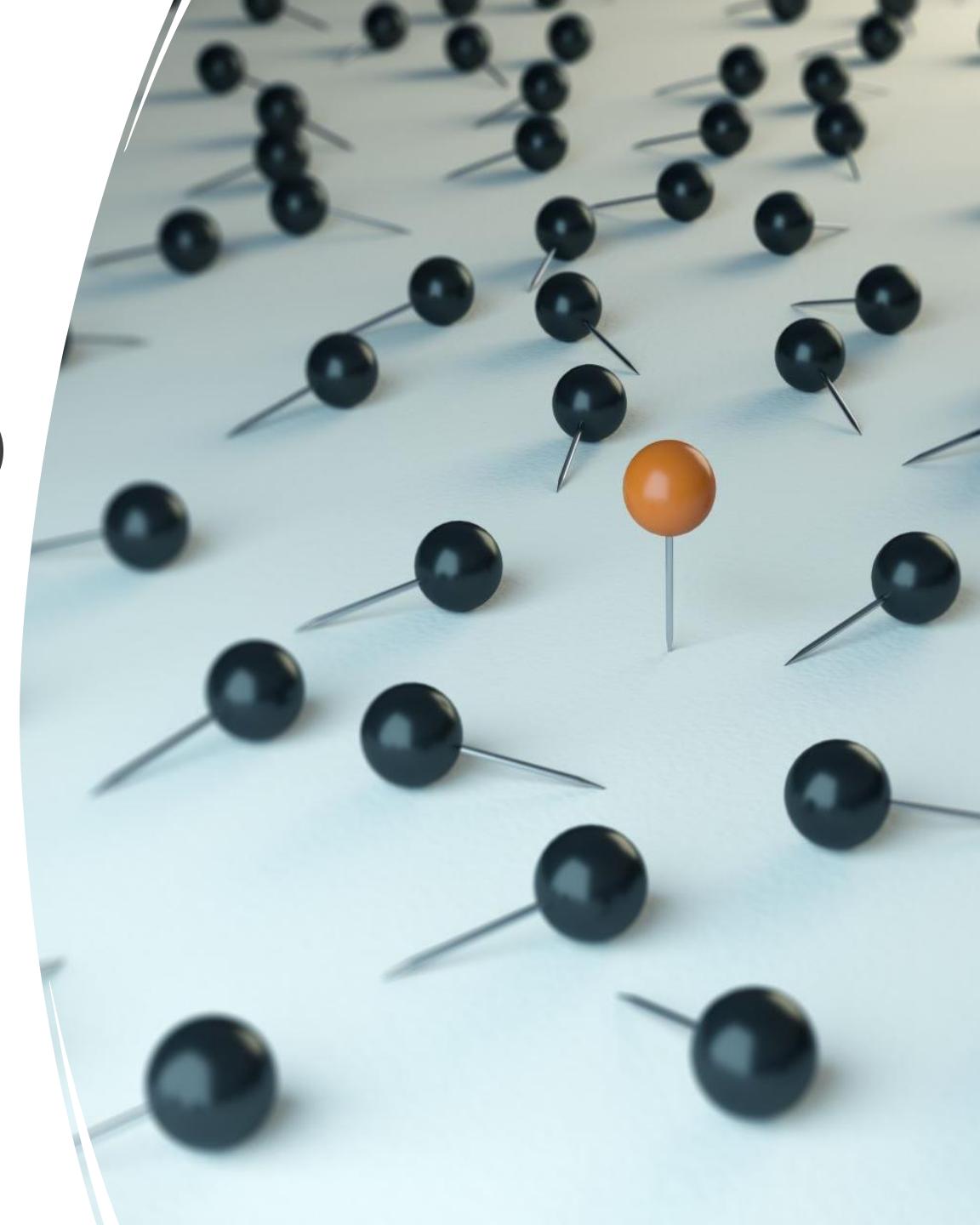
Composition



Aggregation

# Príklady Triedy a OOP

 python



Customer  
class



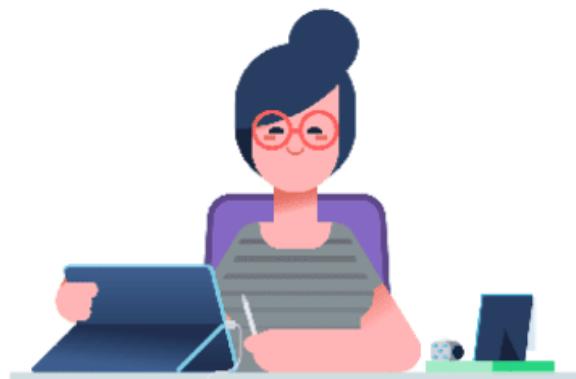
email  
phone

place order  
cancel order



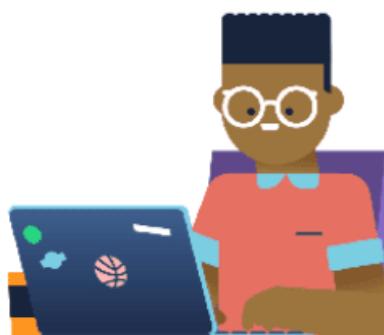
email = lara@company.com  
phone = 614-555-0177

place order  
cancel order



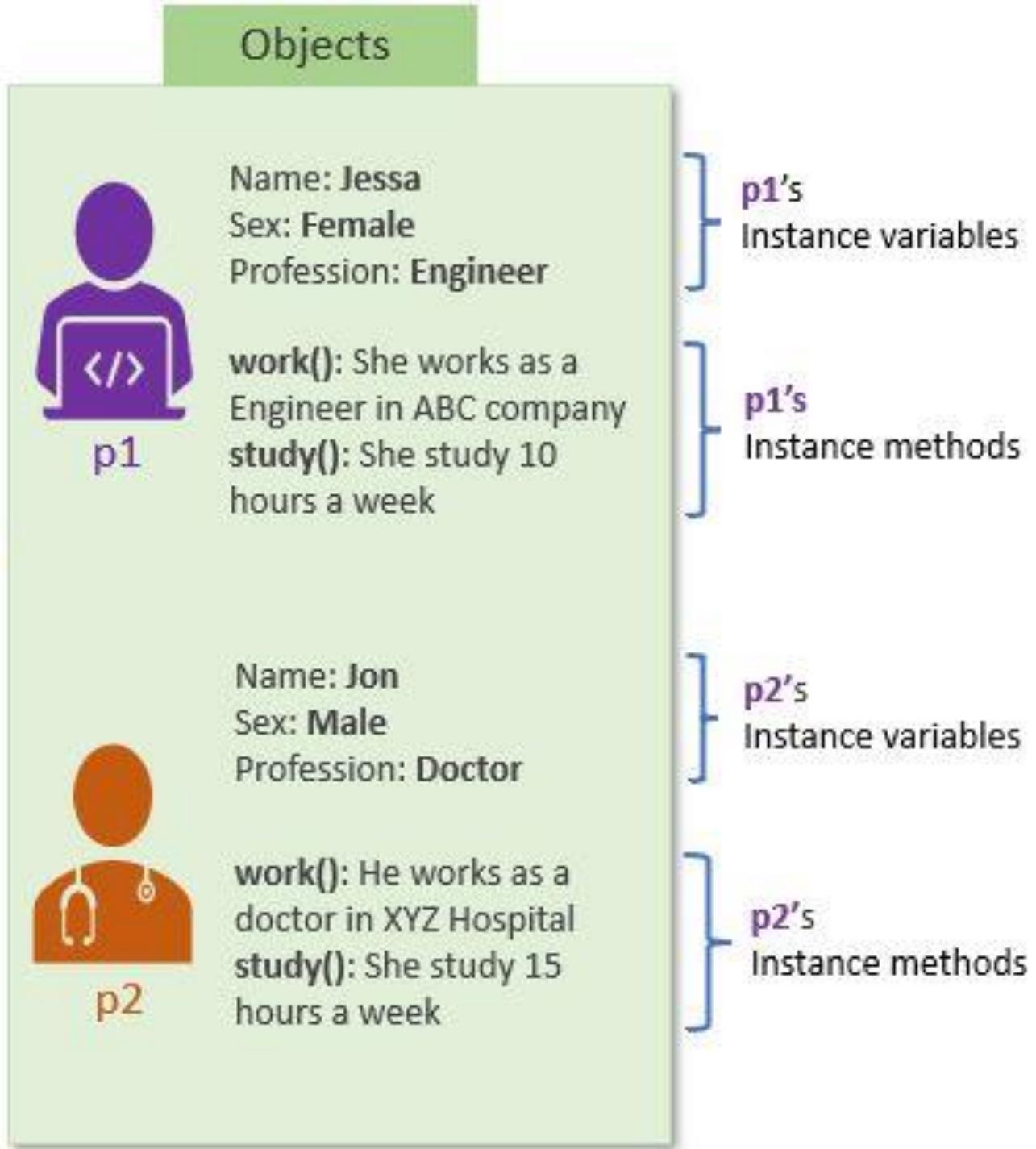
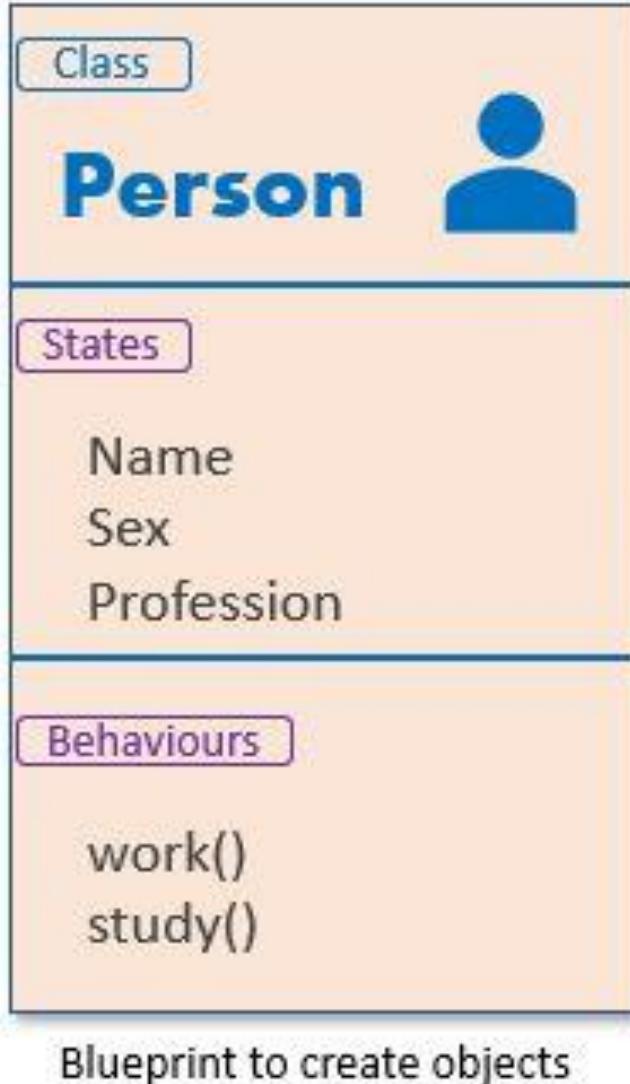
email = tess@company.com  
phone = 602-555-0191

place order  
cancel order



email = dave@company.com  
phone = 317-555-0188

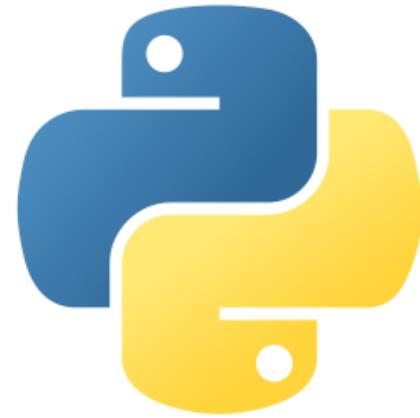
place order  
cancel order





# Regulárne Výrazy Regexy

---



python

# Čo sú Regulárne Výrazy?

Retázec popisujúci celú množinu reťazcov,  
konkrétny regulárny jazyk

```
s/^𓁃𓁃*𓁄-𓁅*(&𓁆??𓁃)𓁃$/\${1}/mg
```



geek and poke

*ANCIENT EGYPTIAN REGEXP*

regexp | regex | re

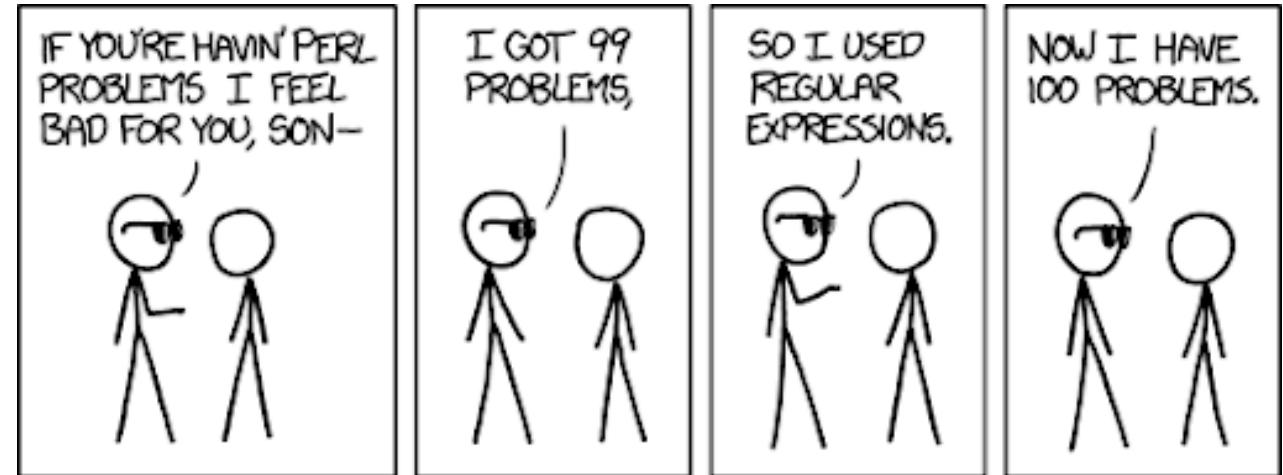
# Načo je to dobré?

## Vyhľadávanie textu

- Zistenie, či vstupný text vyhovuje zadanému regulárному výrazu
- Zistenie pozície vo vstupnom teste, kde sa nachádza zhoda s regulárnym výrazom

## Manipulácia s textom

- Zámena resp. zmena textu
- Výber všetkých zhodných výskytov so zadaným regulárnym výrazom



# Definované triedy alebo množiny znakov

Trieda znakov	Význam
\w	Písmeno alebo číslica z množiny (a-zA-Z_0-9)
\W	Iný znak ako písmeno alebo číslica
\s	"biely znak" (Whitespace) čiže znak, ktorý v informatike predstavuje biele miesto, jednoducho nie je priamo viditeľný. Príkladom takéhoto znaku môže byť napr. znak medzery, znak tabulátora alebo prípadne iného riadiaceho znaku z <a href="#">ASCII</a> . medzera a \f \n\ \r \t \v viď. nižšie v tabuľke
\S	Iný ako tzv. "biely znak" (Non whitespaces)
\d	Číslica (Decimal Digit) čiže znaky 0 až 9
\D	Iný znak ako číslica (Non decimal digit)
\t	znak tabulátora
\r	Carriage Return je špeciálnym riadiacim znakom slúžiacim na návrat kurzora na začiatok riadku
\v	Vertical Tab
\f	Form Feed
\n	Nový riadok (New Line)
\e	Escape

## Regular expression boundary

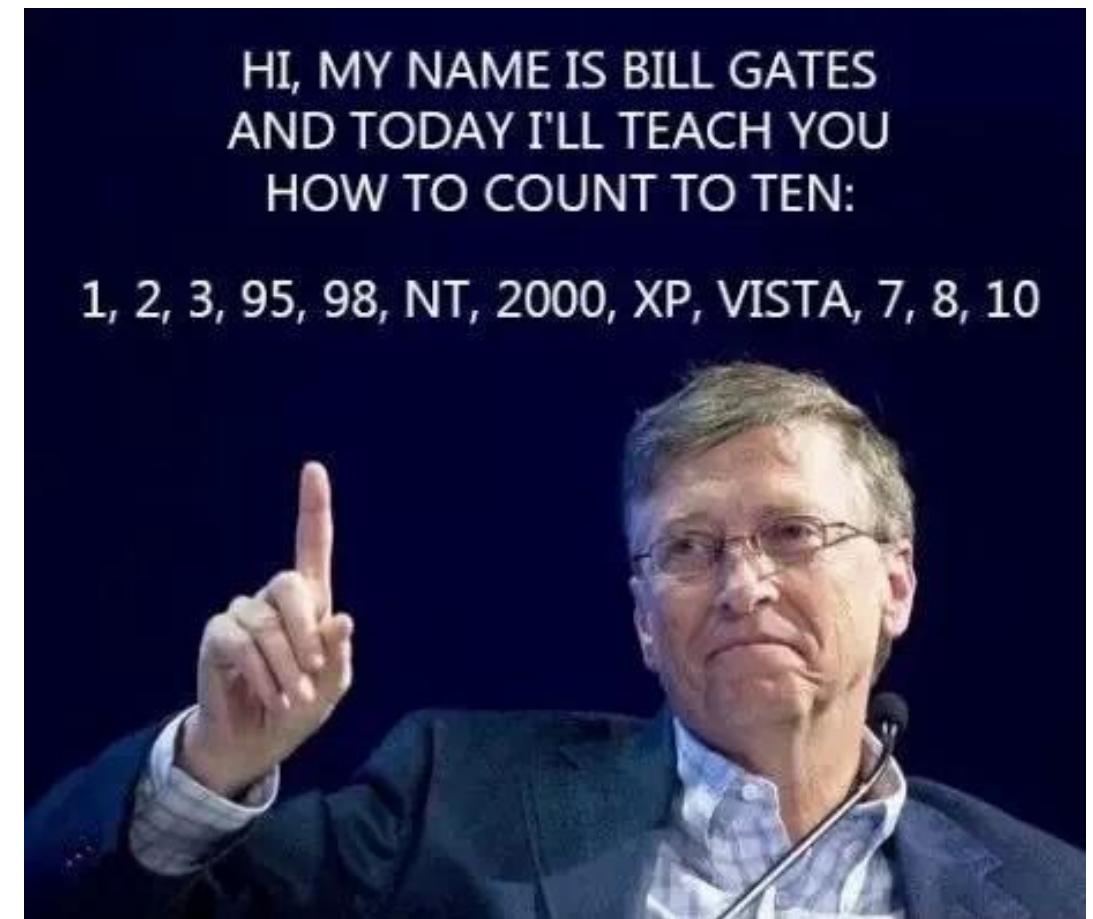
The diagram illustrates a regular expression boundary with the following annotations:

- match anything contained within brackets**: Points to the first pair of brackets `(\w._%+-)`.
- as many times as possible**: Points to the quantifier `+`  after the first bracket.
- match the @ symbol**: Points to the character `@`.
- match upper and lower case A through Z**: Points to the character class `[a-zA-Z]`.
- match at least two times but no more than four times**: Points to the quantifier `{2,4}`.
- match the . symbol**: Points to the character `.`.
- match any .,\_%+-**: Points to the character class `(\w._%+-)`.
- match any character A-Z upper or lower case and any number 0 to 9**: Points to the character class `(\w)`.

The regular expression itself is: `/[\w._%+-]+@[\\w.-]+\.\[a-zA-Z]{2,4}/`

# Hľadáme slovo Bill

- Regulárny výraz
- Bii?ll?.
- Tento výraz nájde Bill, Biil, Bil a Biill



# Príklady regulárnych výrazov

- **A(d|l)am** reťazec "Adam" a "Alam"
- **Ba\*f** reťazce "Bf", "Baf", "BAAF", "Baaaf"
- **\d{3} \d{2}** Postupnosť 3 číslic, medzera a 2 číslice PSČ
- **<[^>]\*>** Tag v jazyku HTML - ľubovoľný text uzavretý medzi špicaté zátvorky (poznámka: tento výraz je zjednodušený, v skutočnom HTML situácii komplikujú reťazca v úvodzovkách)
- **[0-9a-fA-F]+(, ?[0-9a-fA-F]+)\*** Zoznam hexadecimálnych čísel, oddelených čiarkami a nepovinnými medzerami

## Table of Contents

[re — Regular expression operations](#)

- [Regular Expression Syntax](#)
- [Module Contents](#)
  - [Flags](#)
  - [Functions](#)
  - [Exceptions](#)
- [Regular Expression Objects](#)
- [Match Objects](#)
- [Regular Expression Examples](#)
  - [Checking for a Pair](#)
  - [Simulating scanf\(\)](#)
  - [search\(\) vs. match\(\)](#)
  - [Making a Phonebook](#)
  - [Text Munging](#)
  - [Finding all Adverbs](#)
  - [Finding all Adverbs and their Positions](#)
  - [Raw String Notation](#)

# re — Regular expression operations

**Source code:** [Lib/re/](#)

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings ([str](#)) as well as 8-bit strings ([bytes](#)). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

Regular expressions use the backslash character ('\\') to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python's usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write '\\\\\\' as the pattern string, because the regular expression must be \\, and each backslash must be expressed as \\ inside a regular Python string literal. Also, please note that any invalid escape sequences in Python's usage of the backslash in string literals now generate a [SyntaxWarning](#) and in the future this will become a [SyntaxError](#). This behaviour will happen even if it is a valid escape sequence for a regular expression.

The solution is to use Python's raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with 'r'. So `r"\n"` is a two-character string containing '\\' and '\n', while "\n" is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.

# RegexBuddy



RegexBuddy

Java 8   Case sensitive   Exact spacing   Dot doesn't match line breaks   ^\$ don't match at line breaks   Default line breaks   Reset

Match   Replace   Split   Copy   Paste   History

"[^\\n]\*(?:\\n[^\\n]\*)\*"   Programming: String  
Programming: String (escape quotes)  
Programming: String (multiline; escape quotes)

Create   Convert   Debug   Use   Library   Test   GREP   Forum

Detailed   Explain Token   Insert Token   Compare Java 13–14   Whole file   LF only

Export   RegExMagic

Java 8 & Java 13–14

- All selected applications handle your regular expression in the same way
- A Match the character "" literally**
- I Match any single character NOT present in the list below**
  - Between zero and unlimited times, as many times as possible, giving backslash character
  - The literal character ""
  - The backslash character
- G Match the regular expression below**
  - Between zero and unlimited times, as many times as possible, giving backslash character
  - Match the backslash character
  - Match any single character that is NOT a line break character (line feed or carriage return)
  - I Match any single character NOT present in the list below**
    - Between zero and unlimited times, as many times as possible, giving backslash character
    - The literal character ""
    - The backslash character
- A Match the character "" literally**

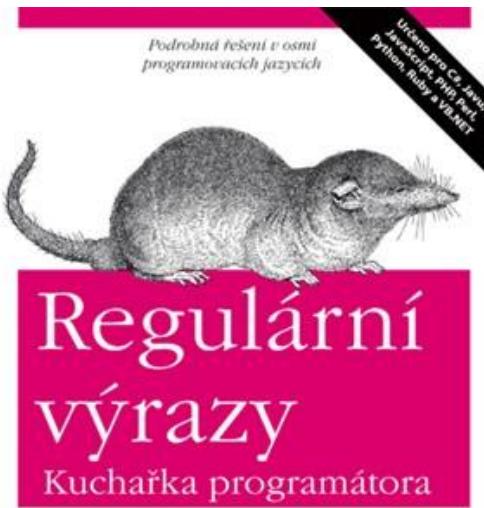
"string"  
"two""strings"  
"multi"  
"line"  
"string"  
"string with \"quotes\""  
"multi"  
"line"  
"string"  
"with"  
"\\"quotes\""

Start Length

Match 1 of 6: "string" 0 8

# RegexBuddy

- Jan Goyvaerts
- Steven Levithan



A screenshot of the RegexBuddy application. The main window has a green header bar with the title 'RegexBuddy' and various toolbars. The main area shows a search input field containing the regex pattern '[^\r\n]\*'. Below the input is a toolbar with icons for Create, Convert, Test, Debug, Use, Library, GREP, and Forum. A dropdown menu shows 'RegexBuddy4.rbl' and a 'Read Only' checkbox. To the right, there's a 'Programming: String (multiline; escape quotes)' section with a note about quotes appearing in strings. The bottom half of the window displays a code editor with several lines of Java code, with the regex pattern highlighted in blue. The code includes strings like "string", "two", "strings", and "multi-line-string".

Untitled Pattern Save (ctrl-s) New

by gskinner GitHub Sign In

Menu

Pattern Settings

My Patterns

Cheatsheet

RegEx Reference

Community Patterns

Help

## Expression

/([A-Z])\w+/g

Text Tests NEW

29 matches (0.9ms)

RegExr was created by gskinner.com, and is proudly hosted by Media Temple.

Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & JavaScript flavors of RegEx are supported. Validate your expression with Tests mode.

The sidebar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and view patterns you create or favorite in My Patterns.

Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.

RegExr is an online tool to **learn, build, & test** Regular Expressions (RegEx / RegExp).

- Supports **JavaScript & PHP/PCRE** RegEx.
- Results update in **real-time** as you type.
- **Roll over** a match or expression for details.
- Validate patterns with suites of **Tests**.
- **Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegEx Reference** with help & examples.
- **Undo & Redo** with ctrl-Z / Y in editors.
- Search for & rate **Community Patterns**.

### Tools

Replace List Details Explain

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

**Capturing group #1.** Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

**Character set.** Match any character in the set.

**A-Z Range.** Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

**)**

**\w Word.** Matches any word character (alphanumeric & underscore).

**+** **Quantifier.** Match 1 or more of the preceding token.

Want to support RegExr? Consider disabling your ad-blocker for this domain. We'll show a non-intrusive, dev-oriented ad in this area.



## SAVE &amp; SHARE

[Save Regex](#) ctrl+s

## FLAVOR



&lt;/&gt; PCRE2 (PHP &gt;=7.3)

&lt;/&gt; PCRE (PHP &lt;7.3)

&lt;/&gt; ECMAScript (JavaScr...)

&lt;/&gt; Python

&lt;/&gt; Golang

&lt;/&gt; Java 8



&lt;/&gt; .NET (C#)

## FUNCTION

&gt;\_ Match



☒ Substitution

☒ List

☒ Unit Tests

## TOOLS

[Code Examples](#)**DOPPLER**

All your environment variables, in one place

If you're running an ad blocker, consider whitelisting regex101 to support the website. [Read more.](#)

## REGULAR EXPRESSION

no match

" insert your regular expression here

' gm



## TEST STRING

insert your test string here

## EXPLANATION

An explanation of your regex will be automatically generated as you type.

## MATCH INFORMATION

Detailed match information will be displayed here automatically.

## QUICK REFERENCE

Search reference

☒ All Tokens

★ Common Tokens



○ General Tokens

☒ Anchors

☒ Meta Sequences

A single character of: a, b or c

[abc]

A single character of: a, b, c o... [[ab][cd]]

[ ]

A character except: a, b or c

[^abc]

A character in the range: a-z

[a-z]

A character not in the range: a-z

[^a-z]

A character in the range: a-z or ... [a-zA-Z]

[ ]

**Formatters**

- JSON Formatter
- HTML Formatter
- XML Formatter
- SQL Formatter

**Validators**

- JSON Validator
- HTML Validator
- XML Validator - XSD
- XPath Tester
- Credit Card Number Generator & Validator
- Regular Expression Tester (RegEx)**
- Java Regular Expression Tester (RegEx)
- Cron Expression Generator - Quartz

**Encoders & Decoders**

- Url Encoder & Decoder
- Base 64 Encoder & Decoder
- Convert File Encoding
- QR Code Generator

**Code Minifiers / Beautifier**

- JavaScript Beautifier
- CSS Beautifier
- JavaScript Minifier
- CSS Minifier

**Converters**

- XSD Generator
- XSLT (XSL Transformer)
- XML to JSON Converter
- JSON to XML Converter
- CSV to XML Converter
- CSV to JSON Converter
- Epoch Timestamp To Date

**Cryptography & Security**

## Regular Expression Tester

This free regular expression tester lets you test your regular expressions against any entry of your choice and clearly highlights all matches. It is JavaScript based and uses [XRegExp library](#) for enhanced features.

Consult the [regular expression documentation](#) or the [regular expression solutions to common problems](#) section of this page for examples. If you need more examples or solutions, please [contact me](#).

**Regular expression :****Entry to test against :****Replace with (Optional):**

You can make use of \$1, \$2, \$3 and so on if you are using parenthesis groups in your regular expression. \t \n \r are supported.

**Flags:**

- i - Case-insensitive
- m - Multiline
- g - Global (don't stop at first match)
- s - Dot matches all INCLUDING line breaks (XRegExp only).

**TEST MATCH****REPLACE**

## Regular Expression - Documentation

### Metacharacters

**Formatters**[JSON Formatter](#)  
[HTML Formatter](#)  
[XML Formatter](#)  
[SQL Formatter](#)**Validators**[JSON Validator](#)  
[HTML Validator](#)  
[XML Validator - XSD](#)  
[XPath Tester](#)  
[Credit Card Number Generator & Validator](#)  
[Regular Expression Tester \(RegEx\)](#)  
**Java Regular Expression Tester (RegEx)**  
[Cron Expression Generator - Quartz](#)**Encoders & Decoders**[Url Encoder & Decoder](#)  
[Base 64 Encoder & Decoder](#)  
[Convert File Encoding](#)  
[QR Code Generator](#)**Code Minifiers / Beautifier**[JavaScript Beautifier](#)  
[CSS Beautifier](#)  
[JavaScript Minifier](#)  
[CSS Minifier](#)**Converters**[XSD Generator](#)  
[XSLT \(XSL Transformer\)](#)  
[XML to JSON Converter](#)  
[JSON to XML Converter](#)  
[CSV to XML Converter](#)  
[CSV to JSON Converter](#)  
[Epoch Timestamp To Date](#)**Cryptography & Security**

# Java Regular Expression Tester

This [free Java regular expression tester](#) lets you test your regular expressions against any entry of your choice and clearly highlights all matches. It is based on the [Pattern class](#) of Java 8.0.

Consult the [regular expression documentation](#) or the [regular expression solutions to common problems](#) section of this page for examples. If you need more examples or solutions, please [contact me](#).

**Java Regular Expression :****Entry to test against :****Replace with (Optional):**

You can make use of \$1, \$2, \$3 and so on if you are using parenthesis groups in your regular expression. \t \n \r are supported.

**Flags:**

- |                                     |  |
|-------------------------------------|--|
| <input type="checkbox"/> Dotall     | <input type="checkbox"/> Comments                |
| <input type="checkbox"/> Multiline  | <input type="checkbox"/> Literal                 |
| <input type="checkbox"/> Unix lines | <input type="checkbox"/> Unicode case            |
| <input type="checkbox"/> Canon EQ   | <input type="checkbox"/> Unicode character class |

**TEST MATCH****REPLACE FIRST****REPLACE ALL**

## Regular Expression - Documentation

### Metacharacters

## Related Tools

Text Editor Online

## Regex Tester

Regex Replace

Word Counter

Character Count

Case Converter

Reverse Text

Number to Words

## Other Tools By Category



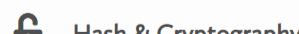
Image Utilities



Formatter &amp; Minifier



Internet Testing



Hash &amp; Cryptography



Number Conversion



Text &amp; String Utilities

## Regex Tester Online Tool

Regular Expression

([A-Z])\w+

Flags(Part 1)

g ON(global)



i OFF(ignore case)



m OFF(multi-line)



Flags(Part 2)

s OFF(dotAll)



u OFF(Unicode)



y OFF(sticky)



- 1 This online **Regex Tester** tool helps you to test if your regular expression is working correctly. It support Matching h2-highlight and 6 different Flags, powered by **Javascript RegExp**.

## About Regex Tester Online Tool:

This online Regex Tester tool helps you to test if your regular expression is working correctly. It support Matching h2-highlight and 6 different Flags, powered by Javascript RegExp.

Flags Type	Description
g	(global match), find all matches
i	ignore case
m	multiline; treat beginning and end characters (^ and \$) as working over multiple lines
s	allows . to match newlines
u	Unicode; treat pattern as a sequence of Unicode
y	sticky; matches only from the index indicated by the lastIndex



# Regex Crossword

Welcome to the fantastic world of nerdy regex fun! Start playing by selecting one of the puzzle challenges below. There are a wide range of difficulties from beginner to expert.

[How to play »](#)

## Mobile (NEW!)

Try our new mobile version! Optimized for phones and solving puzzles on the go.

[Play »](#)

## Tutorial

A step by step tutorial, teaching you the different symbols and regex patterns.

[Play »](#)

## Intermediate

So you've got skills eh? Let's see how you handle a tougher challenge...

[Play »](#)

## Double Cross

Don't get cross-eyed, the clues are coming from all

## Beginner

Cut your teeth on an easy set of crosswords, learning the basics of regular expressions.

[Play »](#)

## Experienced

Now it's getting difficult. We are ramping up the size and complexity. Try to keep up!

[Play »](#)

## Cities

Rome wasn't built in a day you know, but see if you can

## Palindromeda

Bend your mind around these cubistic 2D palindrome puzzles.

[Play »](#)

## Volapük

This is not a crossword, this is an exercise in pure regex



# Borders

^ start of Exp  
\$ end of Exp

^Hello  
bye.\$

Hello....  
... bye.

# Quantifiers

?	0x or 1x	ab?c	“abc”, “ac”
*	0x to $\infty$	ab*c	“ac”, “abc”, “abbbbc”
+	1x to $\infty$	ab+c	“abc”, “abbbc”
{n}	Nx	ab{3}c	“abbcc”
{n,m}	form Nx to Mx	ab{1,3}c	“abc”, “abbc”, “abbcc”
{n, }	from Nx to $\infty$	ab{3, }c	“abbbbc”

# Brackets

[0-9]

any character from 0 to 9

[a-z]

any character from a - z

[A-Z]

any character from A - Z

[abc]

any character in the given set

[^abc]

any character outside the given set

(red|blue|green)

any of the alternatives

# Metacharacters

.

single character

/

normal char from metachar

/w

word character [a-zA-Z0-9\_]

/d

digit character [0-9]

# Validácia čísel

1. Kladné celé čísla nedefinovanej dĺžky:

➤ `^\d+$`

2. Kladné celé čísla maximálnej dĺžky (v našom príklade 10)

➤ `^\d{1,10}$`

3. Pozitívne celé čísla pevnej dĺžky (v našom príklade 5):

• `^\d{5}$`

4. Záporné celé čísla nedefinovanej dĺžky:

• `^- \d+$`

5. Záporné celé čísla maximálnej dĺžky (v našom príklade 10)

• `^- \d{1,10}$`

# Validácia čísel 2

1. Záporné celé čísla pevnej dĺžky (v našom príklade 5)
  - $^-\backslash\text{d}\{5\}\$$
2. Celé čísla nedefinovanej dĺžky:
  - $^-\text{?}\backslash\text{d}+\$$
3. Celé čísla maximálnej dĺžky (v našom príklade 10)
  - $^-\text{?}\backslash\text{d}\{1,10\}\$$
4. Celé čísla pevnej dĺžky (v našom príklade 5)
  - $^-\text{?}\backslash\text{d}\{5\}\$$
5. Počet nedefinovaných dĺžok s desatinnými miestami alebo bez nich (1234.1234)
  - $^-\text{?}\backslash\text{d}*\backslash\text{.}\{\text{0,1}\}\backslash\text{d}+\$$
6. Čísla s 2 desatinnými miestami (0,00)
  - $^-\text{?}\backslash\text{d}*\backslash\text{.}\backslash\text{d}\{2\}\$$

# Čísla mien s voliteľným znakom dolára a oddelovačmi tisícov a voliteľnými 2 miestami

- `^$?\\-?([1-9]{1}[0-9]{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2}))$|^\\-?\\$?([1-9]{1}\\d{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2}))$|^\\($?([1-9]{1}\\d{0,2}(\\,\\d{3})*(.\\d{0,2})?|[1-9]{1}\\d{0,}(\\.\\d{0,2})?|0(\\.\\d{0,2})?|(\\.\\d{1,2})))\\)$`

# Percento od 0 do 100 s voliteľnými 2 miestami a znamienkom % na konci

- ^-?[0-9]{0,2}(\.[0-9]{1,2})?%?\$ | ^-?(100)(\.[0]{1,2})?%?\$

# Hexadecimálny farebný kód (#FFFFFF)

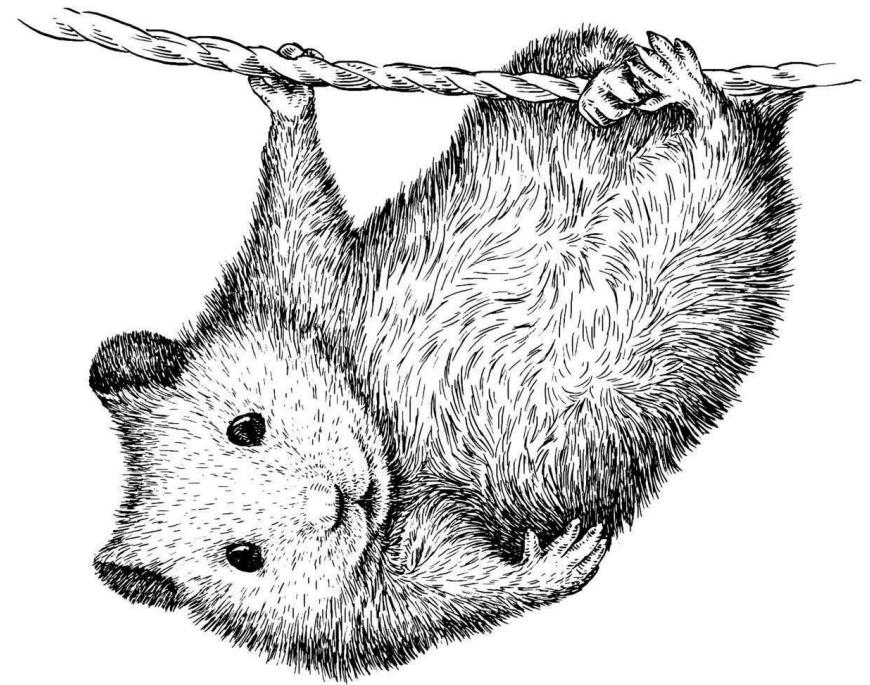
- ^#?([a-f0-9]{6}|[a-f0-9]{3})\$

# Alfanumerické hodnoty

- ^[ a-zA-Z0-9 ]+\$

# Emailová adresa

- ^[-a-zA-Z0-9~!\$%^&\*\_+=]{\'.}[ -a-zA-Z0-9~!\$%^&\*\_+=]{\'.}+@[ -a-zA-Z0-9\_-][-a-zA-Z0-9\_-]\*(\.[-a-zA-Z0-9\_-]+)\*\.(aero|arpa|biz|com|coop|edu|gov|info|int|mil|museum|name|net|org|pro|travel|mobi|[a-zA-Z][a-zA-Z])|([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}))(:[0-9]{1,5})?\$\_



Googling for  
the Regex

*Every. Damn. Time.*

# Regulárny výraz email

```
public String validateEmail(String email) {  
    var re = "^(\\^<>()\\[\\]\\.,;:\\s@\\"]+\\(.\\^<>()\\[\\]\\.,;:\\s@\\"]+\\))*\\(\\\".+\\\")\\)@\\((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\])|(([a-zA-Z\\-0-9]+\\.)+[a-zA-Z]{2,}))$";  
    return re.test(email);  
}
```

# IP adresa (IPV4)

- `^(?:(?:25[0-5]|2[0-4][0-9]| [01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]| [01]?[0-9][0-9]?)$`

# Formát dátumu ISO (rrrr-mm-dd)

- ^[0-9]{4}-(((0[13578] | (10|12))- (0[1-9] | [1-2][0-9] | 3[0-1])) | (02- (0[1-9] | [1-2][0-9])) | ((0[469] | 11)- (0[1-9] | [1-2][0-9] | 30)))\$

# Formát dátumu ISO (rrrr-mm-dd) s oddelovačmi - / . "

- `^[0-9]{4}([- /.])(((0[13578] | (10|12))\1(0[1-9] | [1-2][0-9] | 3[0-1])) | (02\1(0[1-9] | [1-2][0-9])) | ((0[469] | 11)\1(0[1-9] | [1-2][0-9] | 30)))$`

# Formát dátumu v USA (mm / dd / rrrr)

- ^(((0[13578] | (10|12)) / (0[1-9] | [1-2][0-9] | 3[0-1])) | (02 / (0[1-9] | [1-2][0-9]))) | ((0[469] | 11) / (0[1-9] | [1-2][0-9] | 30))) / [0-9]{4}\$

# Hodiny a minúty, 24-hodinový formát (HH:MM)

- ^ (20|21|22|23|[01]\d|\d) ((:[0-5]\d){1,2})\$

---

# Mutovatelnost' (Mutability)

 python



# Immutable v Pythonе

- Znamená, keď **nemôžete meniť typ objektu v priebehu času**
- Ak **nie je možné zmeniť hodnotu objektu** je známy ako **nemenný (immutable) objekt**
- Po **vytvorení nemenného objektu** **zostáva jeho hodnota trvalá a nemenná**

# Mutovateľnosť/Nemutovateľnosť

mutable

1. list
2. set
3. dict
4. byte array

immutable

1. number (int, float, complex)
2. string
3. tuple
4. frozen set
5. bytes

# Princíp identity a funkcia id()

## Úplna identita:

- Objekt.atribut/metoda
- Kto/Čo . Čo chcem dostat/vykonať
- Karol.zaplat()
- Karol.vek

## Anonymná identita:

- Neuvediem meno objektu . Čo chcem dostat/vykonať
- zaplat()

- Vracia **celé číslo** (alebo dlhé celé číslo), pre ktoré je zaručené, že je **jedinečné** a **konštantné** pre tento objekt počas jeho životnosti
- Podobné ako v jazyku C adresa objektu v pamäti
- Používanie v **operátore is a is not**

# Používanie id()

```
1 a = 1  
2 b = a  
3 c = b  
4 d = 1  
5  
6 print id(a)  
7 print id(b)  
8 print id(c)  
9 print id(d)
```



40928888  
40928888  
40928888  
40928888

# Operátor is a in

## Testovanie identity

```
>>> a = 'pub'
>>> b = ''.join(['p', 'u', 'b'])
>>> a == b
True
>>> a is b
False
```

## Testovanie začlenenia

```
meno = 'laco'
'a' in meno
'x' in meno

rodina = ['mama', 'otec', 'brat']
'sestra' in rodina
'mama' in rodina

obchod=['chleba', 'mlieko', 'maso', 'maslo', 'ryza']

for jedlo in obchod:
    print "chcem " + jedlo
```

is je `id(a) == id(b)`



# Manažment: Máme skvelé Datasety

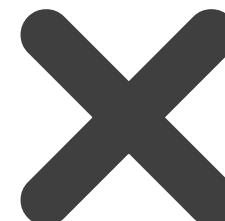
 python

'St Albans',  
'St.Ablans',  
'St.albans',  
'St. Alans',  
'S.Albans',  
'St..Albans',  
'S.Albnas',  
'St.Albnas',  
'St.Al bans',  
'St.Algans',  
'Sl.Albans',  
'St. Allbans',  
'St, Albans'<sup>5</sup>,

# Spájanie Retázcov

```
names = ['raymond', 'rachel', 'matthew', 'roger',
         'betty', 'melissa', 'judith', 'charlie']
```

```
s = names[0]
for name in names[1:]:
    s += ', ' + name
print s
```



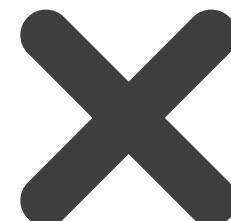
```
print ', '.join(names)
```



# Rozbal'ovanie Sekvencií

```
p = 'Raymond', 'Hettinger', 0x30, 'python@example.com'
```

```
fname = p[0]
lname = p[1]
age = p[2]
email = p[3]
```



```
fname, lname, age, email = p
```



# Aktualizácia Sekvencií

```
print(f"\n2. Spôsob Python")
studenti2 = deque(['Adam', 'Rachel', 'Matej', 'Radoslav',
'Brano', 'Maros', 'Julia', 'Cyril'])
print(f"Zoznam studentov2: {studenti2}")
```



```
del studenti2[0]
print(f"Zoznam studentov2 po del[0] - Adam: {studenti2}")
```

```
studenti2.popleft()
print(f"Zoznam studentov2 po popleft - Rachel: {studenti2}")
```

```
studenti2.appendleft('Marek')
print(f"Zoznam studentov2 po appendleft('Marek') - Marek:
{studenti2}")
```

# Hľadanie Sekvencií

```
print(f"\n2. Spôsob Python")
def hladaj2(sekvencia, ciel):
    for i, value in enumerate(sekvencia):
        if value == ciel:
            break
    else:
        return -1
    return i
```



```
print(hladaj2(studenti, "Adam"))
print(hladaj2(studenti, "Julia"))
print(hladaj2(studenti, "Jula"))
```

# Modul Secrets

---

 python



# Modul Secrets

- **Generovanie bezpečných náhodných čísel** na správu tajných informácií
- Modul tajných informácií sa používa na generovanie kryptograficky silných náhodných čísel vhodných **na správu údajov**, ako sú **heslá, autentifikácia účtov, bezpečnostné tokeny a súvisiace tajomstvá**
- **Secrets by sa mali používať prednostne pred** predvoleným generátorom pseudonáhodných čísel v module **random**, ktorý je určený na modelovanie a simuláciu, nie na bezpečnosť alebo kryptografiu

## Table of Contents

**secrets** — Generate secure random numbers for managing secrets

- Random numbers
- Generating tokens
  - How many bytes should tokens use?
- Other functions
- Recipes and best practices

## Previous topic

**hmac** — Keyed-Hashing for Message Authentication

## Next topic

Generic Operating System Services

## This Page

[Report a Bug](#)  
[Show Source](#)

# secrets — Generate secure random numbers for managing secrets

*New in version 3.6.*

**Source code:** [Lib/secrets.py](#)

The **secrets** module is used for generating cryptographically strong random numbers suitable for managing data such as passwords, account authentication, security tokens, and related secrets.

In particular, **secrets** should be used in preference to the default pseudo-random number generator in the **random** module, which is designed for modelling and simulation, not security or cryptography.

**See also:** [PEP 506](#)

## Random numbers

The **secrets** module provides access to the most secure source of randomness that your operating system provides.

**class secrets.SystemRandom**

A class for generating random numbers using the highest-quality sources provided by the operating system.  
See [random.SystemRandom](#) for additional details.

**secrets.choice(sequence)**

Return a randomly chosen element from a non-empty sequence.

**secrets.randrange(n)**

Return a random int in the range [0, n).

**secrets.randbits(k)**

Return an int with k random bits.

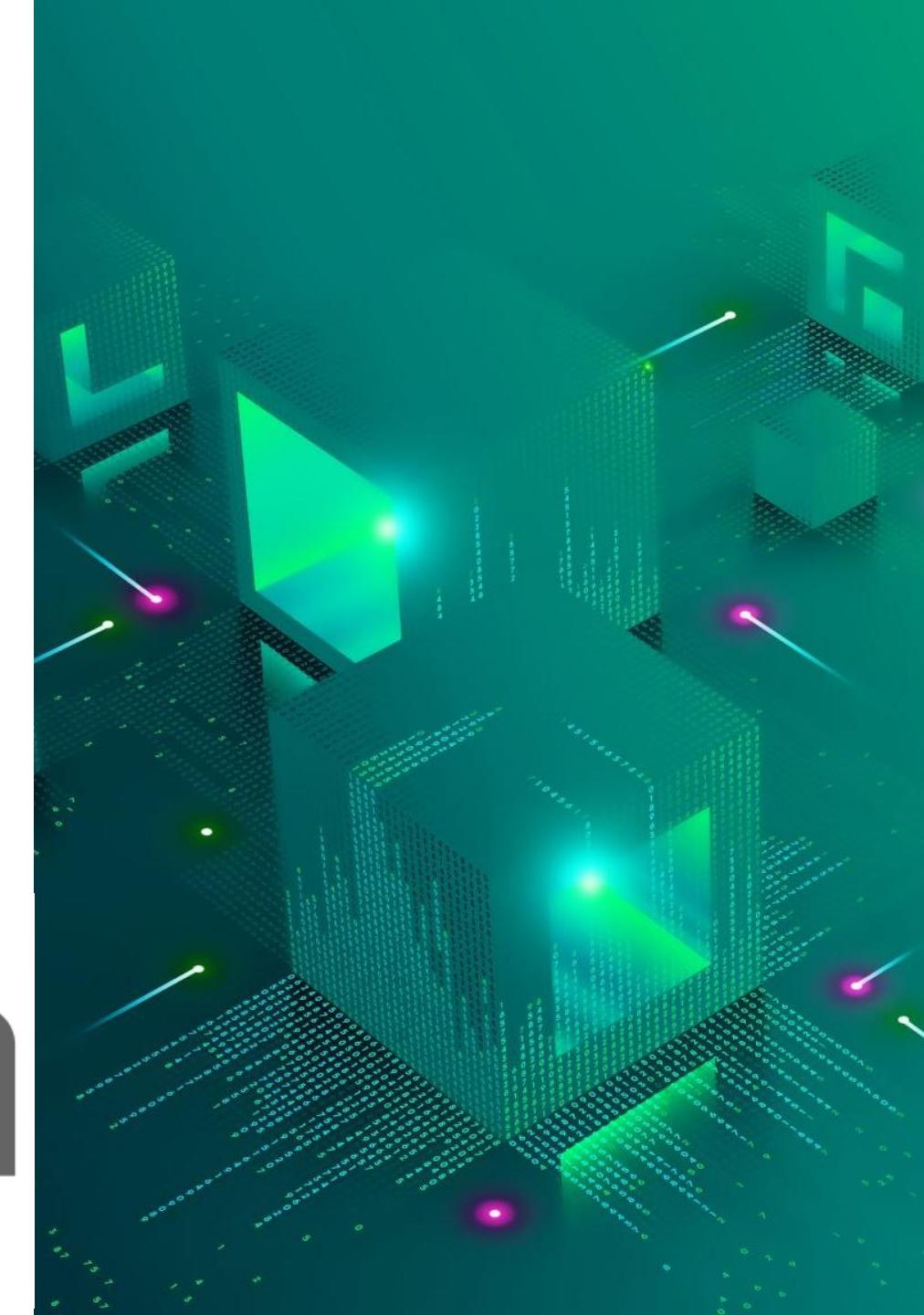
# Secrets a Generovanie hesla

```
1 # https://docs.python.org/3/library/secrets.html
2 import string
3 import secrets
4
5 abeceda = string.ascii_letters + string.digits
6 print(f"abeceda: {abeceda}")
7
8 heslo = ''.join(secrets.choice(abeceda) for i in range(8))
9 print(f"heslo: {heslo}")
```

```
abeceda: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
heslo: khRg1MRM
```

# AI a Detekcia Jazyka (langdetect)

 python



# langdetect

▶ 5.6s

AI + ⌂ ⌂ ⌂ ...

```
1 #!pip install langdetect
2 # https://github.com/Mimino666/langdetect
3 # https://pypi.org/project/langdetect/
4 # noinspection PyUnresolvedReferences
5 from langdetect import detect
6 vstupny_text = input("Zadajte lubovolny text v lubovolnom jazyku: ")
7 print(f"Identifikovany jazyk -> {detect(vstupny_text)}")
```

Zadajte lubovolny text v lubovolnom jazyku: uno y dos  
Identifikovany jazyk -> es

# Abstraktné Dátové Štruktúry

---

 python



# Stack

Top →  
(insertion, deletion)



# Queue

Front →  
(deletion)

Rear ←  
(insertion)



# Deque

Front →  
(insertion, deletion)

Rear ←  
(insertion,  
deletion)



# Deque (Double-Ended Queue)

- Kontajneru podobný zoznamu s rýchlym pridávaním (append) a odstraňovaní (pop) na oboch koncoch
- Sú **zovšeobecnením zásobníkov** a **frontov**
- Vyslovuje sa ako „deck“
- Sú **optimalizované pre rýchle operácie s pevnou dĺžkou**
- Môže narásť do ľubovoľnej dĺžky
- **class collections.deque([iterable[, maxlen]])**

# Abstraktné Dátové Štruktúry



Add to left:

`extendleft()`

`appendleft()`

Left Side

Deque

Right Side

Add to right:

`extend ()`

`append ()`

Remove from left:

`popleft()`

Remove from right:

`pop ()`

## Table of Contents

<b>collections</b> — Container datatypes
▪ <b>ChainMap</b> objects
▪ <b>ChainMap</b> Examples and Recipes
▪ <b>Counter</b> objects
▪ <b>deque</b> objects
▪ <b>deque</b> Recipes
▪ <b>defaultdict</b> objects
▪ <b>defaultdict</b> Examples
▪ <b>namedtuple()</b> Factory Function for Tuples with Named Fields
▪ <b>OrderedDict</b> objects
▪ <b>OrderedDict</b> Examples and Recipes
▪ <b>UserDict</b> objects
▪ <b>UserList</b> objects
▪ <b>UserString</b> objects

# collections — Container datatypes

**Source code:** [Lib/collections/\\_\\_init\\_\\_.py](#)

This module implements specialized container datatypes providing alternatives to Python's general purpose built-in containers, [\*\*dict\*\*](#), [\*\*list\*\*](#), [\*\*set\*\*](#), and [\*\*tuple\*\*](#).

<a href="#"><b>namedtuple()</b></a>	factory function for creating tuple subclasses with named fields
<a href="#"><b>deque</b></a>	list-like container with fast appends and pops on either end
<a href="#"><b>ChainMap</b></a>	dict-like class for creating a single view of multiple mappings
<a href="#"><b>Counter</b></a>	dict subclass for counting <a href="#"><b>hashable</b></a> objects
<a href="#"><b>OrderedDict</b></a>	dict subclass that remembers the order entries were added
<a href="#"><b>defaultdict</b></a>	dict subclass that calls a factory function to supply missing values
<a href="#"><b>UserDict</b></a>	wrapper around dictionary objects for easier dict subclassing
<a href="#"><b>UserList</b></a>	wrapper around list objects for easier list subclassing
<a href="#"><b>UserString</b></a>	wrapper around string objects for easier string subclassing

Previous topic

[calendar](#) — General calendar-related functions

## ChainMap objects

# deque objects

## Table of Contents

- `collections`** — Container datatypes
  - **`ChainMap`** objects
    - **`ChainMap`**  
Examples and Recipes
  - **`Counter`** objects
  - **`deque`** objects
    - **`deque`** Recipes
  - **`defaultdict`** objects
    - **`defaultdict`**  
Examples
  - **`namedtuple()`** Factory Function for Tuples with Named Fields
  - **`OrderedDict`** objects
    - **`OrderedDict`**  
Examples and Recipes
  - **`UserDict`** objects
  - **`UserList`** objects
  - **`UserString`** objects

## Previous topic

- `calendar`** — General calendar-related functions

## Next topic

- `collections.abc`** — Abstract Base Classes for Containers

### `class collections.deque([iterable[, maxlen]])`

Returns a new deque object initialized left-to-right (using `append()`) with data from *iterable*. If *iterable* is not specified, the new deque is empty.

Deques are a generalization of stacks and queues (the name is pronounced “deck” and is short for “double-ended queue”). Deques support thread-safe, memory efficient appends and pops from either side of the deque with approximately the same O(1) performance in either direction.

Though `list` objects support similar operations, they are optimized for fast fixed-length operations and incur O(*n*) memory movement costs for `pop(0)` and `insert(0, v)` operations which change both the size and position of the underlying data representation.

If *maxlen* is not specified or is `None`, deques may grow to an arbitrary length. Otherwise, the deque is bounded to the specified maximum length. Once a bounded length deque is full, when new items are added, a corresponding number of items are discarded from the opposite end. Bounded length deques provide functionality similar to the `tail` filter in Unix. They are also useful for tracking transactions and other pools of data where only the most recent activity is of interest.

Deque objects support the following methods:

#### `append(x)`

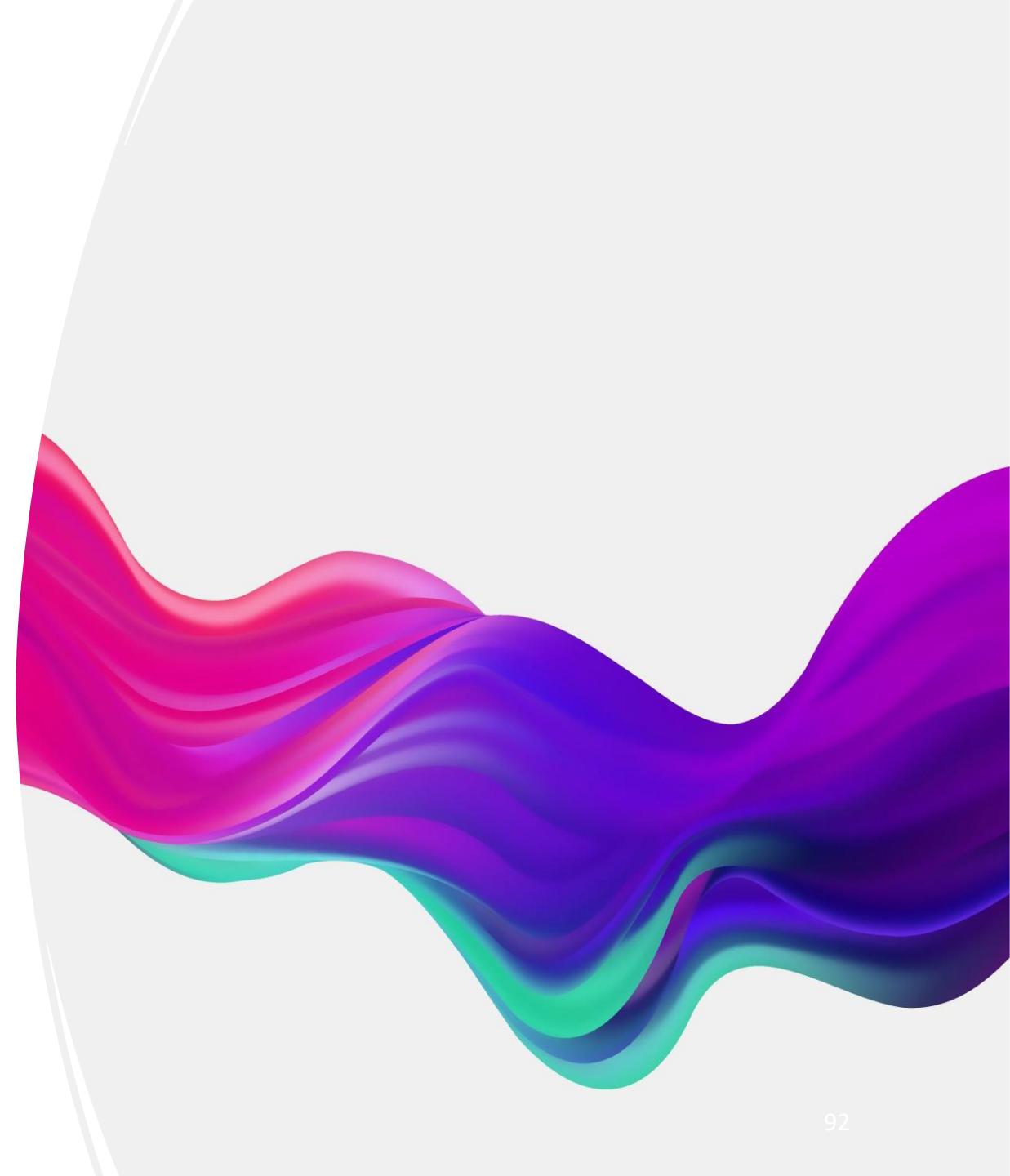
Add *x* to the right side of the deque.

#### `appendleft(x)`

Add *x* to the left side of the deque.

#### `clear()`

# Type Hinting



## Table of Contents

### typing — Support for type hints

- Relevant PEPs
- Type aliases
- NewType
- Annotating callable objects
- Generics
- Annotating tuples
- The type of class objects
- User-defined generic types
- The `Any` type
- Nominal vs structural subtyping
- Module contents
  - Special typing primitives
    - Special types
    - Special forms
    - Building generic types and type aliases
    - Other special directives

# typing — Support for type hints

*New in version 3.5.*

**Source code:** [Lib/typing.py](#)

**Note:** The Python runtime does not enforce function and variable type annotations. They can be used by third party tools such as type checkers, IDEs, linters, etc.

This module provides runtime support for type hints. For the original specification of the typing system, see [PEP 484](#). For a simplified introduction to type hints, see [PEP 483](#).

The function below takes and returns a string and is annotated as follows:

```
def greeting(name: str) -> str:  
    return 'Hello ' + name
```

In the function `greeting`, the argument `name` is expected to be of type `str` and the return type `str`. Subtypes are accepted as arguments.

New features are frequently added to the `typing` module. The `typing_extensions` package provides backports of these new features to older versions of Python.

For a summary of deprecated features and a deprecation timeline, please see [Deprecation Timeline of Major](#)

▶ 0.1s

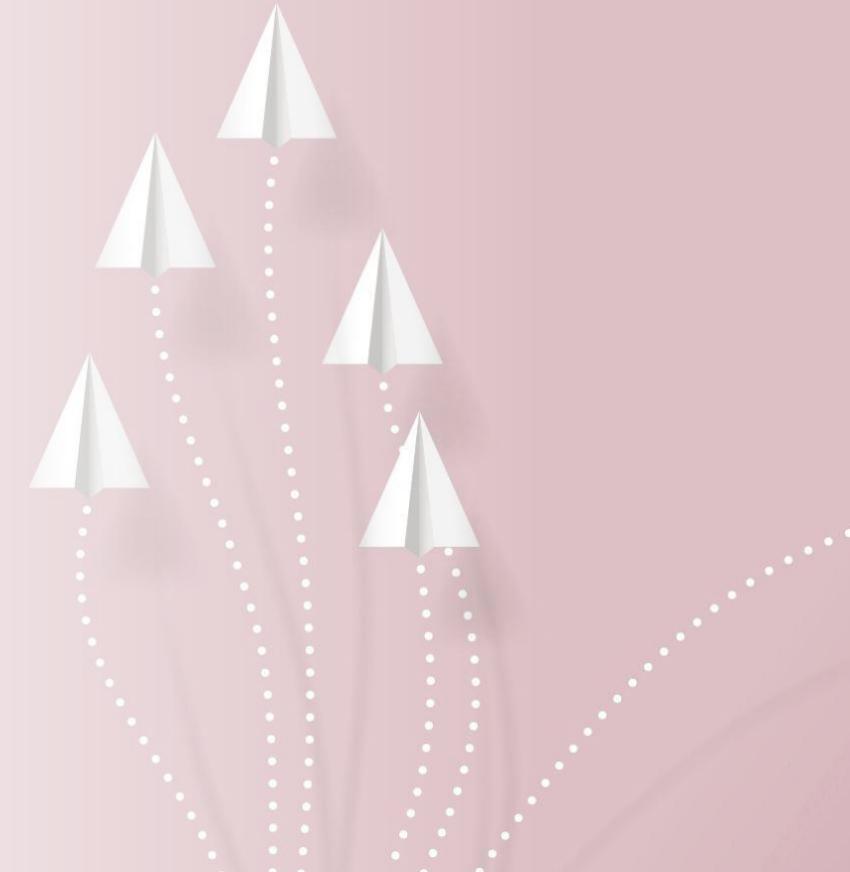
```
1 # noinspection PyShadowingNames
2 def vypis_ahoj(meno: str) -> str:
3     return "Ahoj, " + meno
4
5 vypis_ahoj("Karol")
```

'Ahoj, Karol'

```
1 # noinspection PyShadowingNames
2 def naformatuj_text(text: str, zarovnanie: bool = True) -> str:
3     if zarovnanie:
4         return f"{text.title()}\n{'-' * len(text)}"
5     else:
6         return f" {text.title()} ".center(40, "*")
7
8 print(naformatuj_text("Python je krasny jazyk"))
9 print(naformatuj_text("Python je krasny jazyk", zarovnanie = False))
10
```

```
Python Je Krasny Jazyk
-----
***** Python Je Krasny Jazyk *****
```

# Inštančné, Triedne a Statické Metódy



# Prehľad OOP Metód

```
▶ 1 class Trieda:  
2     def metoda_instancna(self):  
3         return "Inštančná (Objektová) Metóda", self  
4  
5     @classmethod  
6     def metoda_triedy(cls):  
7         return "Metóda Triedy (Triedová Metóda)", cls  
8  
9     @staticmethod  
10    def metoda_staticka():  
11        return "Statická Metóda"
```



# Inštančné Metódy

- Prvá metóda na MyClass, nazývaná metóda, je metóda bežnej inštancie. Toto je základný typ metódy bez dekorátorov, ktorý budete používať väčšinu času.
- Môžete vidieť, že metóda má 1 parameter, self, ktorý pri volaní metódy ukazuje na inštanciu MyClass (ale samozrejme metody inštancie môžu akceptovať viac ako len jeden parameter).
- Prostredníctvom parametra self môžu metódy inštancie voľne pristupovať k atribútom a iným metódam toho istého objektu. To im dáva veľa sily, pokiaľ ide o úpravu stavu objektu.
- Nielenže môžu upravovať stav objektu, ale metódy inštancie môžu tiež pristupovať k samotnej triede prostredníctvom atribútu self. `__class__`. To znamená, že metódy inštancie môžu tiež upravovať stav triedy.

# Metódy Triedy

- Porovnajme to s 2 metódou, `MyClass.classmethod`. Túto metódu som označil dekorátorom `@classmethod`, aby som ju označil ako metódu triedy.
- Namiesto akceptovania parametra `self`, metódy triedy berú pri volaní metódy parameter `cls`, ktorý ukazuje na triedu – a nie na inštanciu objektu.
- Pretože metóda triedy má prístup iba k tomuto argumentu `cls`, nemôže zmeniť stav inštancie objektu. To by si vyžadovalo prístup k sebe. Metódy triedy však môžu stále upravovať stav triedy, ktorý platí pre všetky inštancie triedy.

# Statické Metód

- Tretia metóda, MyClass.staticmethod, bola označená dekorátorom @staticmethod, aby sa označila ako statická metóda.
- Tento typ metódy neberie ani self, ani parameter cls (ale samozrejme je možné akceptovať ľubovoľný počet ďalších parametrov).
- Preto statická metóda nemôže zmeniť stav objektu ani stav triedy. Statické metódy sú obmedzené v tom, ku ktorým údajom majú prístup – a sú to predovšetkým spôsob, ako vytvoriť priestor názovov vašich metód.

# Dekorátory (Decorators)

 python



# Dekorátory

- Výkonný a užitočný nástroj
- Umožňuje **zabalíť ďalšiu funkciu**, aby sme **rozšírili správanie zabalenej funkcie bez** toho, aby sme ju **natrvalo upravovali**
- V dekorátoroch sa **funkcie berú ako argument inej funkcie** a potom sa volajú vo wrapperu funkcie

## Výhody

- Logovanie a ladenie
- Kontrola prístupu a autentifikácia



# Lambdy λ



# python

$y = g(x)$

Secant Lines

Tangent Line  $T$

$x+h$

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$
$$f'(x) = \lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h}$$
$$= \lim_{h \rightarrow 0} \frac{x^2 + 2xh + h^2 - x^2}{h}$$
$$= \lim_{h \rightarrow 0} \frac{2xh + h^2}{h}$$
$$= \lim_{h \rightarrow 0} h(2x + h)$$

103

# Lamby $\lambda$

- Sú **malé, anonymné funkcie**, ktoré podliehajú **prísnejšej**, ale **stručnejšej syntaxi** ako bežné funkcie Pythonu.
- Alonzo Church formalizoval lambda kalkul, jazyk založený na čistej abstrakcii, v 30. rokoch 20. storočia.

1 # Výraz sa skladá z:

2 # Kľúčové slovo (keyword): lambda

3 # Viazaná premenná (bound variable) – argument: x

4 # Telo (body): x

5

6 **lambda** x: x

7 **lambda** x: x + 100

8 # Uvedenú funkciu môžete použiť na argument tak, že funkciu a jej argument uzavtoríte do zátvoriek:

9 (**lambda** x: x + 100)(10)

```
▶ 0.1s AI + ⌂ ...  
1 print(list(map(lambda x: x.upper(), ['cat', 'dog', 'cow'])))  
2 print(list(map(lambda x: x.capitalize(), ['cat', 'dog', 'cow'])))  
3  
4 print(list(filter(lambda x: 'o' in x, ['cat', 'dog', 'cow'])))  
5  
6 even = lambda x: x%2 == 0  
7 print(list(filter(even, range(11))))  
8  
9 from functools import reduce  
10 print(reduce(lambda acc, x: f'{acc} | {x}', ['cat', 'dog', 'cow']))
```

```
['CAT', 'DOG', 'COW']  
['Cat', 'Dog', 'Cow']  
['dog', 'cow']  
[0, 2, 4, 6, 8, 10]  
cat | dog | cow
```

```
▶ 0.0s AI + ⌂ ...  
1 ids = ['id1', 'id2', 'id30', 'id3', 'id22', 'id100']  
2 # Lexikografické triedenie  
3 print(sorted(ids))  
4  
5 # Číselné triedenie  
6 sorted_ids = sorted(ids, key=lambda x: int(x[2:]))  
7 print(sorted_ids)  
['id1', 'id100', 'id2', 'id22', 'id3', 'id30']  
['id1', 'id2', 'id3', 'id22', 'id30', 'id100']
```

# Problémy Lambdy $\lambda$

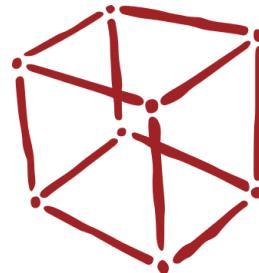
Problémy s  
čitateľnosťou

Zavedenie  
funkčného  
spôsobu  
myslenia

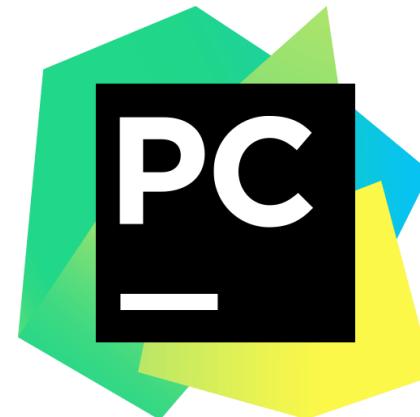
Ťažšia syntax s  
klúčovým slovom  
lambda

# Aké IDE mám použiť?

109



NetBeans



*wxPython*



Visual Studio



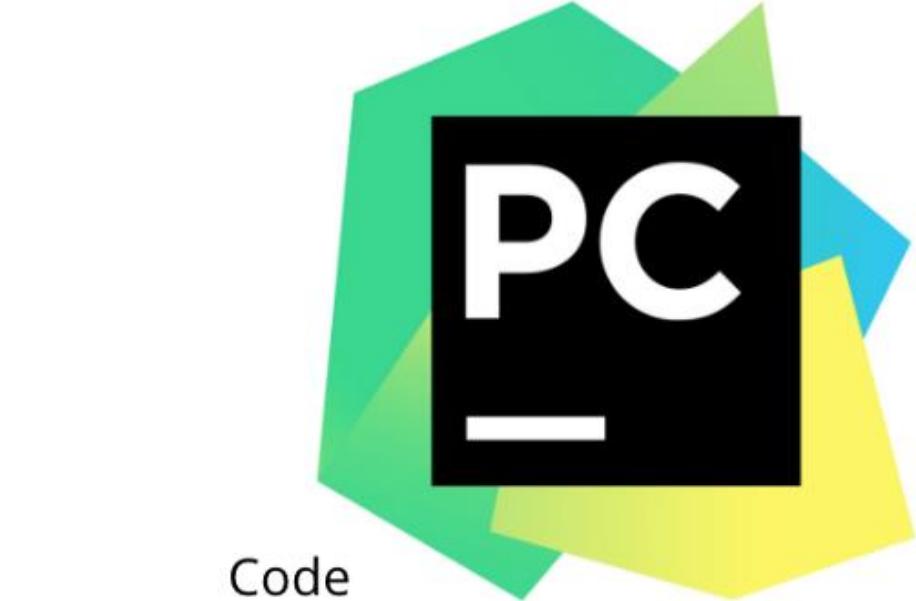
# Chceme úplne všetko!

---





Flexibility



Code  
Completion





```
print("Sorry, we are down for maintenance")
print("We'll be back shortly")
```

Individual Edition is now

# ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

## Anaconda Distribution

[Download !\[\]\(224db00632883ee98f973941824cc050\_img.jpg\)](#)

For Windows  
Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers





### Open Source

Access the open-source software you need for projects in any field, from data visualization to robotics.



### User-friendly

With our intuitive platform, you can easily search and install packages and create, load, and switch between environments.



### Trusted

Our securely hosted packages and artifacts are methodically tested and regularly updated.

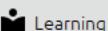
 ANACONDA NAVIGATOR[Sign in to Anaconda Cloud](#)[Home](#)

Applications on

base (root)

Channels

Refresh



Documentation

Developer Blog





**CMD.exe Prompt**  
0.1.1  
Run a cmd.exe terminal with your current environment from Navigator activated

[Launch](#)



**JupyterLab**  
1.2.6  
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Launch](#)



**jupyter**  
Notebook  
6.0.3  
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Launch](#)



**Powershell Prompt**  
0.0.1  
Run a Powershell terminal with your current environment from Navigator activated

[Launch](#)



**Qt Console**  
4.6.0  
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Launch](#)



**Spyder**  
4.0.1  
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)



**VS Code**  
1.52.1  
Streamlined code editor with support for development operations like debugging, task running and version control.

[Launch](#)



**Glueviz**  
0.15.2  
Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Install](#)



**Orange 3**  
3.23.1  
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)



**RStudio**  
1.1.456  
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

[Install](#)

# ANACONDA NAVIGATOR

[i Upgrade Now](#)[Connect ▾](#)[Home](#)[Environments](#)[Learning](#)[Community](#)

Anaconda Notebooks NEW!  
Cloud notebooks with hundreds of packages ready to code

A full Python IDE directly from the browser

[Documentation](#)[Anaconda Blog](#)

Applications on

base (root)

Run a cmd.exe terminal with your current environment from Navigator activated

[Launch](#)

JupyterLab

3.3.2

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Launch](#)

Application launch error



Application **notebook** launch may have produced errors.

```
[I 08:13:13.848 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.4.1  
[W 2022-10-12 08:13:15.513 LabApp] 'password' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.  
[W 2022-10-12 08:13:15.513 LabApp] 'password' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to
```

[Copy text](#)[Ok](#)

Watson Studio Cloud provides you the to analyze and visualize data, to cleanse shape data, to create and train machine learning models. Prepare data and build elts, using open source data science tools or visual modeling.

[Launch](#)

PowerShell Prompt

0.0.1

a PowerShell terminal with your current environment from Navigator activated

[Launch](#)

# Learn Python, JavaScript, C, C++, and Java

This tool helps you learn Python, JavaScript, C, C++, and Java programming by visualizing code execution. You can use it to debug your homework assignments and as a supplement to online coding tutorials.

[Start writing and visualizing code now](#)

Related services: [JavaScript Tutor](#), [C Tutor](#), [C++ Tutor](#), [Java Tutor](#)

**Over ten million people in more than 180 countries** have used Python Tutor to visualize over 100 million pieces of code. It's the most widely-used program visualization tool for computing education.

You can also embed these visualizations into any webpage. Here's an example showing recursion in Python:

Python 3.6

```
1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)
```

[Edit this code](#)

▶ line that just executed  
▶ next line to execute

< Prev    Next >

Step 11 of 22

Visualized using [Python Tutor](#)  
[Customize visualization](#)

Frames      Objects

Global frame

listSum → function listSum(numbers)

myList → tuple [0 1]

listSum → tuple [0 1]

numbers → tuple [0 2]

f → 1

rest → tuple [0 1 None]

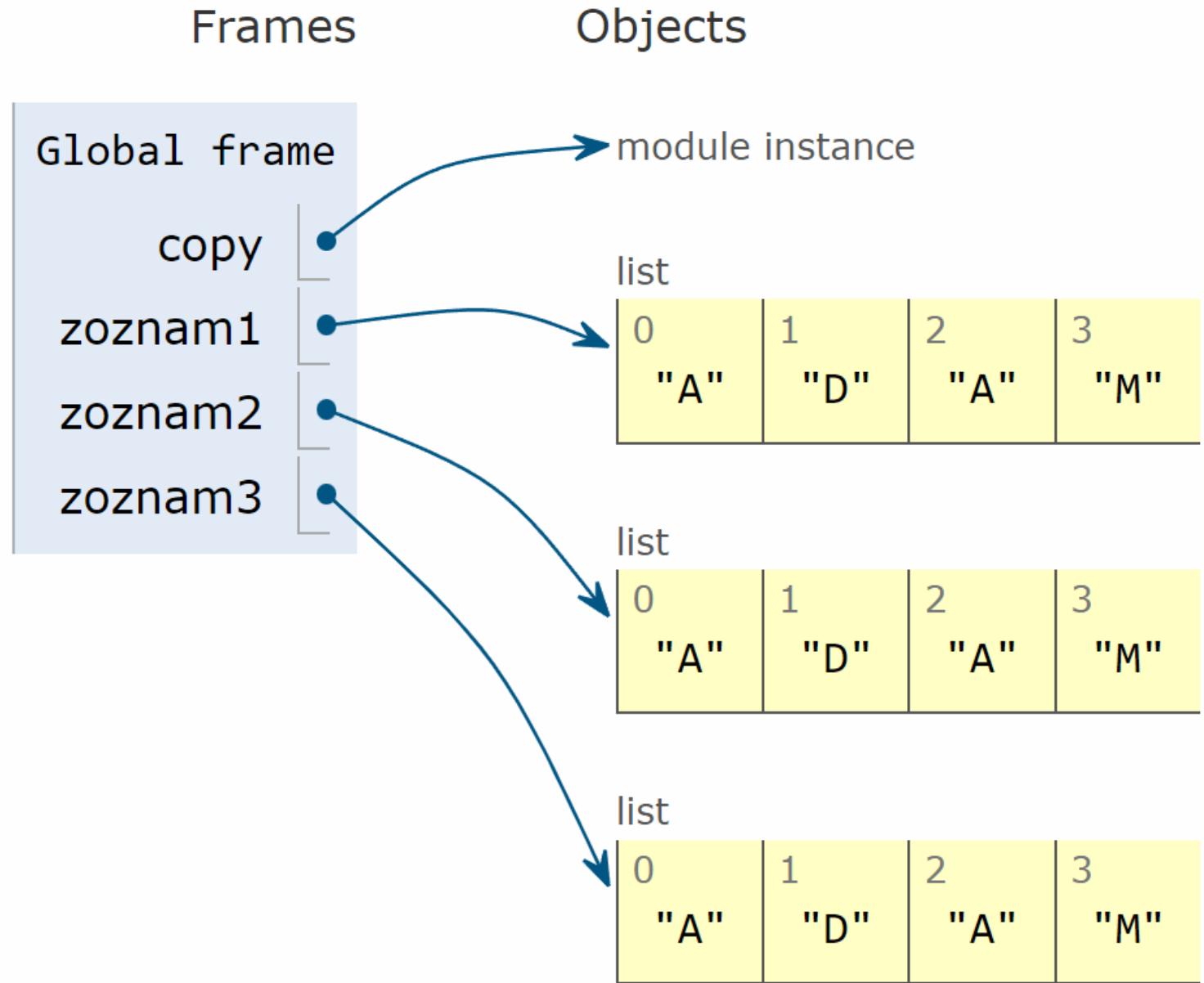
listSum → tuple [0 3]

numbers → tuple [0 3]

f → 2

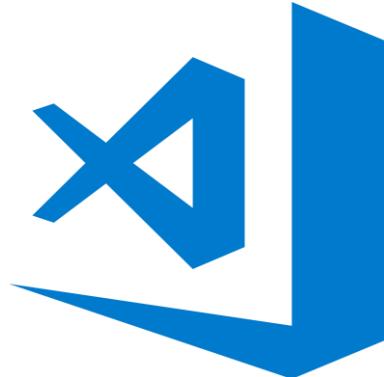
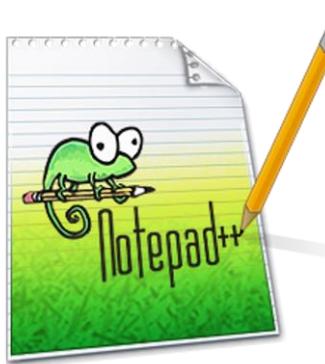
rest → tuple [0 3 None]

# Modul copy

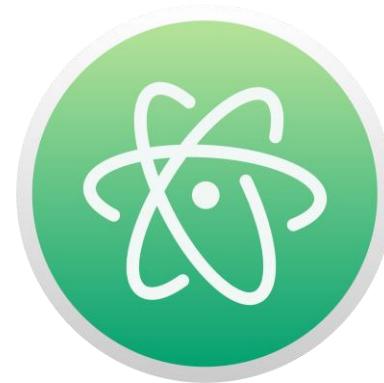




# Aký Editor mám použiť?



```
:::  
iLE880j. :jD888880j:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G8888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```



IDE ≠ editor



# Dashboard

Welcome, [itacademysk](#)**CPU Usage:** 0% used – 0.00s of 100s. Resets in 1 hour, 56 minutes [More Info](#)**File storage:** 0% full – 100.0 KB of your 512.0 MB quota [More Info](#)[Upgrade Account](#)

## Recent Consoles

[+ 5 -](#)*You have no recent consoles.*

## New console:

[\\$ Bash](#)[>>> Python ▾](#)[More...](#)

### Version

[2.7](#)[3.7](#)[3.8](#)

## Recent Files

[+ 5 -](#)

/home/itacademysk/.bashrc  
/home/itacademysk/.gitconfig  
/home/itacademysk/.profile  
/home/itacademysk/.pythonstartup.py  
/home/itacademysk/.vimrc

[+ Open another file](#)[Browse files](#)

## Recent Notebooks

[+ 5 -](#)

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

## All Web apps

*You don't have any web apps.*[Open Web tab](#)

[MySQL](#)[Postgres](#)

## Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

**New password:**

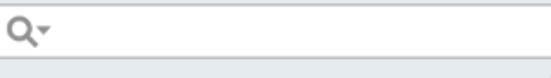
A password input field with a small circular icon containing a question mark in the top right corner.

**Confirm password:**

A password input field with a small circular icon containing a question mark in the top right corner.

**Initialize MySQL**

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.



## Editor > Live Templates

By default expand with **Tab**

### Python

- compd (Dict comprehension)
- compdi (Dict comprehension with 'if')
- compg (Generator comprehension)
- compgi (Generator comprehension with 'if')
- compl (List comprehension)
- compli (List comprehension with 'if')
- comps (Set comprehension)
- compsi (Set comprehension with 'if')
- iter (Iterate (for ... in ...))
- itere (Iterate (for ... in enumerate))
- main (if \_\_name\_\_ == '\_\_main\_\_')
- prop (Property getter)
- props (Property getter/setter)
- propsd (Property getter/setter/deleter)
- super ('super(...)' call)

### R

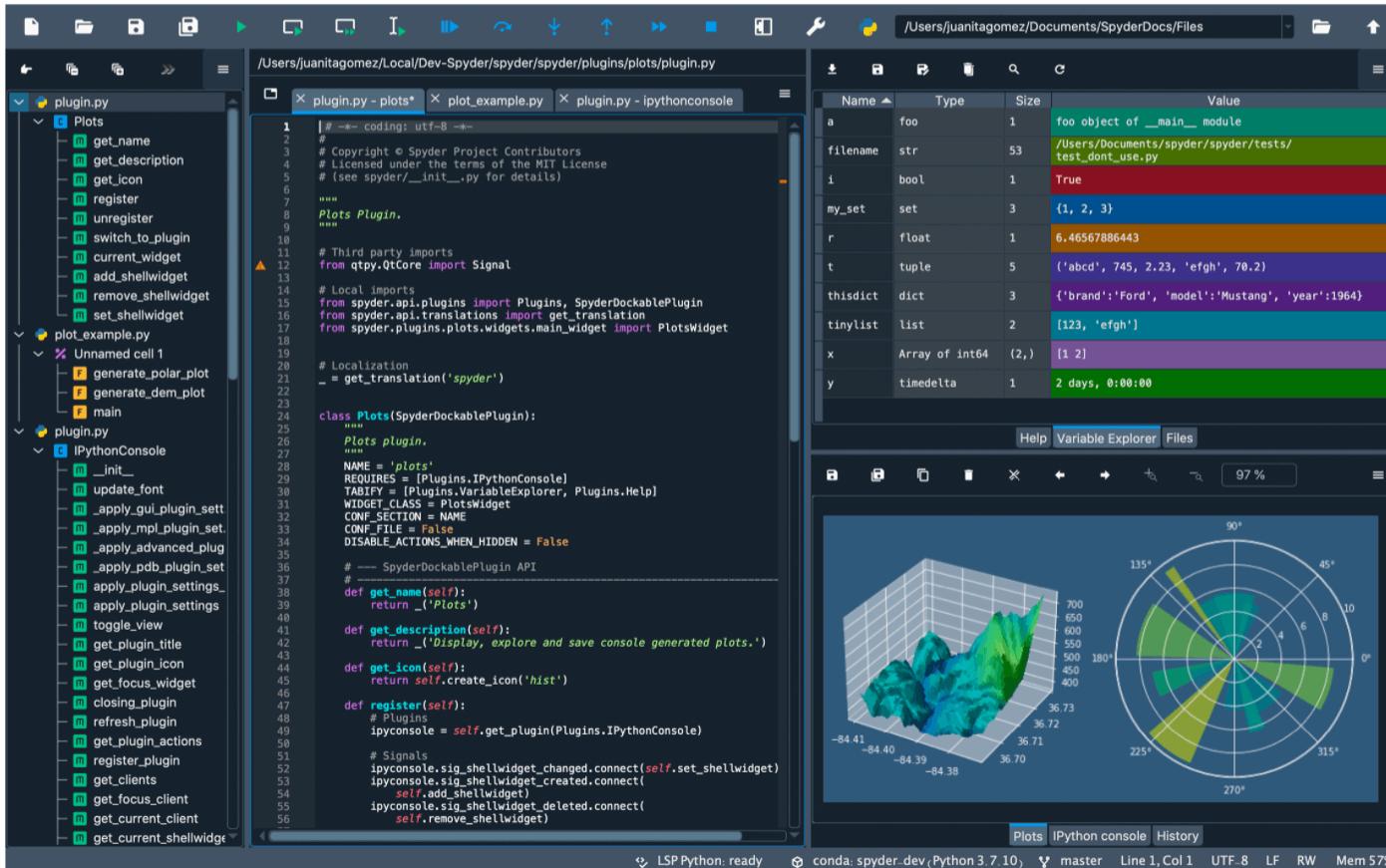
### React

No live templates are selected



# spyder

The  
Scientific  
Python  
Development  
Environment



The screenshot displays the Spyder IDE interface with the following components:

- Code Editor:** Shows the file `/Users/juanitagomez/Local/Dev-Spyder/spyder/spyder/plugins/plots/plugin.py`. The code defines a `Plots` plugin for Spyder, which includes methods for generating polar plots and displaying them in the IPython console.
- Variable Explorer:** A table showing the current state of variables:

Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 78.2)
thisdict	dict	3	{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
tinylist	list	2	[123, 'efgh']
x	Array of int64	(2,)	[1 2]
y	timedelta	1	2 days, 0:00:00

- Plots:** Two plots are displayed: a 3D surface plot of a mountain-like function and a polar plot showing concentric rings with radial labels.
- Bottom Status Bar:** Shows the Python version (LSP Python: ready), conda environment (conda: spyder.dev Python 3.7.10), and system status (master, Line 1, Col 1, UTF-8, LF, RW, Mem 57%).

VERSION

Spyder 5

Search ...



WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

▶ WORKSHOPS

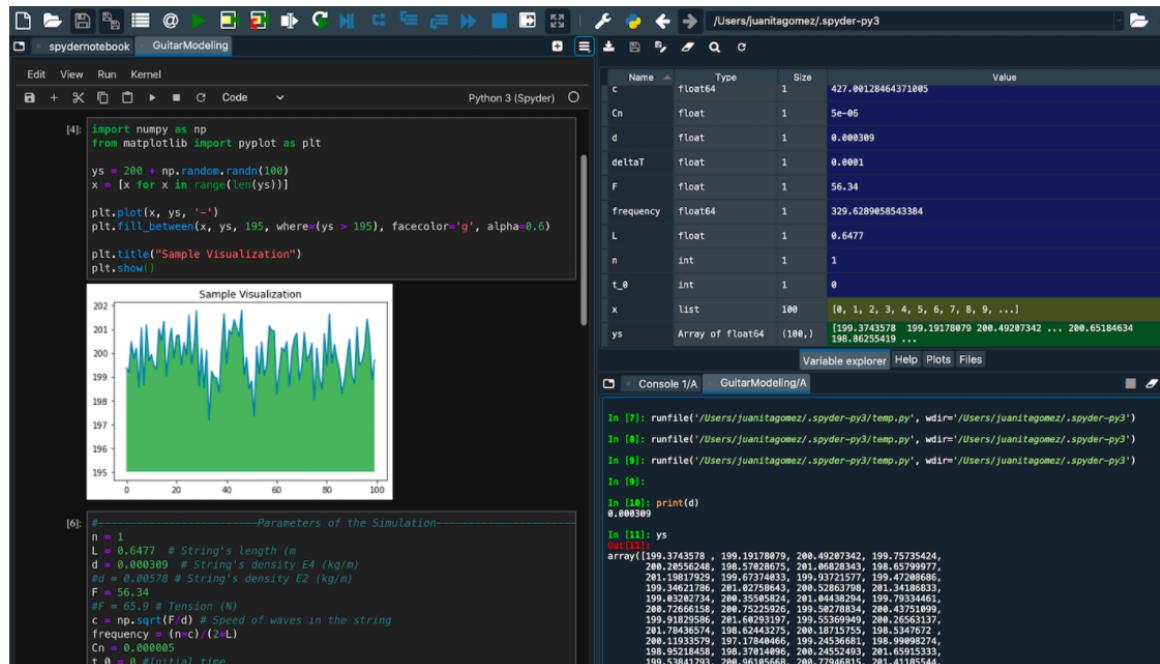
FAQ

# Spyder Notebook

## Warning

Currently, this plugin is still being ported to Spyder 5, and will likely not yet work or experience serious issues on this version of Spyder. A compatible version is expected soon. Thanks for your patience!

**Spyder-notebook** is a plugin that allows you to open, edit and interact with Jupyter Notebooks right inside Spyder.



conda install spyder-notebook -c spyder-ide

[Edit this page](#)[On this page](#)[Installing the Notebook](#)[Using the Notebook](#)[Connecting an IPython Console](#)[Additional Options](#)[OPEN CHAT](#)

## Útržky kódu

Filtrovať útržky kódu

Adding form fields →

Camera Capture →

Cross-output communication →

display.Javascript to execute Jav... →

Downloading files or importing da... →

Adding form fields

Vložit'

Forms example

Forms support multiple types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

```
#@title Example form fields
#@markdown Forms support many types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

no_type_checking = '' #@param
string_type = 'example' #@param
slider_value = 142 #@param {type: "number", min: 0, max: 200}
number = 102 #@param {type: "number", min: 0, max: 200}
date = '2010-11-05' #@param {type: "date", min: "2010-01-01", max: "2020-12-31"}
pick_me = "monday" #@param [ 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday' ]
select_or_input = "apples" #@param {type: "select", options: [ "apples", "oranges", "bananas" ]}
```

[Zobrazit zdrojový zápisník](#)

+ Kód + Text Kopírovať na Disk

```
tiene = True
# r g b, c m y k, w
farby_vlastne = ["black","pink", "b", "#CCCC00"]

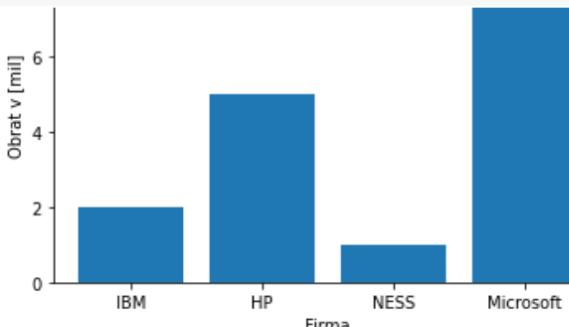
print(type(y))
print(y)

plt.pie(y, labels = menovky, startangle = 0, explode = vykrojenie, shadow = tiene, colors = farby_vlastne)
plt.legend(title = "Produkty ABC s.r.o.")
plt.title("Analyza predaja produktov za Q1-2022")
plt.show()

x1 = np.array(["IBM", "HP", "NESS", "Microsoft"])
y1 = np.array([2, 5, 1, 9])

plt.bar(x1, y1)
plt.title("Porovnanie IT firiem 2022")
plt.xlabel("Firma")
plt.ylabel("Obrat v [mil]")
plt.show()

nahoda = np.random.normal(100, 10, 200)
print(nahoda)
plt.hist(nahoda)
plt.show()
```



```
[ 79.76069196  99.4155264   114.29926387  101.33767141  88.49106384
 111.70288892  91.32702578  102.53587004  108.38846479  114.34889501
 98.79114202  117.40488367  89.26174251  94.12100639  101.96805716]
```

C:\Users\miros

▼ ↑

Python\_I\_Za...

Edit View Run Kernel

Python 3 (Spyder)

Name Type Size Value

## Kurz Python - 1. deň

Miroslav Reiter | miroslav.reiter@it-academy.sk | <https://www.linkedin.com/in/miroslav-reiter/>

Kurz Python | <https://www.it-academy.sk/kurz/python/> | <https://github.com/miroslav-reiter>

## Komentáre, kódovanie, tlač a docstringy

```
[1]: # -*- coding: utf-8 -*-
# Toto je komentár (jednoriadkový)
"""Toto je docstring (document string)"""

# Pozor na nespravne zalomovanie riadku (Enter)
# SyntaxError: EOL while scanning string Literal
# Emotikony https://emojipedia.org/
print("Python je fajnovy jazyk!")
print(__doc__)
print("🎲 🎲 🎲")
print("Co bolo skorej? --> ", min(['\N{CHICKEN}', '\N{EGG}']))
```

Python je fajnovy jazyk!
Toto je docstring (document string)
🎲 🎲 🎲
Co bolo skorej? --> 🎲

## Premenné a typy

•••

Milujem Python
Milujem Python
Milujem Python

Nazov produktu je: Hypoteka pre mladych 2021
Splatka vasej hypoteky je: 600 Eur
Uroková sadzba je: 1.5 % p.a.

Do k diagnostici: False

Editor Notebook

LSP Python: ready Kite: ready (no index) conda: base (Python 3.8.5) main [86] Line 8, Col 1 UTF-8 CRLF RW Mem 47% CPU 17% 09:26

Name Type Size Value

Variable explorer Help Plots Files

Console 1/A

Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]:



C:\Users\miros\AppData\Local\Temp\kite\_tutorial.py

temp.py kite\_tutorial.py

```

1 # Welcome to...
2 #
3 #      `hmy+.      :::
4 #      .mMMMMMNho: ` NMMm
5 #      :NMMMMMMMMMdS/.` NMMm      :ss:
6 #      +NMMMMMMMMMMMMmy+ NMMm      -MMM-   ---
7 #      `oMMMMMMMMMMMMMMMo NMMm      /ss/   :MM+
8 #      `yMMMMMMMMNshmNMMMN` NMMm      /MM+
9 #      .dMMMMMMMMm/hmhssydmMM+ NMMm      `/yhy. shhy ohmMMMhhhh. ..ydmmdho-
10 #     omMMMMMd/mMMMMMhsosy` NMMm      .omMMmo. mMMN odmMMMdsss. omMNsoshNMNy
11 #     .+dMMMy/mMMMMMMMMMd- NMMm-yNMMy/` mMMN /MM+ sMMN: `:NMMy
12 #     `ymo/NMMMMMMMMMd NMMmNMNMMN` mMMN :MM+ MMNdddNNMMN
13 #     `hMMMMMMMMMM: NMMm+mMMNs. mMMN :MM+ MMN//////////////:
14 #     `:yNMMMMMMMMh NMMm `/dMMNy- mMMN :MM+ ` sMMNo` `:
15 #     .+mMMMMMd- NMMm `/dMMNy- mMMN .MMNddNN/ +NMNdhydNNMs
16 #     `:yMMMy yhhs   `/hhh shhs :ydmmdho: `/sdmmmmhs:`
17 #     `om.
18
19 """
20 Kite is your Python programming copilot. Kite will try to show you the
21 right information at the right time as you code to prevent you from context
22 switching out of your current line of thought.
23
24 This tutorial will teach you how to use all of Kite's core features. You
25 should be able to learn everything in 5 minutes.
26
27 If you get stuck at any point, please visit https://help.kite.com/ or file
28 an issue at https://github.com/kiteco/issue-tracker.
29 """
30
31
32
33
34 """ PART 0: BEFORE WE START =====
35
36 Spyder will by default try to start the Kite backend when the editor first
37 starts. You can change this behavior by opening settings, clicking on
38 "Completion and linting", "Advanced", and then changing Kite's "Start Kite
39 Engine on editor startup" setting.
40
41 Look for the Kite indicator in the bottom left corner of Spyder's status
42 bar – It will tell you if Kite is ready and working. If the indicator reads
43 "not running", then you'll have to start the Kite Engine manually before
44 proceeding with the rest of this tutorial.
45 """
46
47
48
49

```

↓ ☰ 🔍 C

Name	Type	Size	Value
a	int	1	5
b	str	1	Karol

Variable explorer Help Plots Files

Console 1/A

Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')

In [2]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')  
5  
Karol

In [3]:



For Teams      Download

# Code Faster. Stay in Flow.

Kite adds AI powered code completions to your code editor, giving developers superpowers.



Download for Free

```
1 import os
2 import sys
3
4 def count_py_files_in_repos(dirname):
5     if os.path.exists(os.path.join(dirname, '.git')):
6         count = 0
7         for root, dirs, files in os.walk(dirname):
8             count += len([f for f in files if f.endswith('.py')])
9         print('{} has {} Python files'.format(dirname, count))
10        for name in os.listdir(di)
```



dirname  
dirs  
dict

kite.com

```
1 import os
2 import sys
3
4 def count_py_files_in_repos(dirname):
5     i|
```

```
C:\Windows\System32\cmd.exe - pip install matlib
Microsoft Windows [Version 10.0.16299.785]
(c) 2017 Microsoft Corporation. Všetky práva vyhradené.

C:\Windows\System32>cd C:\Program Files (x86)\Python37-32\Scripts

C:\Program Files (x86)\Python37-32\Scripts>pip instal xlwt
ERROR: unknown command "instal" - maybe you meant "install"

C:\Program Files (x86)\Python37-32\Scripts>pip install xlwt
Collecting xlwt
  Downloading https://files.pythonhosted.org/packages/44/48/def306413b25c3d01753603b1a222a011b8621aed27cd7f89cbc27e6b0f4/xlwt-1.3.0-py2.py3-none-any.whl (99kB)
    100% |██████████| 102kB 826kB/s
Installing collected packages: xlwt
Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: 'c:\\\\program files (x86)\\\\python37-32\\\\Lib\\\\site-packages\\\\xlwt'
Consider using the `--user` option or check the permissions.

You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Program Files (x86)\Python37-32\Scripts>pip install matlib
Collecting matlib
```

PIP a easy install



**Seems like I've  
installed wrong version  
of Python...**

sys Variables		String Methods		Datetime Methods					
argv	Command line args	capitalize() *	lstrip()	today()	fromordinal(ordinal)				
builtin_module_names	Linked C modules	center(width)	partition(sep)	now(timezoneinfo)	combine(date, time)				
byteorder	Native byte order	count(sub, start, end)	replace(old, new)	utcnow()	strftime(date, format)				
check_interval	Signal check frequency	decode()	rfind(sub, start ,end)	fromtimestamp(timestamp)	utcfromtimestamp(timestamp)				
exec_prefix	Root directory	encode()	rindex(sub, start, end)						
executable	Name of executable	endswith(sub)	rjust(width)						
exitfunc	Exit function name	expandtabs()	rpartition(sep)						
modules	Loaded modules	find(sub, start, end)	rsplit(sep)						
path	Search path	index(sub, start, end)	rstrip()						
platform	Current platform	isalnum() *	split(sep)						
stdin, stdout, stderr	File objects for I/O	isalpha() *	splittlines()						
version_info	Python version info	isdigit() *	startswith(sub)						
winver	Version number	islower() *	strip()						
<b>sys.argv</b> for \$ python foo.py bar -c qux --h		isspace() *	swapcase() *						
sys.argv[0]	foo.py	istitle() *	title() *						
sys.argv[1]	bar	isupper() *	translate(table)						
sys.argv[2]	-c	join()	upper() *						
sys.argv[3]	qux	ljust(width)	zfill(width)						
sys.argv[4]	--h	lower() *							
<b>os Variables</b>		<b>Note</b> Methods marked * are locale dependant for 8-bit strings.							
altsep	Alternative sep								
curdir	Current dir string								
defpath	Default search path								
devnull	Path of null device								
extsep	Extension separator								
linesep	Line separator								
name	Name of OS								
pardir	Parent dir string								
pathsep	Patch separator								
sep	Path separator								
<b>Note</b> Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"									
<b>Class Special Methods</b>									
__new__(cls)	__lt__(self, other)								
__init__(self, args)	__le__(self, other)								
__del__(self)	__gt__(self, other)								
__repr__(self)	__ge__(self, other)								
__str__(self)	__eq__(self, other)								
__cmp__(self, other)	__ne__(self, other)								
__index__(self)	__nonzero__(self)								
__hash__(self)									
__getattr__(self, name)									
__getattribute__(self, name)									
__setattr__(self, name, attr)									
__delattr__(self, name)									
__call__(self, args, kwargs)									
<b>Indexes and Slices (of a=[0,1,2,3,4,5])</b>									
	len(a)	6							
	a[0]	0							
	a[5]	5							
	a[-1]	5							
	a[-2]	4							
	a[1:]	[1,2,3,4,5]							
	a[:5]	[0,1,2,3,4]							
	a[:-2]	[0,1,2,3]							
	a[1:3]	[1,2]							
	a[1:-1]	[1,2,3,4]							
	b=a[:]	Shallow copy of a							
<b>Time Methods</b>									
	replace()	utcoffset()							
	isoformat()	dst()							
	__str__( )	tzname()							
	strftime(format)								
<b>Date Formatting (strftime and strptime)</b>									
	%a	Abbreviated weekday (Sun)							
	%A	Weekday (Sunday)							
	%b	Abbreviated month name (Jan)							
	%B	Month name (January)							
	%c	Date and time							
	%d	Day (leading zeros) (01 to 31)							
	%H	24 hour (leading zeros) (00 to 23)							
	%I	12 hour (leading zeros) (01 to 12)							
	%j	Day of year (001 to 366)							
	%m	Month (01 to 12)							
	%M	Minute (00 to 59)							
	%p	AM or PM							
	%S	Second (00 to 61 <sup>4</sup> )							
	%U	Week number <sup>1</sup> (00 to 53)							
	%w	Weekday <sup>2</sup> (0 to 6)							
	%W	Week number <sup>3</sup> (00 to 53)							
	%x	Date							
	%X	Time							
	%y	Year without century (00 to 99)							
	%Y	Year (2008)							
	%Z	Time zone (GMT)							
	%%	A literal "%" character (%)							
1. Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.									
2. 0 is Sunday, 6 is Saturday.									
3. Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.									
4. This is not a mistake. Range takes account of leap and double-leap seconds.									
Available free from <a href="http://AddedBytes.com">AddedBytes.com</a>									

PC Settings X

Search

> Appearance & Behavior

Keymap

Editor

- > General
- Font
- > Color Scheme
- > Code Style
- Inspections
- File and Code Templates
- File Encodings
- Live Templates**
- File Types
- > Emmet
- Images
- Intentions
- Language Injections
- Spelling
- TextMate Bundles
- TODO

Plugins

> Version Control

> Project: test1

> Build, Execution, Deployment

> Languages & Frameworks

> Tools

Editor > Live Templates

By default expand with **Tab** ▼

>  **Python**

- compd (Dict comprehension)
- compdi (Dict comprehension with 'if')
- compg (Generator comprehension)
- compgi (Generator comprehension with 'if')
- compl (List comprehension)
- compli (List comprehension with 'if')
- comps (Set comprehension)
- compsi (Set comprehension with 'if')
- iter (Iterate (for ... in ...))
- itere (Iterate (for ... in enumerate))
- main (if \_\_name\_\_ == '\_\_main\_\_')
- prop (Property getter)
- props (Property getter/setter)
- propsd (Property getter/setter/deleter)
- super ('super(...)' call)

>  **R**

>  **React**

No live templates are selected

OK Cancel Apply

	Cross Platform	Commercial/Fre e	Auto Code Completion	Multi-Langua ge Support	Integrat ed Python Debugging	Error Markup	Source Control Integrati on	Smart Indent	Bracket Matchin g	Line Numbering	UML Editing / Viewing	Code Folding	Code Templat es	Unit Testing	GUI Designe r (Qt, Eric, etc)	Integrat ed DB Support	Rapid Application	Development
Atom	Y	F			Y	Y	Y	Y	Y	Y		Y	Y					
BlackAdder	Y	C							Y			Y						
BlueFish	L																	
ConTEXT	W	C																
DABO	Y																	
DreamPie		F	Y				Y											
Dr.Python		F			Y													
Editra	Y	F	Y	Y			Y	Y	Y	Y		Y						
Emacs	Y	F	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			
Eric Ide	Y	F	Y		Y	Y		Y		Y		Y			Y			
E-Texteditor	W																	
Geany	Y	F	Y*	Y				Y	Y	Y		Y						*very limited
Gedit	Y	F	Y <sup>1</sup>	Y				Y	Y	Y			Y <sup>2</sup>					1with plugin; 2sort of
Idle	Y	F	Y		Y			Y	Y									
JEdit	Y	F		Y					Y	Y		Y						
KDevelop	Y	F		Y			Y	Y	Y	Y		Y						
Komodo	Y	CF	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y		
NetBeans*	Y	F	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	*pre-v7.0
NotePad++	W	F		Y				Y*		Y								*with plugin
Pfaide	W	C	Y	Y				Y	Y	Y		Y	Y					
PIDA	LW	F	Y	Y				Y	Y	Y		Y						VIM based
PTVS	W	F	Y	Y	Y	Y	Y	Y	Y	Y		Y			Y*		Y	*WPF bsed
PyCharm	Y	CF	Y	Y*	Y		Y	Y	Y	Y		Y		Y				*JavaScript
PyDev(Eclipse)	Y	F	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
Pyscripter	W	F	Y		Y	Y		Y		Y			Y	Y				
PythonWin	W	F	Y		Y			Y	Y	Y		Y						
SciTE	Y	F <sup>1</sup>		Y		Y		Y	Y	Y		Y	Y					1Mac version is commercial
ScriptDev	W	C	Y	Y	Y	Y	Y		Y	Y	Y		Y	Y				
SPE		F	Y								Y							
Spyder	Y	F	Y		Y	Y		Y	Y	Y								
Sublime Text	Y	CF	Y	Y				Y	Y	Y		Y	Y	Y*				extensible w/Python,

# Ktoré GUI a prečo?



*wxPython*



# NetBeans problém autocomplet

## Bug 153261 - Autocompletion does not work as it should

Status: NEW

Product: python

Component: Editor

Version: 6.x

Hardware: All All

Priority: P2 with 2 votes (vote)

Target Milestone: TBD

Assigned To: Torbjorn Norbye

QA Contact: nbpythonqa

URL:

Whiteboard:

Keywords:

Depends on:

Blocks:

Show dependency tree / graph

**Reported:** 2008-11-15 20:27 UTC by erdincyilmazel

**Modified:** 2015-11-27 17:57 UTC ([History](#))

**CC List:** 0 users

See Also:

Issue Type: DEFECT

**Exception Report :**



# Eclipse PyDev



**LiClipse**

get more from Eclipse ...

## What is PyDev?

PyDev is a **Python IDE** for **Eclipse**, which may be used in **Python**, **Jython** and **IronPython** development.

It comes with many goodies such as:

- Django integration
- Code completion
- Code completion with auto import
- Type hinting
- Code analysis
- Go to definition
- Refactoring
- Debugger
- Remote debugger
- Find Referrers in Debugger
- Tokens browser
- Interactive console
- Unittest integration
- Code coverage
- Find References (Ctrl+Shift+G)
- **and many others:**

A screenshot of the PyDev IDE interface. It shows a Python script with code for a Robot class and a TestRobot class. A tooltip is displayed over the 'walk' method in the TestRobot class, listing three options: 'Create walk method at Robot', 'Assign to field (self.walk)', and 'Assign to local (walk)'. The background of the IDE has a light gray theme with some darker gray sections for code blocks.

## PyDev development

PyDev is open source and depends on **you** to help it grow. You can contribute in the form of bug fixes, answers to questions, new features... Another option is financial support via PayPal:

\$7

\$20

\$45

\$100

Corporate sponsorship is also available for your company.

## Search PyDev-related content

Google™ Custom Search



Favorite at:  
**eclipse  
marketplace**



IT ACADEMY

# Asercia

Testovanie programu

Princíp raise-if

Ak je výraz True,  
pokračuje  
sa vo vykonávaní  
príkazov

Ak je výraz False,  
vyvolá sa  
výnimka  
**AssertionError**

```
def kelvinNaFahrenheit(teplota):  
    assert (teplota >= 0), "Menej ako absolutna 0!"  
    return ((teplota-273)*1.8)+32
```

```
print kelvinNaFahrenheit(100)  
print int(kelvinNaFahrenheit(500.55))  
print kelvinNaFahrenheit(-5)
```

# Spracovanie výnimiek

- Try
- Except
- Try
- Finally
- Try
- Except
- Finally

```
try:  
    fh = open("testfile", "w")  
    fh.write("Toto je moj testovaci subor...")  
except IOError:  
    print "Chyba: nemozem najst subor alebo citat data"  
else:  
    print "Obsah uspesne zapisany"  
    fh.close()
```

# Otváranie a zatváranie súborov

```
f = open('data.txt')  
try:  
    data = f.read()  
finally:  
    f.close()  
  
with open('data.txt') as f:  
    data = f.read()
```

# Odstraňovanie súborov

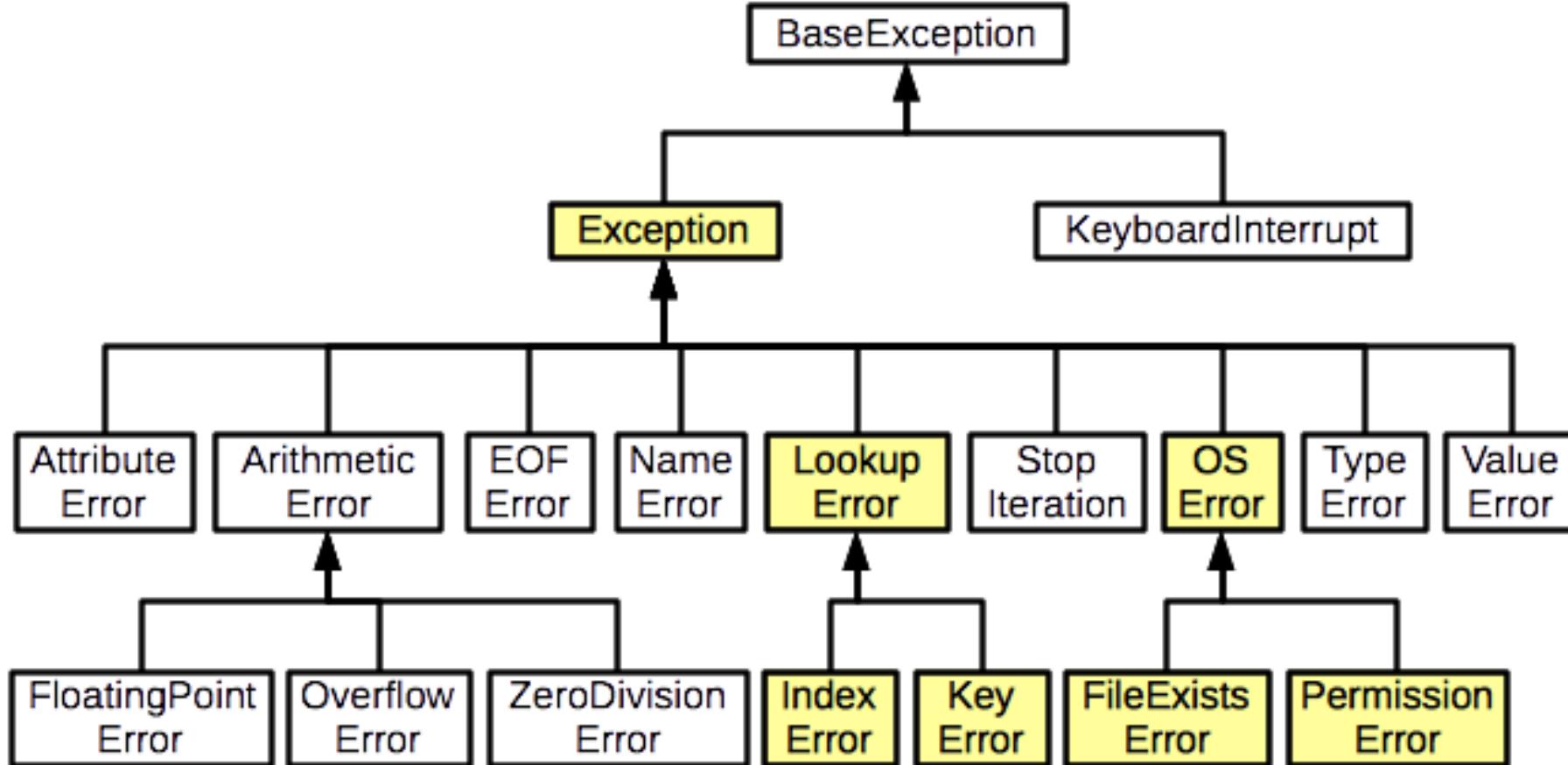
```
try:  
    os.remove('somefile.tmp')  
except OSError:  
    pass  
  
with ignored(OSError):  
    os.remove('somefile.tmp')
```

# Spracovanie výnimiek

UnboundLocalError  
AssertionError  
EOFError KeyError  
IOError SyntaxError  
SystemExit FloatingPointError  
StopIteration OverflowError  
StandardError  
KeyboardInterrupt  
ZeroDivisionError  
ImportError  
Exception

```
def vypocitajCenuDPH(cena):  
    try:  
        return int(cena)  
    except ValueError as exVal:  
        print "Argument neobsahuje cislo"  
        print exVal  
        ## print exVal.message  
    except TypeError as exType:  
        print "Nezadal si argument..."  
  
print vypocitajCenuDPH("xyz")  
print vypocitajCenuDPH(100)  
print vypocitajCenuDPH(None)  
print vypocitajCenuDPH()
```

# Diagram tried výnimiek



Multiexcept: `except (exception1, exception2) as e`

# Premenné triedy

```
class Zamestnanci:  
    pocetZam = 0  
  
    def __init__(self, meno, plat):  
        self.meno = meno  
        self.plat = plat  
        Zamestnanci.pocetZam += 1  
  
    def zobrazPocet(self):  
        print "Pocet zamestnancov %d" % Zamestnanci.pocetZam  
  
    def zobrazZamestnancov(self):  
        print "meno : ", self.meno, ", plat: ", self.plat
```



# Iterátor

Sekvencia automaticky vytvorí iterátor:

- **for i in sekvencia:** urob\_nieco(i)

Čo je ekvivalentné:

1. m = iter(sekvencia)
2. **while** 1:
3. **try**:
4.     i = m.next()
5. **except** StopIteration:
6.     **break**
7.     urob\_nieco(i)

Iterator	Arguments	Results	Example
count()	start, [step]	start, start+step, start+2*step, ...	count(10) --> 10 11 12 13 14 ...
cycle()	p	p0, p1, ... plast, p0, p1, ...	cycle('ABCD') --> A B C D A B C D ...
repeat()	elem [,n]	elem, elem, elem, ... endlessly or up to n times	repeat(10, 3) --> 10 10 10

# Getre a setre

```
class Zamestnanec:  
    pocetZam = 0  
  
    def __init__(self, meno, plat):  
        self.meno = meno  
        self.plat = plat  
        Zamestnanec.pocetZam += 1  
  
    def zobrazPocet(self):  
        print "Počet zamestnancov %d" % Zamestnanec.pocetZam  
  
    def zobrazZamestnancov(self):  
        print "meno : ", self.meno, ", plat: ", self.plat  
  
zam1 = Zamestnanec("Adam", 2000)  
zam2 = Zamestnanec("Eva", 5000)
```

```
##print zam1.vek  
zam1.vek = 7 # Vytvor atribut vek  
print zam1.vek  
zam1.vek = 8 # Modifikuj atribut vek  
print zam1.vek  
del zam1.vek # Vymaz atribut vek  
  
print "Ma zamestnanec vek:", hasattr(zam2, "vek")  
setattr(zam2, "vek", 8) # vy  
print getattr(zam2, "vek")  
delattr(zam2, "vek")
```

Príkaz del

# Zabudované atribúty triedy

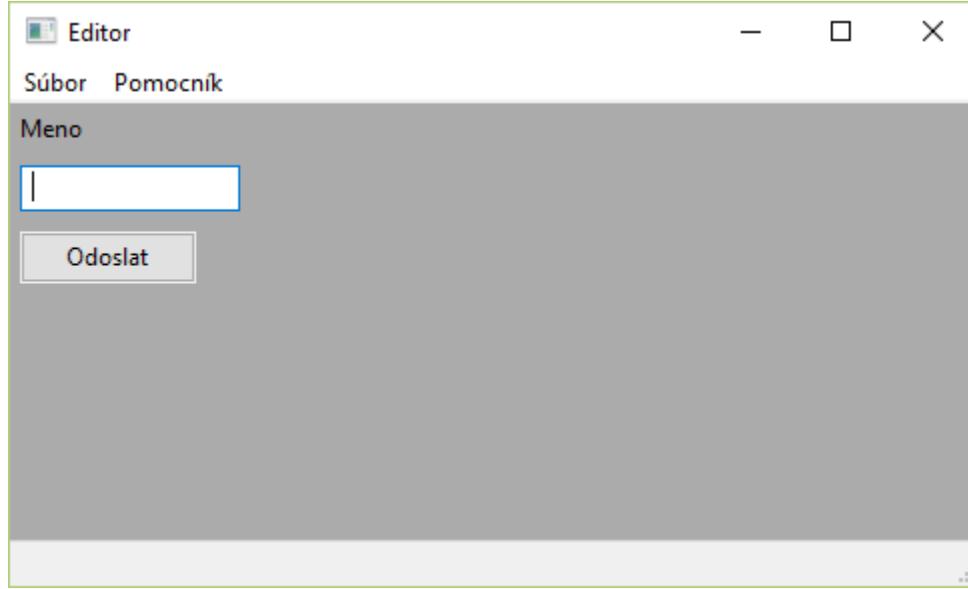
- **\_\_dict\_\_**: Dictionary containing the class's namespace.
- **\_\_doc\_\_**: Class documentation string or None, if undefined.
- **\_\_name\_\_**: Class name.
- **\_\_module\_\_**: Module name in which the class is defined. This attribute is "\_\_main\_\_" in interactive mode.
- **\_\_bases\_\_**: A possibly empty tuple containing the base classes, in the order of their occurrence in the base class list.

# Knižnice

```
import math as matematika  
  
def vypisAhoj ():  
    print "Ahoj novy svet"  
  
meno = "Karol"
```

```
>>> vypisAhoj()  
Ahoj svet  
>>> meno  
'Karol'  
>>> import kniznica  
>>> vypisAhoj()  
Ahoj svet  
>>> kniznica.vypisAhoj()  
Ahoj novy svet  
>>> reload(kniznica)  
<module 'kniznica' from 'C:/Users/Miroslav/Desktop\kniznica.pyc'>  
>>> vypisAhoj()  
Ahoj svet
```

# GUI



```
if __name__ == '__main__':
    app = wx.App(0)
    frame = frmEditor(None)
    frame.Show()
    app.MainLoop()
```

# Zámky/locky

```
# Make a lock
lock = threading.Lock()

# Old-way to use a lock
lock.acquire()
try:
    print 'Critical section 1'
    print 'Critical section 2'
finally:
    lock.release()

# New-way to use a lock
with lock:
    print 'Critical section 1'
    print 'Critical section 2'
```

# Dokumentácia

The screenshot shows a web browser window displaying the wxPython 2.8.9.2 documentation for the `wx.Frame` class. The URL is `https://wxpython.org/docs/api/wx.Frame-class.html`. The page title is "wx.Frame". The top navigation bar includes "Home", "Trees", "Index", and "Help". The right side of the header shows "wxPython 2.8.9.2" and "[frames | no frames]".

**Type Frame**

```
object --+
      |
      Object --+
      |
      EvtHandler --+
      |
      Window --+
      |
      TopLevelWindow --+
      |
      Frame
```

**Known Subclasses:**

`AdvancedSplash, AuiMDIParentFrame, BalloonFrame, DocChildFrame, DocParentFrame,`  
`DocTabbedParentFrame, EventWatcher, FillingFrame, Frame, HtmlHelpFrame,`  
`InspectionFrame, MDIChildFrame, MDIParentFrame, MiniFrame, PreviewFrame, SizedFrame,`  
`SplashScreen, SplashScreen, TestFrame, ToasterBoxWindow`

---

Proxy of C++ Frame class

**Method Summary**

	Frame
	<code>_init_(self, parent, id, title, pos, size, style, name)</code>
	<code>bool Command(self, winid)</code>
	<code>bool Create(self, parent, id, pos, size, style, name)</code> Create the GUI part of the Window for 2-phase creation mode.
	<code>StatusBar CreateStatusBar(self, number, style, winid, name)</code>
	<code>wxToolBar CreateToolBar(self, style, winid, name)</code>
	<code>DoGiveHelp(self, text, show)</code>

# Čo sa oplatí prečítať?

## Slovensko a česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

## Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

## YouTube tutoriály

- [IT Academy](#)

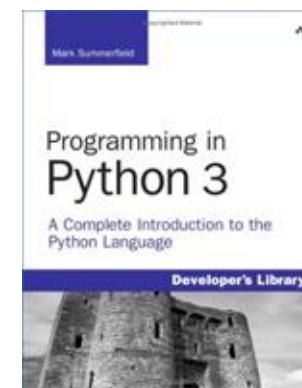
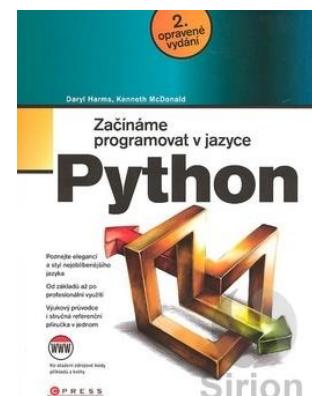
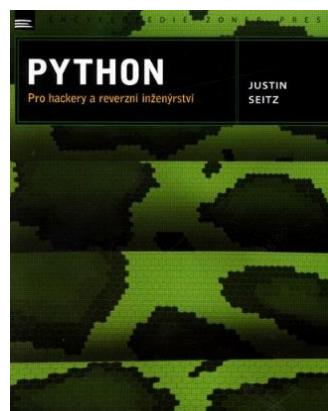
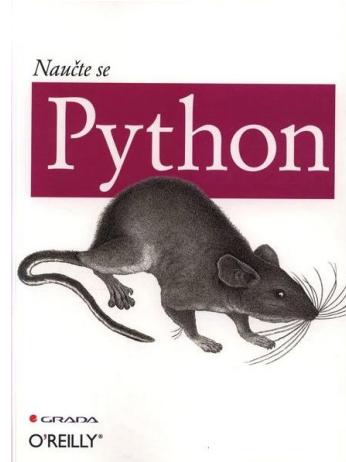
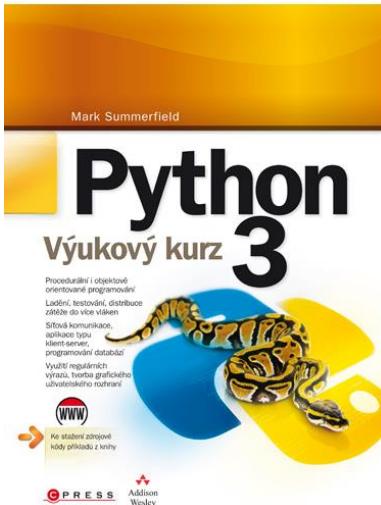
# Čo ti odporúčam si pozrieť?

1. <https://docs.python.org/3/>
2. <https://realpython.com/tutorials/best-practices/>
3. <https://google.github.io/styleguide/pyguide.html>
4. <https://docs.python.org/3/>
5. <http://python2013.input.sk/19prednaska>
6. <https://realpython.com/python3-object-oriented-programming/>
7. <https://jeffknupp.com/blog/2014/06/18/improve-your-python-python-classes-and-object-oriented-programming/>
8. <https://overiq.com/python-101/inheritance-and-polymorphism-in-python/>
9. <https://www.javatpoint.com/python-oops-concepts>
10. <https://www.programiz.com/python-programming/object-oriented-programming>



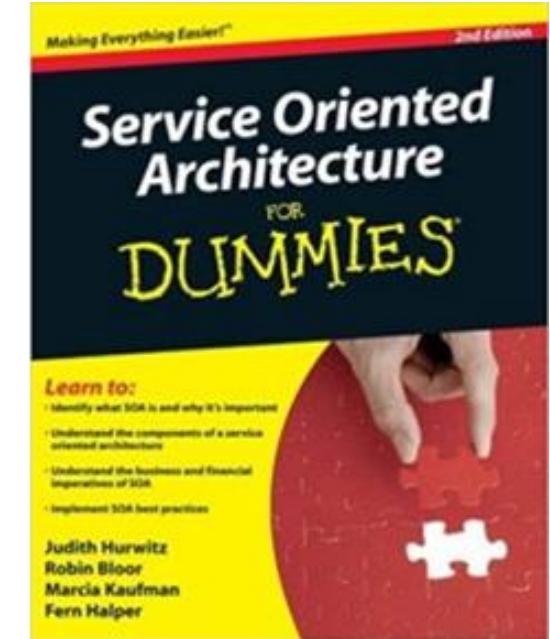
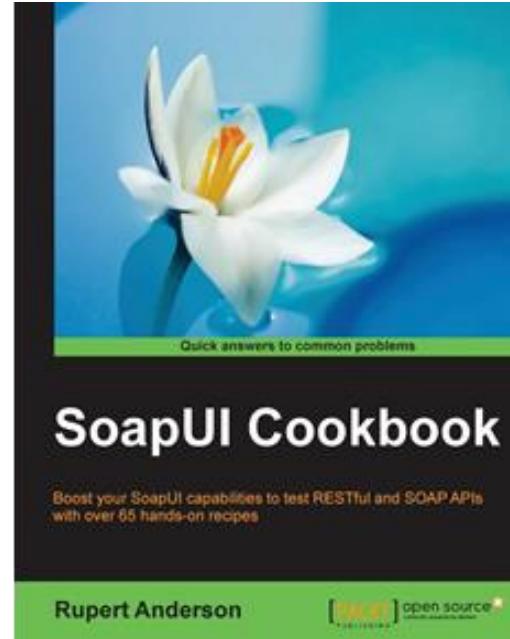
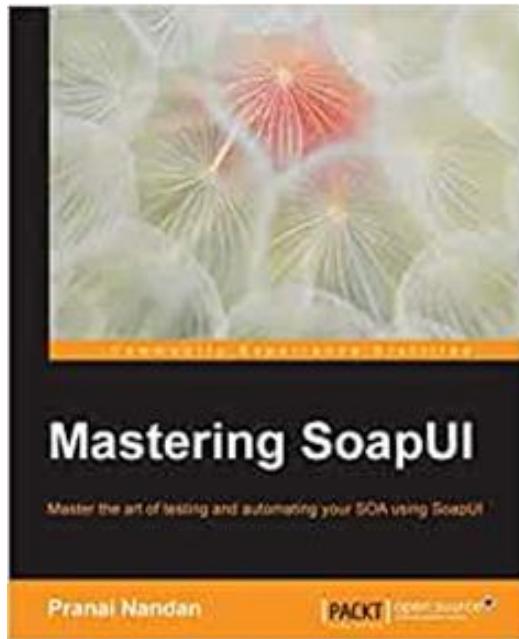
Šup do záložiek

# Čo sa oplatí/neoplatí prečítať SK/CZ?



Mark Pilgrim

# Čo sa oplatí/neoplatí prečítať EN?



# I am programmer



I have Life



I have  
stackoverflow



IT ACADEMY

Home

PUBLIC

Questions

**Tags**

Users

COLLECTIVES

Explore Collectives

FIND A JOB

Jobs

Companies

TEAMS

Create free Team

## Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

python

**python**

Python is a multi-paradigm, dynamically typed, multi-purpose programming language. It is designed to be quick to learn, understand, and...

1870168 questions 695 asked today, 6387 this week

**python-3.x**

USE ONLY IF YOUR QUESTION IS VERSION-SPECIFIC. For questions about Python programming that are specific to version 3+ of the language...

303562 questions 93 asked today, 836 this week

**python-2.7**

Python 2.7 is the last major version in the 2.x series, and is no longer maintained since January 1st 2020. Use the generic [python] tag on all Python...

94965 questions 24 asked this week, 106 this month

[Popular](#) [Name](#) [New](#)**python-requests**

USE ONLY FOR THE PYTHON REQUESTS LIBRARY. Requests is a full-featured Python HTTP library with an easy-to-use, logical API.

18697 questions 8 asked today, 57 this week

**python-imaging-library**

The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's API is lightweight but...

7883 questions 5 asked today, 38 this week

**wxpython**

wxPython is a Python wrapper for the cross-platform C++ GUI API wxWidgets.

7047 questions 7 asked this week, 14 this month

**ipython**

IPython is a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook.

6886 questions 5 asked this week, 26 this month

**python-3.6**

Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more generic [python] and [python-3....]

5602 questions 11 asked this week, 24 this month

**python-asyncio**

to be used for the asyncio Python package which provides mechanisms for writing single-threaded concurrent code. The asyncio package provides...

5492 questions 29 asked this week, 125 this month

**python-import**

For questions about importing modules in Python

5119 questions 11 asked this week, 47 this month

**python-multiprocessing**

multiprocessing is a package that supports spawning processes using an API similar to the threading module in python programming language.

4036 questions 12 asked this week, 46 this month

**python-3.7**

Version of the Python programming language released in June 27, 2018. For issues that are specific to Python 3.7. Use the more generic [python] and...

4034 questions 5 asked this week, 21 this month

# Efektívne používanie klávesnice

**Špeciálne znaky, kde ich nájst' na klávesnici**

The diagram shows a standard QWERTY keyboard with a focus on the numpad and function keys. Special characters are highlighted in various colors (yellow, green, blue, red) to indicate their locations. To the right of the keyboard, there are large, semi-transparent symbols representing common special characters: '#', '&', '!', and '€'. Below the keyboard, a legend provides a key-to-meaning mapping for each highlighted character.

Operátory	Porovnávanie	Oddelovače	Bitové operácie	Zátvorky
+	< >	, Prvkov	& Prienik, A, AND	( ) Zátvorky, Volanie
*	=	. Atribútov	Zjednotenie, OR	{ } Slovníky, Formát
-	! Nerovnosť	: Blokov, Klúčov	<sup>^</sup> XOR	[ ] Zoznamy, Indexy
/	Retázce	; Príkazov	~ Inverzie	Ostatné
%	' " Úvodzovky	Poznámky		Súčasť mena
@	\ Špeciálne znaky	# Komentár	? Pomocník	\$ Nevyužité

# Najdôležitejšie klávesové skratky

## Práca s IDE

- Ctrl + D Delete zmaž riadok
- **Ctrl + Space** Asistent kódu
- **Ctrl + /** Komentáre
- Ctrl + A Označ všetko
- **Alt + /** Dokonči slovo
- Ctrl + F Hľadanie a náhrady
- Ctrl + Shift + F Kompakt režim
- Ctrl + Shift + S Ulož všetko

## Práca s browserom

- Ctrl + T Vytvor nový tab
- Ctrl + W Zatvor aktuálny tab
- Ctrl + Shift + W Zatvor všetky taby
- **Ctrl + Shift + T** Otvor posledný tab
- Ctrl + Shift + J/F12 Web console
- **F11** Fullscreen

**F5 nie je spustenie, ale Refresh**

# PYCON SK 2022

Bratislava

KÚP SI LÍSTOK



Vývojári



Miroslav

Domov

Vytvoriť



## Vývojári

Verejná skupina

Informácie

## Diskusia

Oznámenia

Členovia

Podujatia

Videá

Fotky

Súbory

Hľadať v tejto skupine



Ste člen

Upozornenia

Zdieľať

... Viac



Napísat príspěvok

Pridať fotku/video

Živé video

Viac



Napište niečo...



Fotka/video



Divácka páry



Označiť priateľov

...

## Skratky

Podnikanie na Slovensku

2

UK Manazment Externe...

Testovacia firma

VITA - Virtual It Academy

Startupisti

NOVÁ AKTIVITA



Roland Mondek

10 h

## POZVĀŤ ČLENOV

+ Zadajte meno alebo e-mailovú adresu...



## ČLENOVIA

5 505 členov



## POPIS

Skupina softvérových vývojárov. Táto skupina by mala byť miestom... Zobraziť viac

## TYP SKUPINY

Všeobecné

## VAŠE STRÁNKY



IT Academy



VITA - Virtual It Academy

## KONTAKTY



Evka Rybárska



Jarmila Palenčárová



Stefan Orosi



Ivana Ivka Jasaňová



Hrá Word Blitz



Ivana Pavlíková



Martin Vanko



Lucia Kovačičová

4 h



Lošák Filip



Andrej Nejedlik



Gabika Zubrikova

## SKUPINOVÉ KONVERZÁCIE



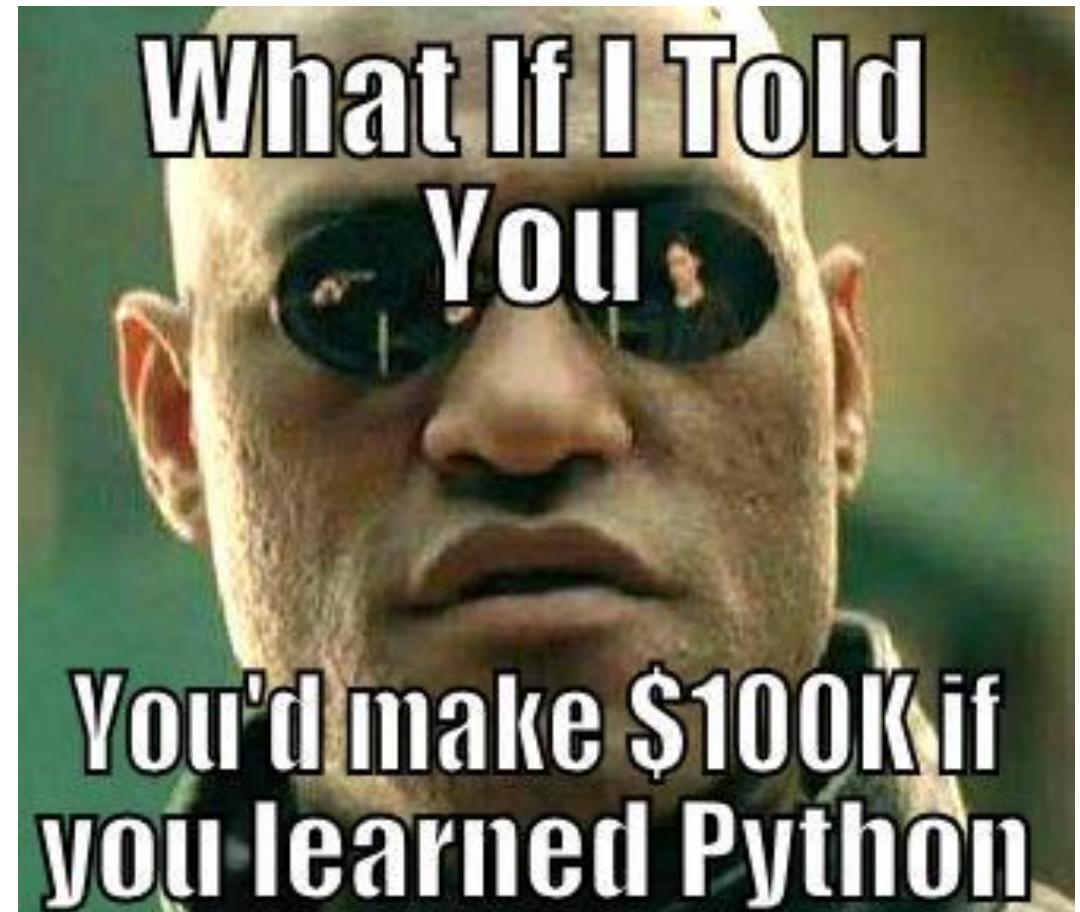
Vytvoriť novú skupinu

Hľadať



# Zen filozofia Pythonu

1. Krásny je lepší než škaredý
2. Explicitný je lepší ako implicitný
3. Jednoduchý je lepší ako zložitý
4. Zložitý je lepší ako komplikovaný
5. Plochý je lepší ako vnorený
6. Riedky je lepší ako hustý
7. Na čitateľnosti záleží
8. Praktickosť vyhráva nad čistotou



import this

# Čaká nás krásna budúcnosť

```
>>> from __future__ import braces  
SyntaxError: not a chance (<pyshell#13>, line 2)  
>>> |
```

No future {} a ;



# Inšpirácia na projekty

# Python Project Ideas

Easy



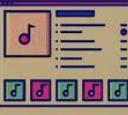
Quote Gener. Number guessing Dice Simulation YT downloader

Mid



Password Manag. Mario Party Web Crawler Email Autom.

Pro



Music Player Face Detection Twitter Clone Twitter Bot

@TheInsaneApp

# Ako skončíme?

## 1. Pridaj si ma na LinkedIn

- [www.linkedin.com/in/miroslav-reiter](http://www.linkedin.com/in/miroslav-reiter)

## 2. Materiály po prednáške

- <https://1drv.ms/p/s!AlrLrycbTQ1a19sf c1MmNNnYMaluWA?e=FTUITc>

The screenshot shows the LinkedIn SlideShare interface. At the top, there is a navigation bar with the LinkedIn logo, the word "SlideShare", a search bar, and a user profile icon. Below the navigation bar, there are tabs for "Home" and "Explore". On the left side, there are buttons for "My Uploads", "My Comments", and "Analytics". A dropdown menu for "Most Recent" is open. There are two main presentation cards displayed:

- Automatizácia a optimalizácia Firemné procesy**  
by Ing. Mgr. Miroslav Reiter, DSc., N  
[miroslav.reiter@...](#)  
1 month ago, 11 slides
- Najväčšie faily na sociálnych sietach a Google reklame**  
by Ing. Mgr. Miroslav Reiter, DSc., N  
[miroslav.reiter@...](#)  
6 months ago, 63 slides

Below each presentation card, there are icons for views (113 or 19), likes (0), comments (0), and downloads (0).



Našim záväzkom je **prispiet' k oživeniu slovenskej ekonomiky** a do konca roka 2021

**pomôcť ďalším 50,000 slovenským firmám a jednotlivcom**

lepšie využívať internet k rastu svojho podnikania, rozvoja kariéry či nájdenia novej práce.

## Online marketing

Google Analytics

Online marketing strategy

Google for Nonprofits

Shopping

YouTube

Google My Business

Google Ads

## Technology & Tools

Workspace (G Suite)

Online Security & Safety

Google for Education

AI/ML

## Business & Soft skills

Entrepreneurship

Leadership

Export

#IamRemarkable\*

Entrepreneurship / Diversity for women

Critical thinking / Media literacy



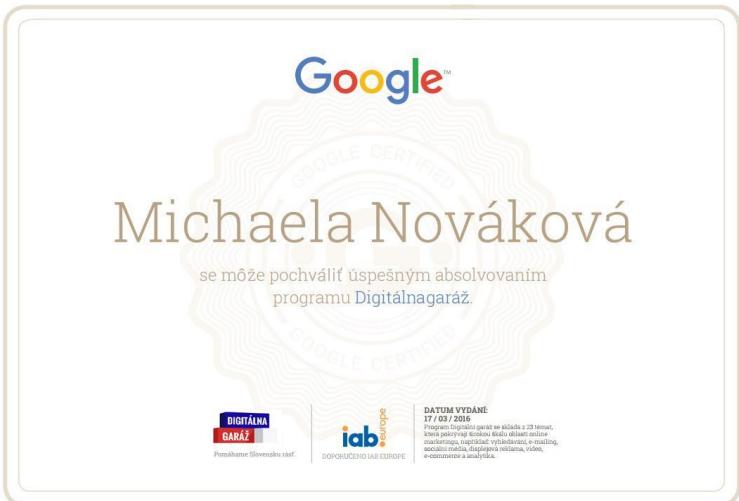
Digitálna garáž Online platforma na výuku digitálneho marketingu a mäkkých zručností

26 tém

106 lekcií

9 hodín obsahu

Dostupné 24/7  
Zadarmo  
Na mieru



≡ | Google Digitálna garáž

1. Základy e-mailového marketingu  
Téma: Využite e-mailové spojenie

☆ ZÁLOŽKA ⌂ ZDIELAŤ:

Prehľad tém

Lekcie 0 / 4

☆ 1. Základy e-mailového marketingu

Sledujte lekcii 6 min

Otestujte si svoje znalosti 1 min

☆ 2. Možnosti e-mailového marketingu

☆ 3. Vytváranie skvelých marketingových e-mailov

☆ 4. Správa úspešných e-mailových kampaní

PRESKOČIŤ NA TEST

YouTube

0:01 / 5:04

ZOBRAZIŤ PREPIS

OTESTUJTE SI SVOJE ZNALOSTI

Hlavné poznatky

Zasielanie bulletinov a akčných ponúk zákazníkom prostredníctvom e-mailu môže zohrať kľúčovú úlohu vo vašom marketingovom pláne. Budujte a upevňujte vzťahy so zákazníkmi. V tomto videu sa pozrieme na:

- vytváranie zoznamu kontaktov,
- zacielenie na publikum na základe záujmov,
- budovanie vzťahov so zákazníkmi.

## Spracovanie a vizualizácia dát v Pythone

Základy dátovej analýzy, spracovanie a vizualizácia dát v programovacom jazyku Python.

Prerekvizitou tohto kurzu sú základné zručnosti v programovacom jazyku Python. Pokiaľ ste absolvovali predošlý kurz s názvom „Základy programovacieho jazyka Python“ prípadne „Objektovo orientované programovanie v Pythone“, určite splňate základné prepoklady pre absolvovanie tohto kurzu.

Opäť (ako v predchádzajúcich kurzoch) budeme pracovať v prostredí Jupyter Notebook, ktorý si môžete nainštalovať aj doma na svojom osobnom počítači prostredníctom GUI Anaconda Navigator.

### 1. Základný balík NumPy:

- Nainštalovanie knižnice NumPy
- Vytvorenie NumPy polí
- Dátové typy a operácie s NumPy poľami
- Indexovanie a prechádzanie NumPy polí
- Univerzálné NumPy funkcie
- Spracovanie a filtrovanie NumPy polí
- Zhrnutie nových znalostí

### 2. Vizualizácia dát:

- Nainštalovanie knižnice pre vizualizáciu dát
- Úvod do vizualizácie v knižničach Matplotlib a Seaborn
- Vizualizácia dát na rozličných príkladoch v spomínaných knižničach
- Prispôsobenie(Customization) výstupov grafov podľa našej potreby

### 3. Spracovanie dát:

- Nainštalovanie a import Pandas knižnice
- Vytváranie Pandas dataframov
- Načítanie súborov (.txt, .xlsx, .csv)
- Spracovanie a analýza dát zo súborov(.txt, .xlsx, .csv)
- Operácie s dátami
- Vytváranie grafov
- Zhrnutie

#### Trvanie:

10 hodín (2 dni)

#### Cena:

€0,-

#### Kategória:

Python

#### Registrácia (počet prihlásených):

18. - 19. 5. 2022 (43/50)

# Vyber si online kurz

Nauč sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zameraj na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia tvojho domova.

Ročné  
predplatné na  
všetky online  
kurzy

~~2299.99€~~

399.99€

Priístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

Kúpiť teraz

## Zadarmo

1. Kurzy SAV

2. Kurzy Grow with Google

3. YouTube kanál IT Academy

<https://www.youtube.com/c/IT-AcademySK>

Platené  
Moje kurzy na [www.vita.sk](http://www.vita.sk)