

Python

Objektové Programovanie



Ako začneme?

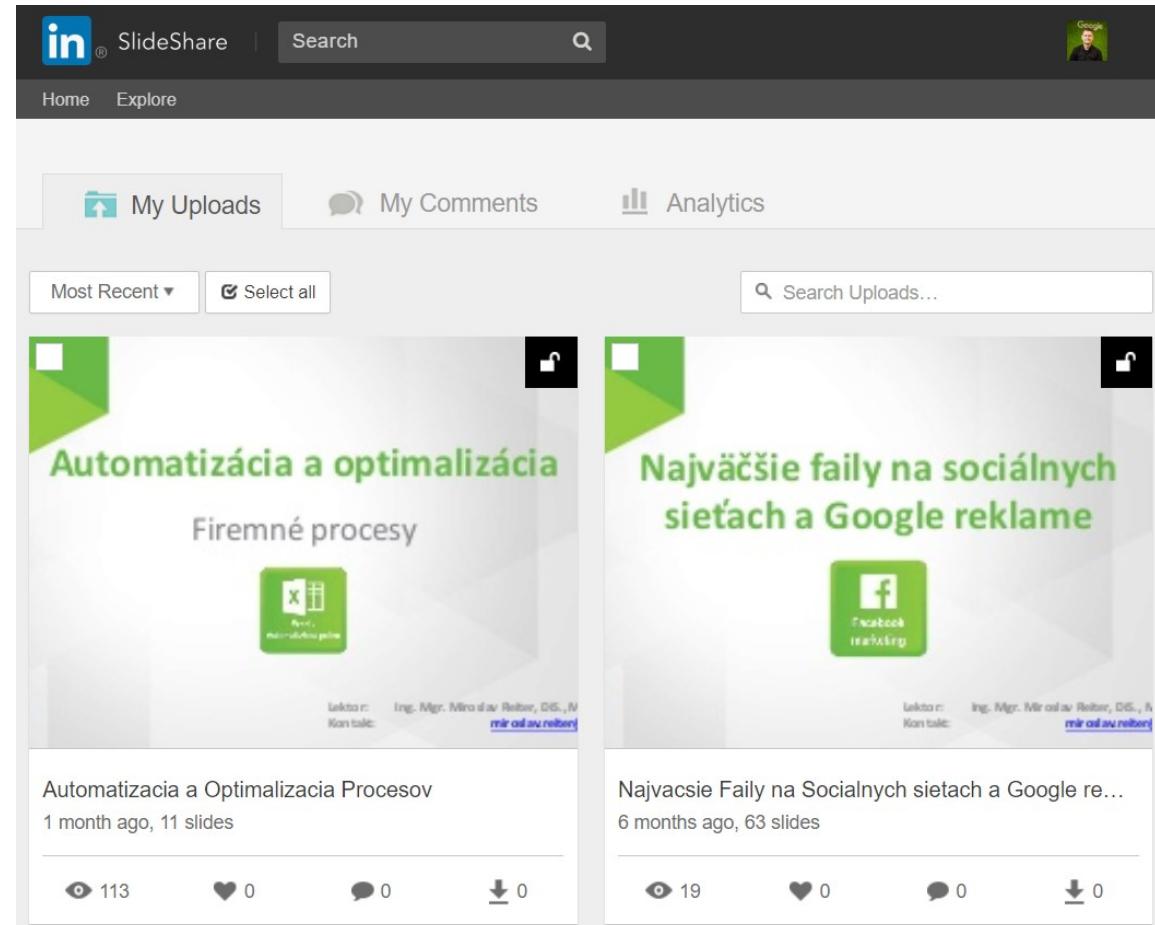
1. Maj nainštalovaný Anaconda Navigator

- <https://www.anaconda.com/products/>

2. Pridaj si ma na LinkedIn

- www.linkedin.com/in/miroslav-reiter

3. Prezentácia a materiály po prednáške



Domov

Ponuka kurzov

Ponuka prednášok

GPU programming in CUDA

[Home](#) » Python - Objektové programovanie

Python - Objektové programovanie

Kurz sa zameriava na pokročilejšie techniky práce s programovacím jazykom Python. Nosnou tému kurzu bude **objektovo orientované programovanie (OOP)**.

Zoznámime sa so základnými princípmi OOP:

1. Dedičnosť
2. Zapuzdrenie
3. Polymorfizmus

Ukážeme si ako v Pythone vytvoriť triedu, inštanciu na danú triedu, ako s nimi pracovať a využívať ich pri tvorbe programu. Zistíme načo slúži v triede konštruktor danej triedy, ako sa vyžíva pri dedičnosti tried a polymorfizme. Ukážeme si rôzne spôsoby dedičnosti, ich výhody a nevýhody v rôznych situáciach. Kurz predstavuje ucelený vstup do objektovo orientovaného programovania a oboznámenie sa so základnými pojvmi tohto prístupu v prostredí jazyka Python.

Trvanie:

10 hodín (2 dni)

Cena:

€0,-

Kategória:

Python

Registrácia (počet prihlásených):

27. - 28. 4. 2022 (43/50)

Moodle Slovenskej akadémie vied

Dostupné kurzy

[Základy programovacieho jazyku Python](#)

Učiteľ: [Miroslav Reiter](#)

Python je open-source, objektovo-orientovaný, vysoko úrovňový programovací jazyk. Python beží na mnohých variantách Unixu, na Macu, aj na Windows (súčasťou kurzu bude inštalácia na ľubovoľnom systéme). Na kurz je potrebné mať vlastné PC (s ľubovoľným operačným systémom podporujúcim Python).

[Základy programovacieho jazyku Python 2](#)

Učiteľ: [Miroslav Reiter](#)

[Objektovo orientované programovanie v Pythone](#)

[Spracovanie a vizualizácia dát v Pythone](#)

V kurze Spracovanie a vizualizácia dát v Pythone bude poslucháč oboznámený knižnicami NumPy, Pandas a Matplotlib.

[Neurónové siete s knižnicou TensorFlow](#)

Kurz je zameraný na výučbu hlbokého učenia v jazyku Python. Poslucháčov naučíme pracovať s knižnicou Tensorflow. Tensorflow je aktuálne najpopulárnejšia knižnica zameraná na vytváranie, trénovanie a testovanie programov hlbokého učenia. Odporúčanou podmienkou pre absolvovanie tohto kurzu je účasť na kurzoch: Základy programovacieho jazyku Python, Objektovo orientované programovanie v Pythone, Spracovanie a vizualizácia dát v Pythone.

ZPP

Účastníci

Odznaky

Kompetencie

Známky

Všeobecné

Základné informácie

Premenné a typy dát

Operátory

Podmienky a cykly

Funkcie

Práca so súbormi

Objekty

Nástenka

Domovská stránka

Kalendár

Súkromné súbory

Content bank

Moje kurzy

Základy programovacieho jazyku Python

[Zapnúť upravovanie](#)

Oznámenia

Informácie o kurze



Sylabus kurzu



Študijné materiály



Jupyter



login: student

heslo: Jupyter2021

Základné informácie

Prednáška



Hello World



Premenné a typy dát

Premenné a typy dát



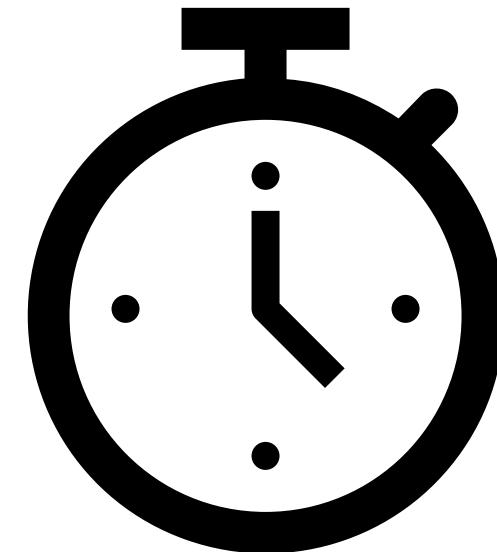
Operátory

Operátory



Úvodné informácie

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



O lektorovi - Miroslav Reiter

10000+
klientov a
500+ firiem

Programátor,
Analytik,
Manažér

Google,
Microsoft
ISTQB tréner

114
certifikácií

83 príručiek a
publikácií

12 škôl

52 projektov

Vlastná firma

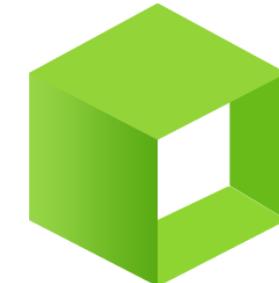


Miroslav Reiter

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



DIGITÁLNA
UNIVERZITA



IT ACADEMY





SPŠD TRNAVA



FAKULTA MANAGEMENTU
Univerzita Komenského
v Bratislave



Kde učím a vzdelávam?

Vyber 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 tvojho domova.

**Ročné
predplatné na
všetky online
kurzy**

~~2299.99€~~

399.99€

Prístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

🛒 **Kúpiť teraz**



260 kurzov v ponuke



Zábavné online lekcie



Akreditované kurzy



9 rokov skúseností



Certifikovaní profesionálni lektori

Moje začiatky

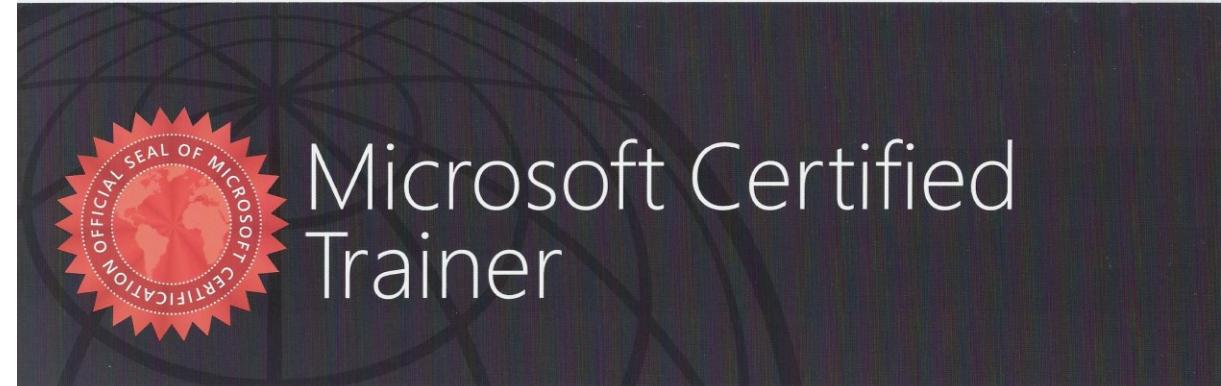


Miroslav Reiter

získava status
Google Certified Trainer

Automation

Google



MIROSLAV REITER

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

N. S. Nadella
Satya Nadella
Chief Executive Officer

Microsoft
CERTIFIED
Trainer

Čo robíte?

1. Študent/učiteľ

2. Zamestnanec

3. Podnikateľ

4. Nezamestnaný/materská

5. Dievča pre všetko



National competence centre for high performance computing
SLOVAKIA



EURO



EuroHPC
Joint Undertaking





Vzdelávanie

Kurzy:
itkurzy.sav.sk



Propagácia

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



HPC služby

Prístup k
výpočtovým
prostriedkom



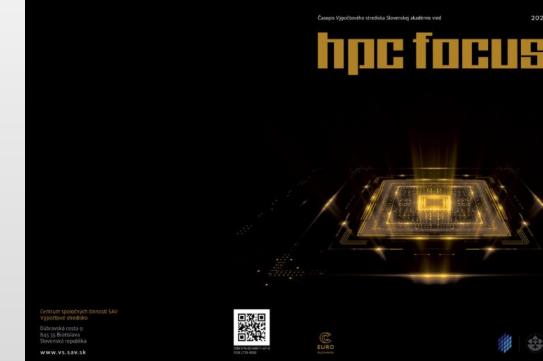
Mapovanie HPC prostredia

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)

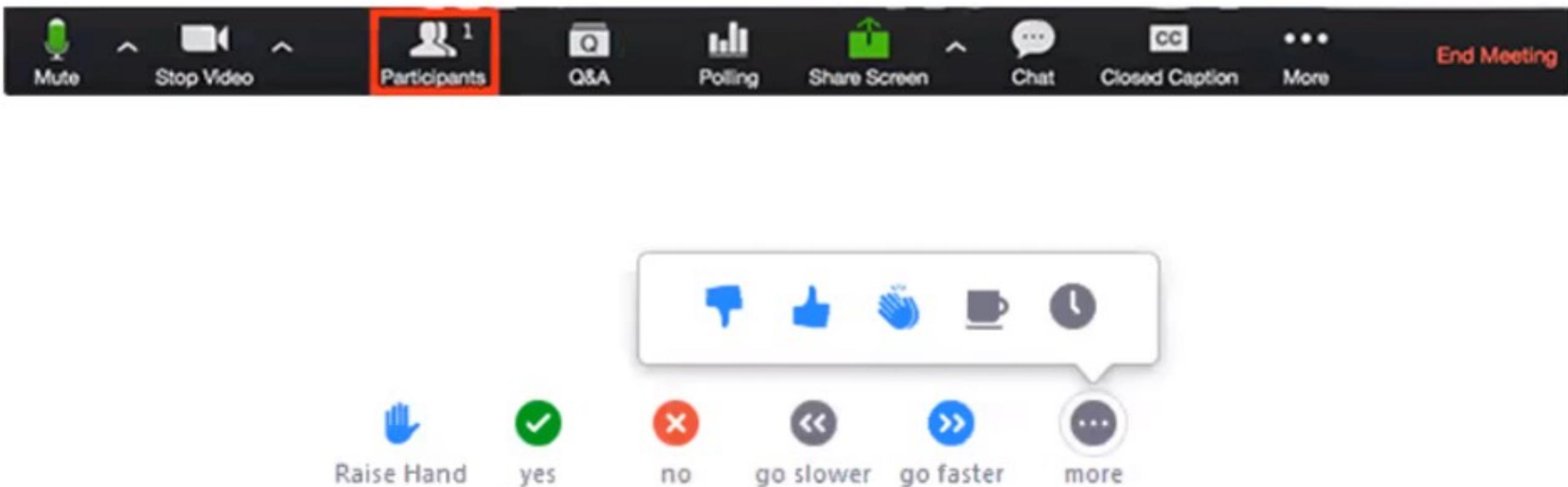
Find a participant

Participant	Color Swatches	Microphone (Green)	Video (Blue)
MR 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. Základy objektového myslenia

3. Základy OO programovania

4. Doplniť si znalosti z programovania

5. Doplniť si znalosti z Anaconda Navigator a Jupytera

Zábava je v zaručená v každom bode :-)



Čo je OOP a na čo je dobré?



nbextension variable inspector

- **Install**

- `pip install jupyter_contrib_nbextensions`
- `jupyter contrib nbextension install --user`

- **Enable**

- `jupyter nbextension enable varInspector/main`

Len priamo z CMD



Configurable nbextensions

disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)

filter: by description, section, or tags ×

- | | | | |
|--|--|--|---|
| <input type="checkbox"/> (some) LaTeX environments for Jupyter | <input type="checkbox"/> 2to3 Converter | <input type="checkbox"/> AddBefore | <input checked="" type="checkbox"/> Autopep8 |
| <input type="checkbox"/> AutoSaveTime | <input type="checkbox"/> Autoscroll | <input type="checkbox"/> Cell Filter | <input type="checkbox"/> Code Font Size |
| <input type="checkbox"/> Code prettify | <input type="checkbox"/> Codefolding | <input type="checkbox"/> Codefolding in Editor | <input type="checkbox"/> CodeMirror mode extensions |
| <input type="checkbox"/> Collapsible Headings | <input type="checkbox"/> Comment/Uncomment Hotkey | <input checked="" type="checkbox"/> contrib_nbextensions_help_item | <input checked="" type="checkbox"/> datestamper |
| <input type="checkbox"/> Equation Auto Numbering | <input checked="" type="checkbox"/> ExecuteTime | <input type="checkbox"/> Execution Dependencies | <input type="checkbox"/> Exercise |
| <input type="checkbox"/> Exercise2 | <input type="checkbox"/> Export Embedded HTML | <input type="checkbox"/> Freeze | <input checked="" type="checkbox"/> Gist-it |
| <input type="checkbox"/> Help panel | <input type="checkbox"/> Hide Header | <input type="checkbox"/> Hide input | <input type="checkbox"/> Hide input all |
| <input type="checkbox"/> Highlight selected word | <input type="checkbox"/> highlighter | <input type="checkbox"/> Hinterland | <input type="checkbox"/> Initialization cells |
| <input type="checkbox"/> isort formatter | <input checked="" type="checkbox"/> jupyter-js-widgets/extension | <input type="checkbox"/> Keyboard shortcut editor | <input type="checkbox"/> Launch QTConsole |
| <input type="checkbox"/> Limit Output | <input type="checkbox"/> Live Markdown Preview | <input type="checkbox"/> Load TeX macros | <input type="checkbox"/> Move selected cells |
| <input type="checkbox"/> Navigation-Hotkeys | <input checked="" type="checkbox"/> Nbextensions dashboard tab | <input checked="" type="checkbox"/> Nbextensions edit menu item | <input type="checkbox"/> nbTranslate |
| <input type="checkbox"/> Notify | <input type="checkbox"/> Printview | <input type="checkbox"/> Python Markdown | <input type="checkbox"/> Rubberband |
| <input type="checkbox"/> Ruler | <input type="checkbox"/> Ruler in Editor | <input type="checkbox"/> Runtools | <input type="checkbox"/> Scratchpad |
| <input type="checkbox"/> ScrollDown | <input checked="" type="checkbox"/> Select CodeMirror Keymap | <input type="checkbox"/> SKILL Syntax | <input type="checkbox"/> Skip-Traceback |
| <input type="checkbox"/> Snippets | <input checked="" type="checkbox"/> Snippets Menu | <input checked="" type="checkbox"/> spellchecker | <input type="checkbox"/> Split Cells Notebook |
| <input type="checkbox"/> Table of Contents (2) | <input type="checkbox"/> table_beautifier | <input checked="" type="checkbox"/> Toggle all line numbers | <input type="checkbox"/> Tree Filter |
| <input checked="" type="checkbox"/> Variable Inspector | <input checked="" type="checkbox"/> zenmode | | |

ExecuteTime

Display when each cell has been executed and how long it took

section: notebook

require path: execute_time/ExecuteTime

compatibility: 4.x, 5.x

Enable

Disable

Last executed 2016-02-17 13:39:49 in 5ms

In [2]: randword(0.5)

Last executed 2016-02-17 13:39:49 in 519ms

Out[2]: 'zsygaoxnhe'



In [1]:

```
1 a = 5
2 b = True
3 c = "Karol"
4 d = []
5 e = {}
6 f = ()
7 g = 6.5
8
```

In []:

1

Variable Inspector

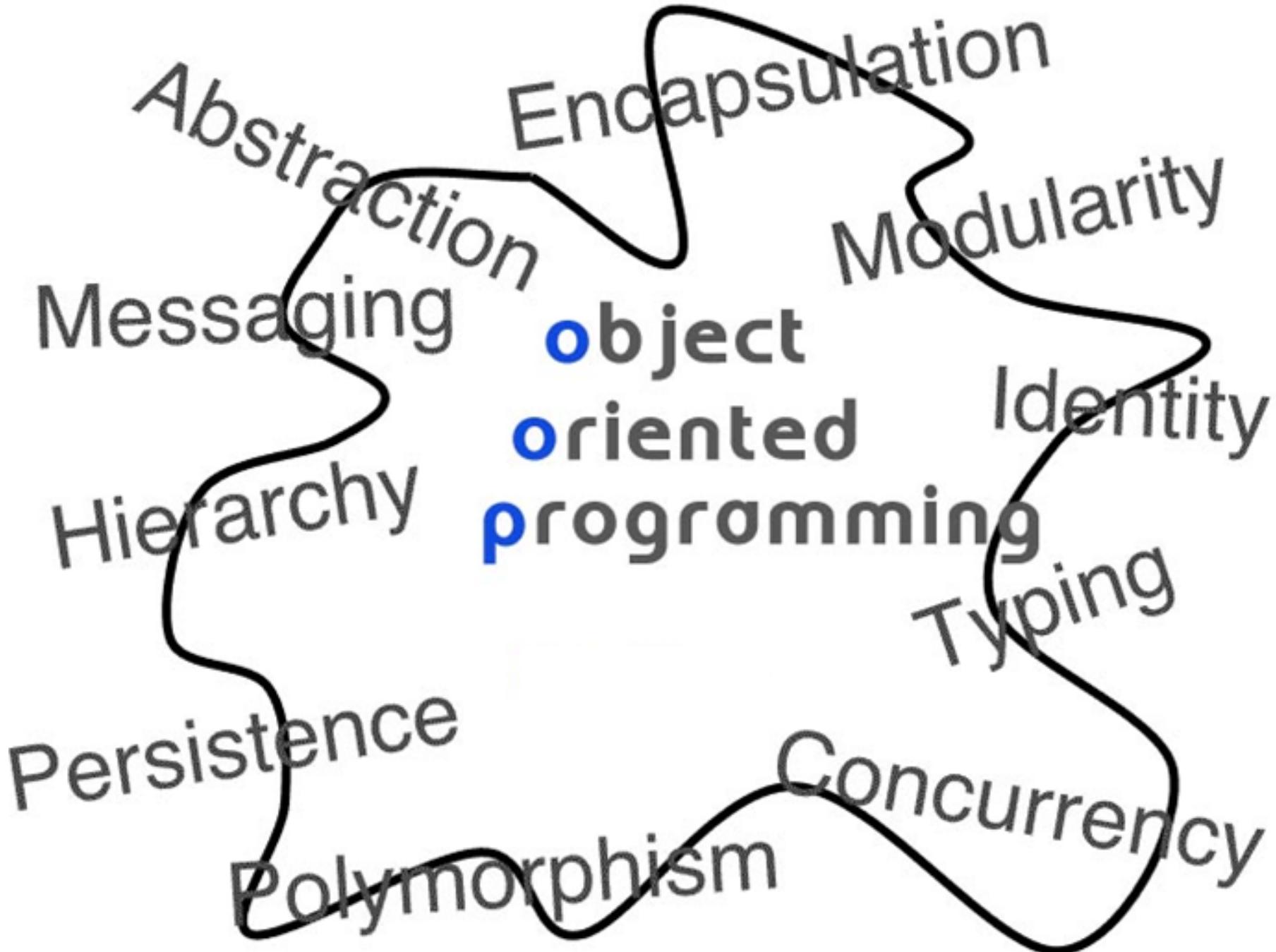
↻[-][x]

X	Name	Type	Size	Value
x	a	int	28	5
x	g	float	24	6.5
x	f	tuple	40	()
x	d	list	56	[]
x	e	dict	64	{}
x	b	bool	28	True
x	c	str	54	Karol

OOP?

OBJEKTOVÉ ORIENTOVANÉ
PROGRAMOVANIE





Prečo používame OOP?

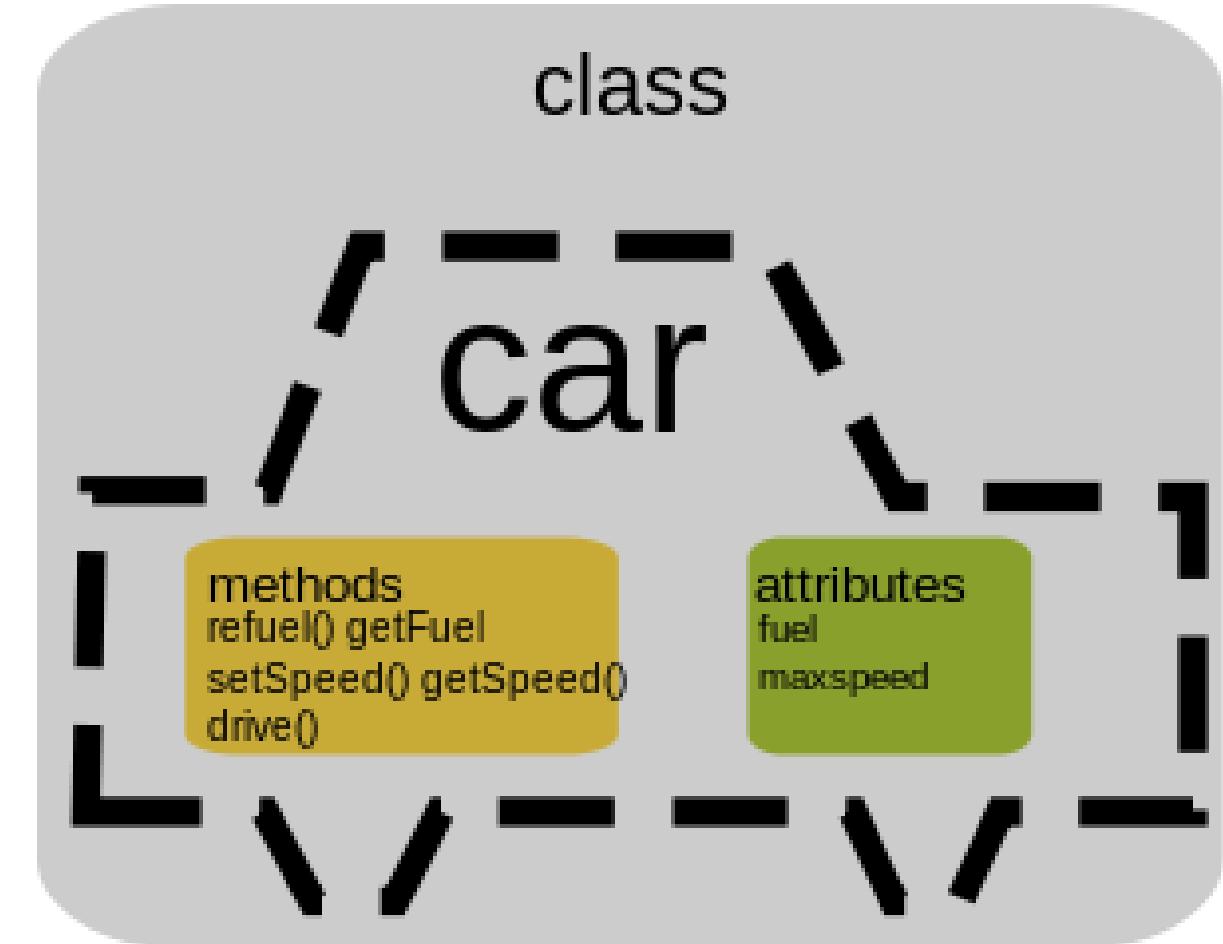
Nová paradigma

Časti programu sú zoskupované do väčších celkov – objektov

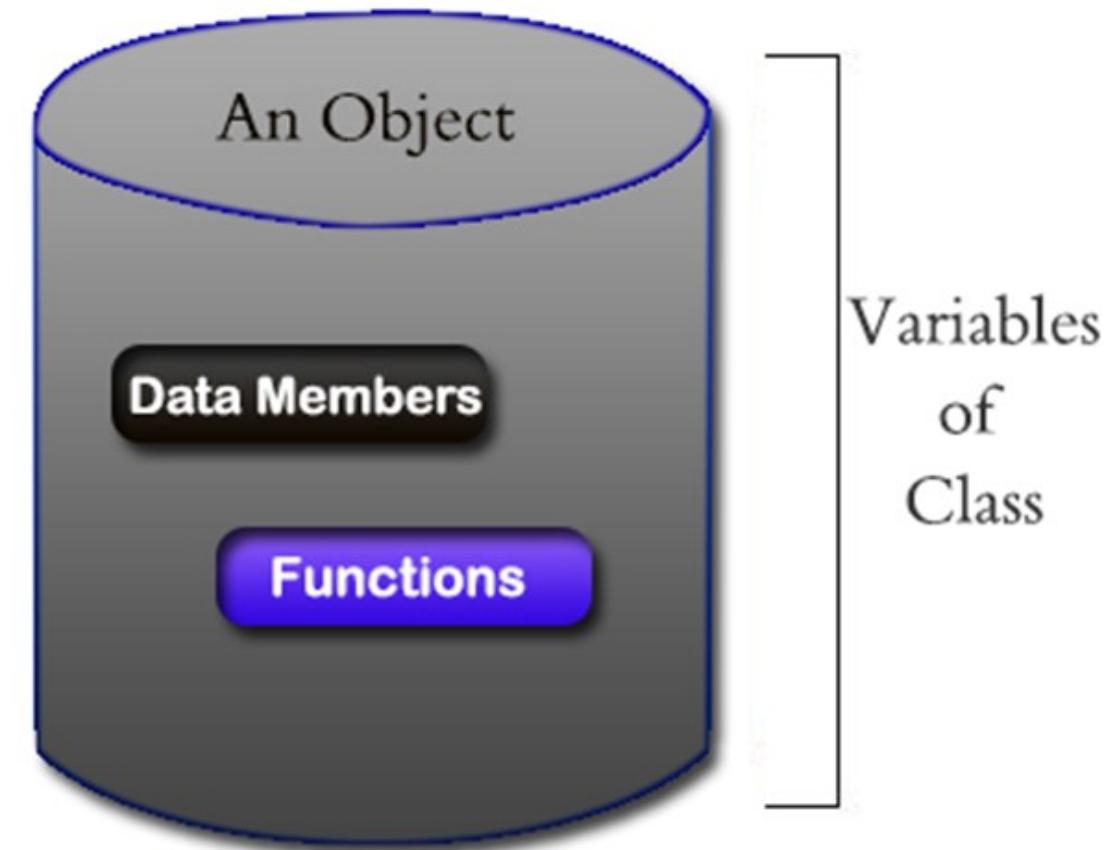
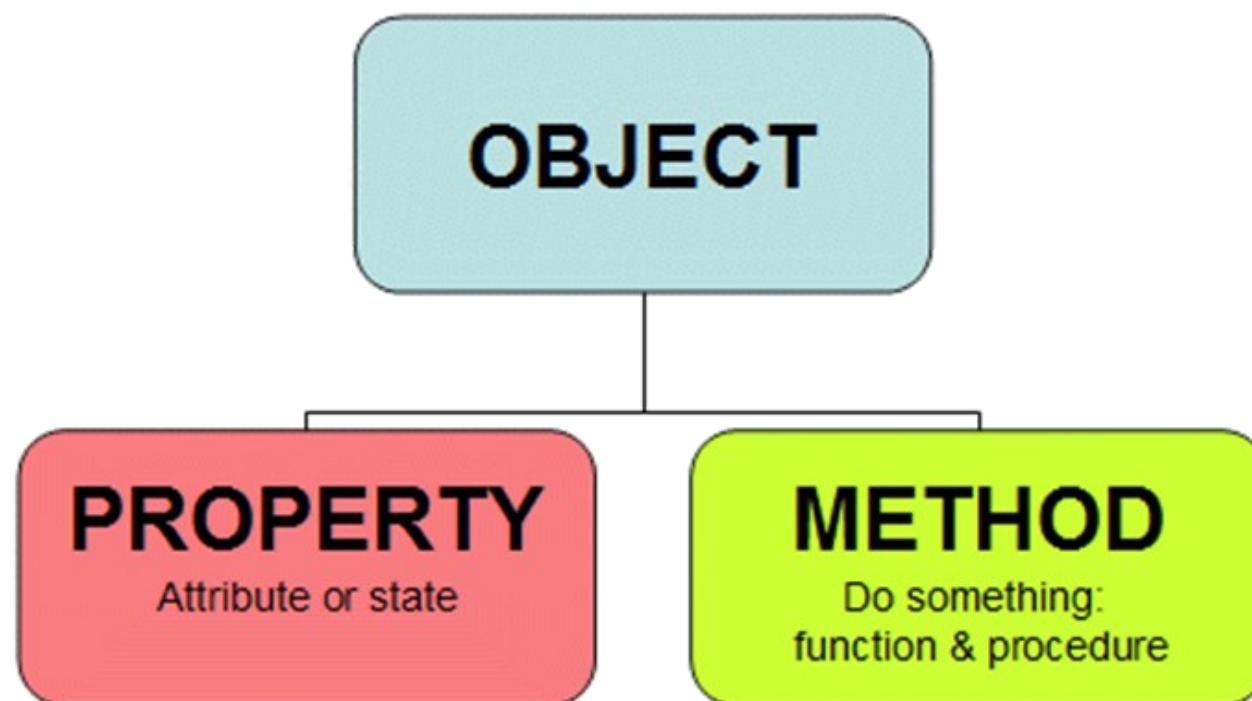
Objekty sa vytvárajú na základe preddefinovaných kritérií – tried

Prehľadnejší kód

Podobnosť k reálnemu svetu



Základy OOP



OOP a objekt

1. Čo je to za objekt?
2. Aké má vlastnosti?
3. Aké bude mať asi správanie?
4. Ako podľa tohto objektu vytvoríme "ideálnu" triedu a naopak?
5. Má/bude mať potomkov?
6. Koľko takýchto objektov potrebujeme/budeme potrebovať?



Triedy a objekty

Trieda



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

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

Objekty (inštancie triedy Pes)

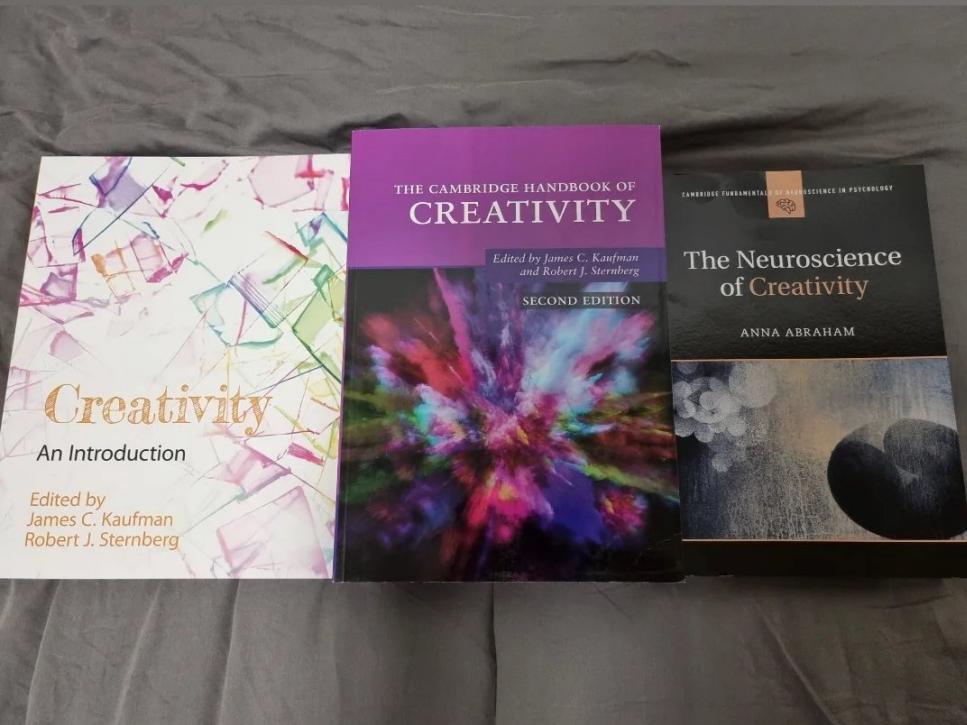


```
pes1 = Pes()  
pes1.meno = "Luigi"
```



```
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 allocated memory when it is created.
Object is created through new keyword. Employee ob = new Employee();	Class is declared using class keyword. <code>class Employee{}</code>
There are different ways to create object in java:- New keyword, newInstance() method, clone() method, And deserialization.	There is only one way to define a class, i.e.. by using class keyword.



OOP a objekty

1. Čo majú tieto objekty spoločné?
2. Čo majú tieto objekty odlišné?
3. Je možné niektoré vlastnosti generalizovať(zovšeobecniť) a používať opakovane?
4. Koľko takýchto objektov potrebujeme/budeme potrebovať?

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']
```

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



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?



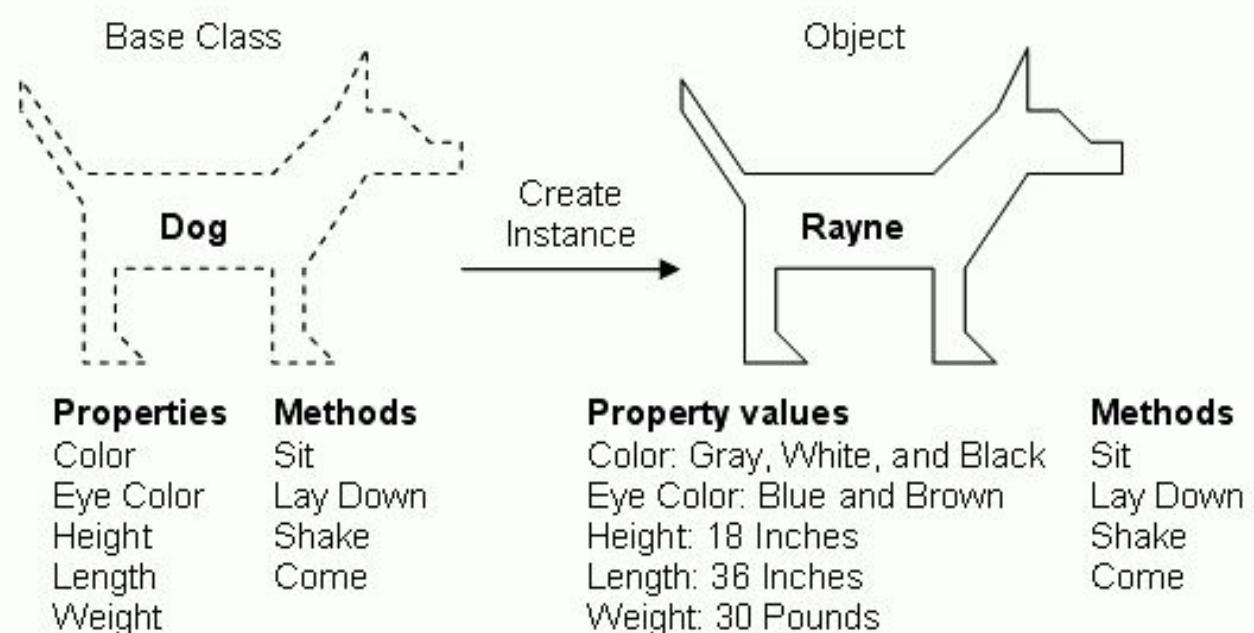
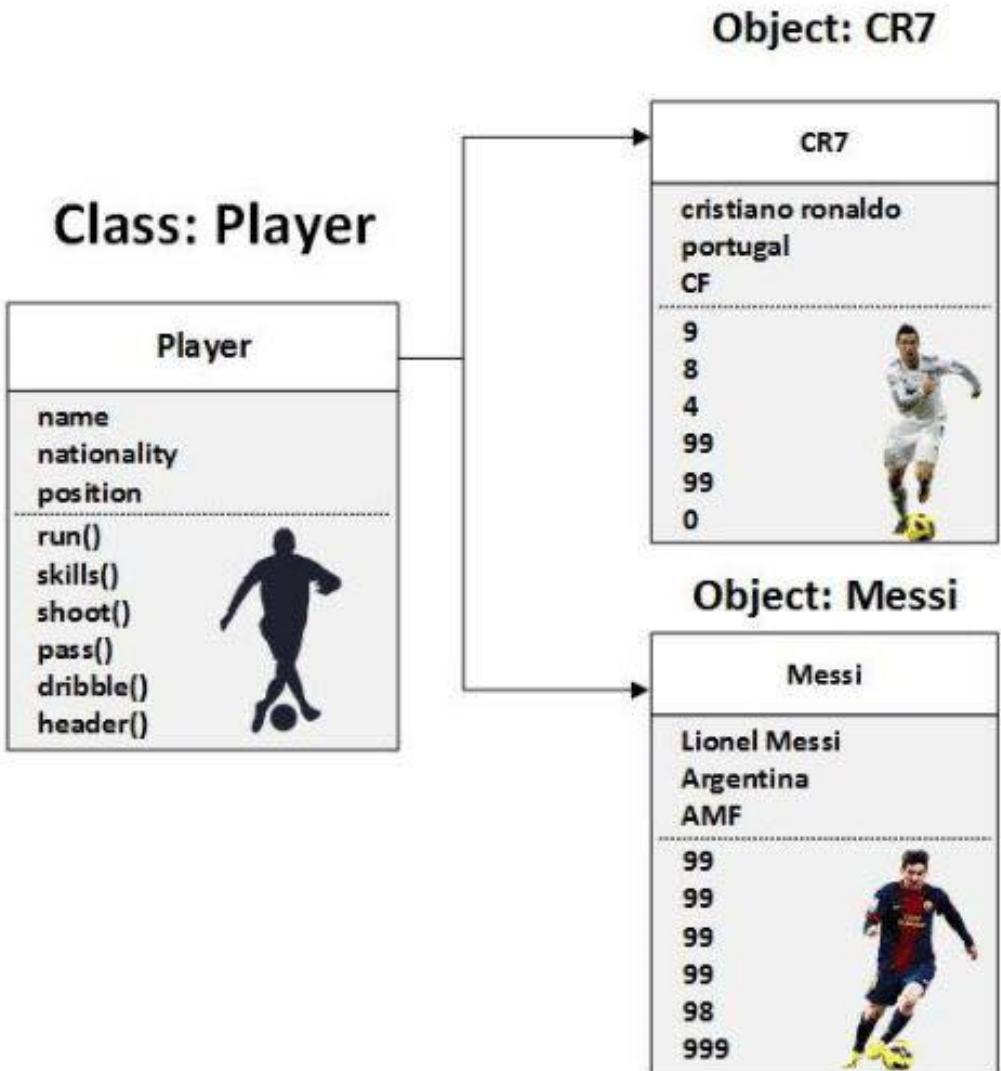
Reply



58



Triedy a objekty



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

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

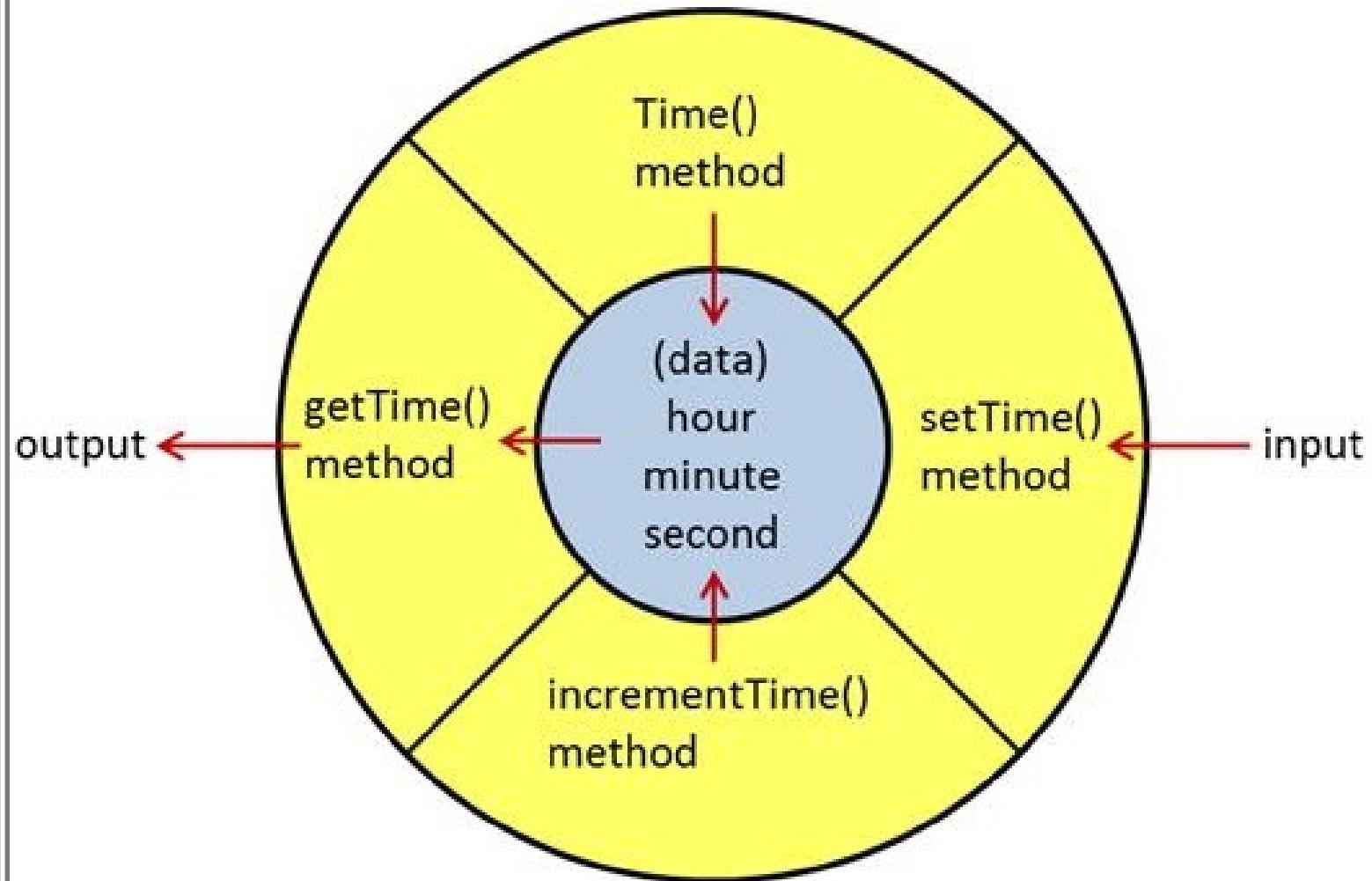
1. # Vytvorenie instance triedy
2. Objekt = Nazov_Triedy(...)
3. # Vytvorenie noveho atributu/zmena hodnoty
4. Objekt.atribut = hodnota
5. # Zavolanie metoda
6. Objekt.metoda(parametre)
7. # Zrušenie/vymazanie danej triedy
8. Del objekt



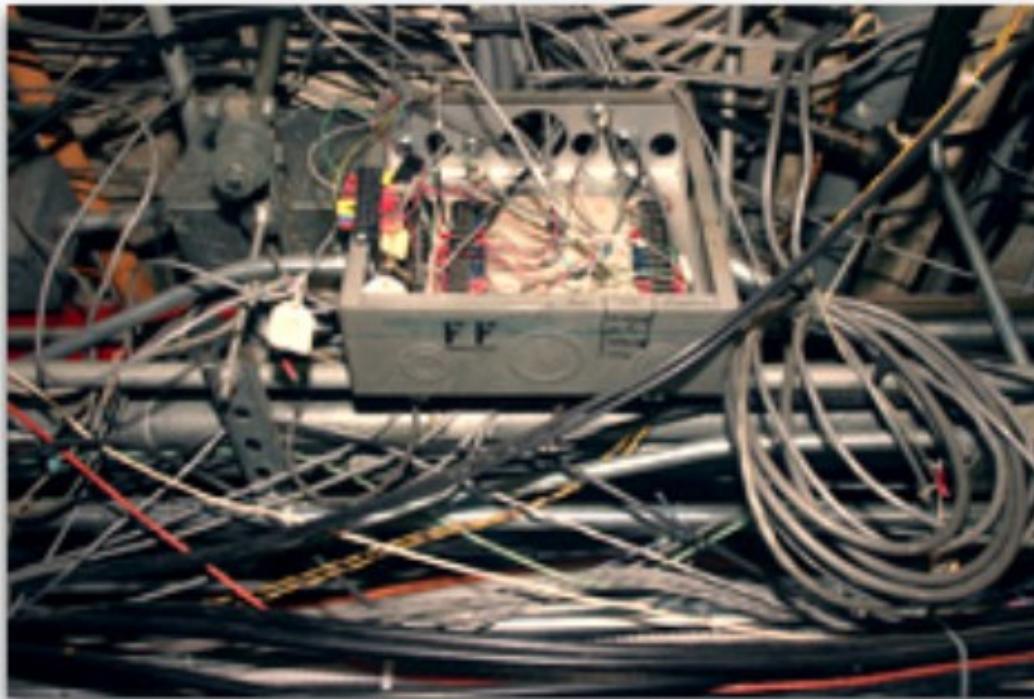
Vyber čo je trieda a čo je atribút?

Výška	Tvar	Celková zazmluvnená suma
Poskytovateľ pomoci	Popis projektu	Miesto realizácie
IČO	Merateľné ukazovatele	Dátum platnosti zmluvy
Operačný program	DPH	Formulár

Zapúzdrenie (enkapuslacia)



Zapúzdrenie (enkapuslacia)



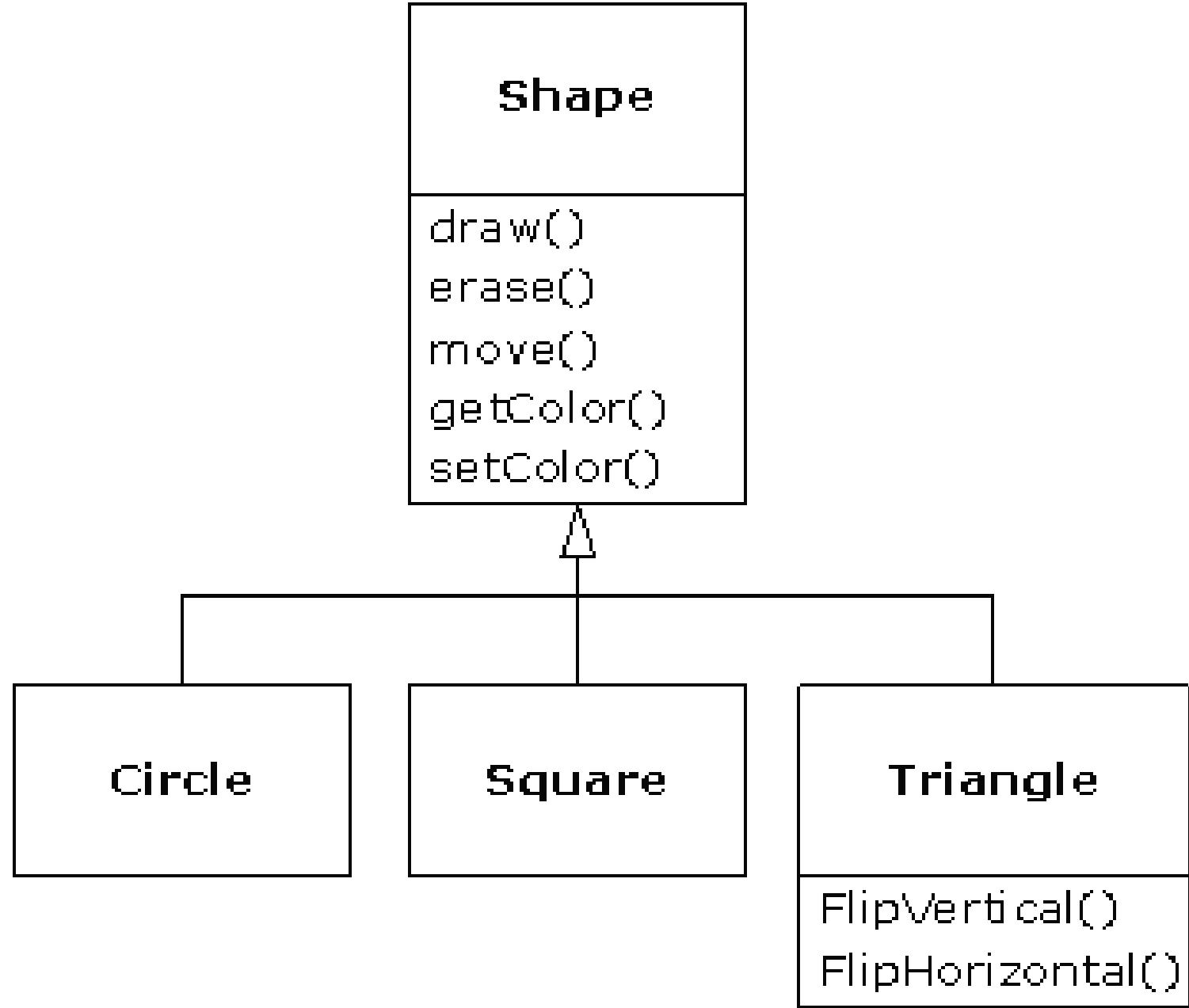
Jedno z týchto riešení je o niečo bezpečnejšie



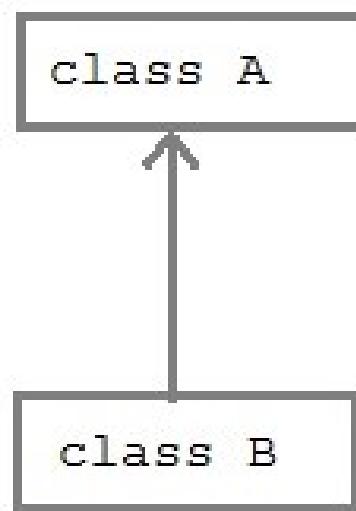
Dedičnosť

1. Trieda má/bude mať potomkov?
2. Potrebujeme odvodeniny?
3. Chcem rovnaké/podobné/odlišné správanie?
4. Akí sú predkovia? Aké majú vlastnosti a správanie?

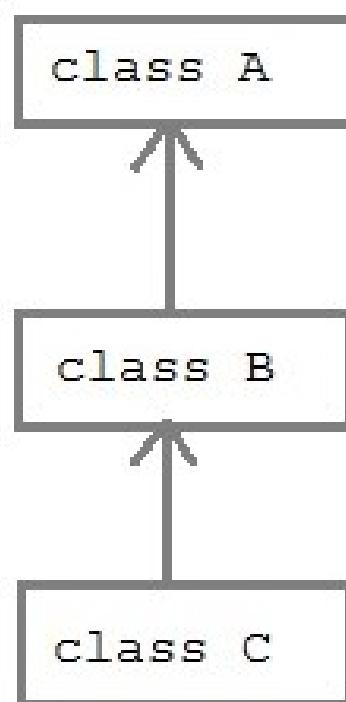
Dedičnost (Inheritance)



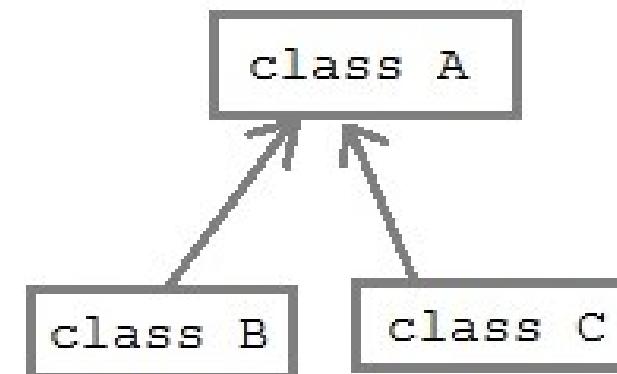
Typy dedičností



**Simple
Inheritance**



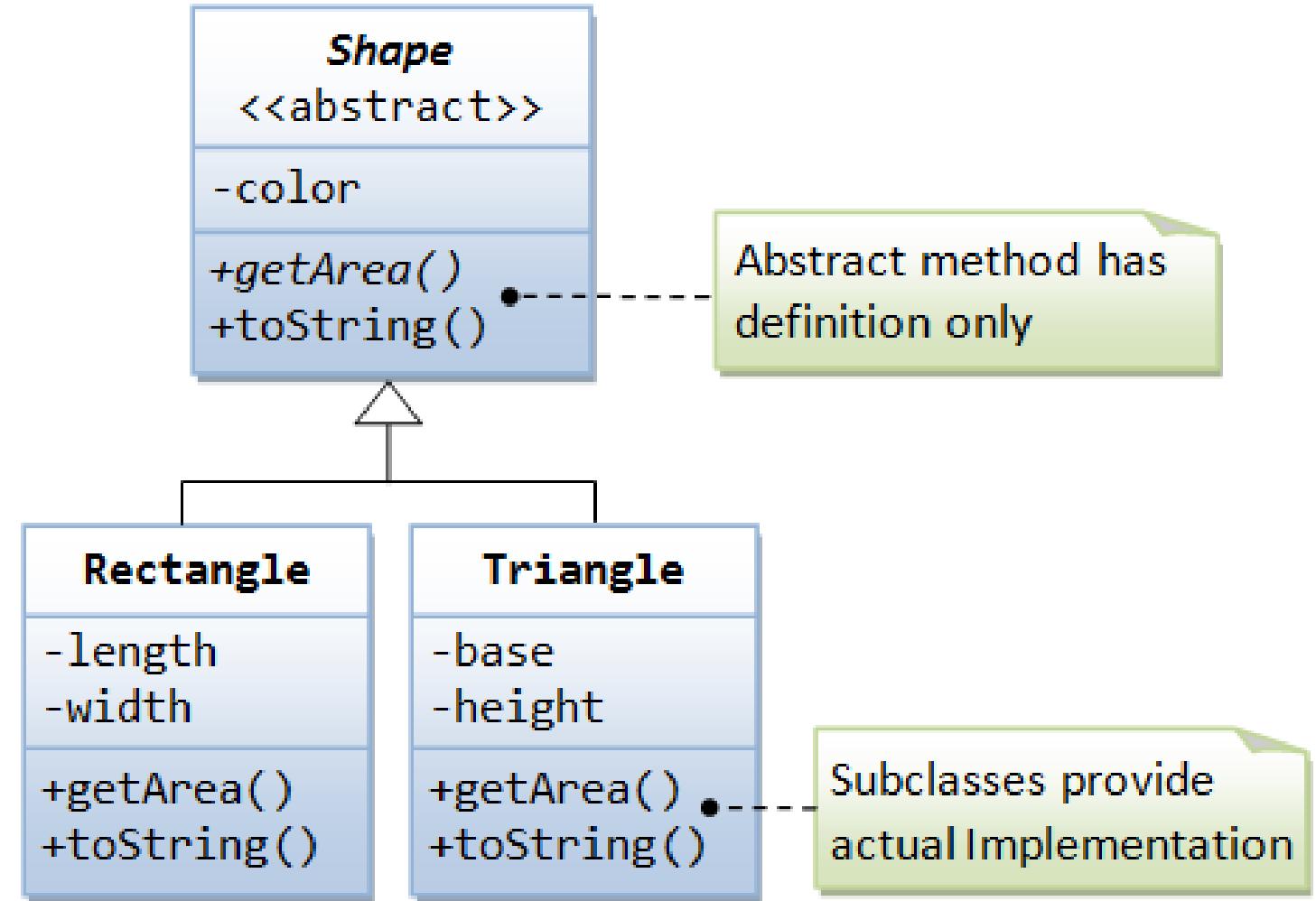
**Multilevel
inheritance**



**Heirarchical
inheritance**

Abstrakcia (Abstraktné Triedy)

- Pod abstrakciou sa vo všeobecnosti chápe zameranie sa na **kľúčové vlastnosti** nejakého prvku reálneho sveta (alebo aj nereálneho)
- V OOP to zúžitkujeme to hlavne pri abstraktných triedach, čo vám ukážem na obľúbenom príklade s útvarmi, ktoré budeme vykresľovať



Abstraktné Triedy

Use the [abc](#) module to create abstract classes. Use the [abstractmethod](#) decorator to declare a method abstract, and declare a class abstract using one of three ways, depending upon your Python version.

In Python 3.4 and above, you can inherit from [ABC](#). In earlier versions of Python, you need to specify your class's metaclass as [ABCMeta](#). Specifying the metaclass has different syntax in Python 3 and Python 2. The three possibilities are shown below:

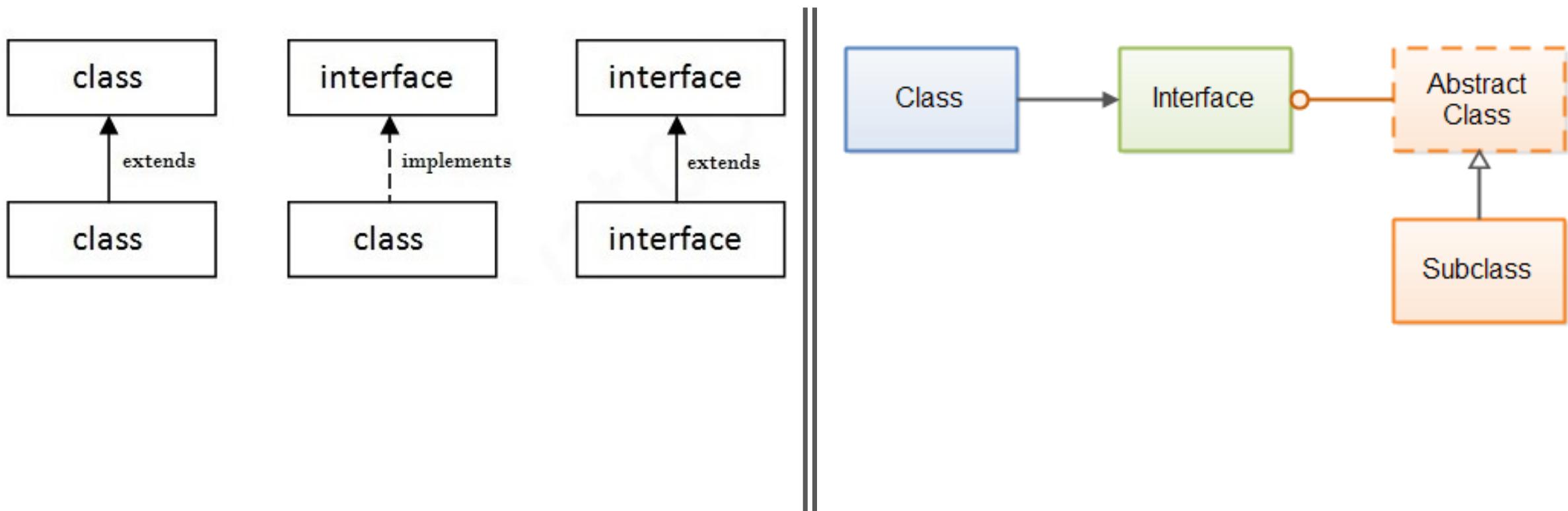
```
# Python 3.4+
from abc import ABC, abstractmethod
class Abstract(ABC):
    @abstractmethod
    def foo(self):
        pass
```

```
# Python 3.0+
from abc import ABCMeta, abstractmethod
class Abstract(metaclass=ABCMeta):
    @abstractmethod
    def foo(self):
        pass
```

```
# Python 2
from abc import ABCMeta, abstractmethod
class Abstract:
    __metaclass__ = ABCMeta

    @abstractmethod
    def foo(self):
        pass
```

Rozhrania a Dedičnosť



Polymorfizmus

- Umožňuje zmeniť nejakú metódu rodičovskej triedy u potomka
- Zmenenú metódu jednoducho u potomka znova definujeme podľa potrieb potomka
- Ak Python vykonáva nejakú metódu napr. `speak()`, tak sa toto vykonávanie prispôsobí (adaptuje) tej inštancii, ktorá túto metódu zavolala
 1. Pes → Woof
 2. Kačka → Quack
 3. Mačka → Meow



Polymorfizmus (Mnohotvárnost')

DID YOU KNOW

WHAT IS POLYMORPHISM ?

"A boy starts **LOVE** with the word **FRIENDSHIP** but girl ends **LOVE** with the same word **FRIENDSHIP**. Word is the same but attitude is different.This beautiful concept of **OOP** is nothing but **POLYMORPHISM...**"

-Broken Heart Programmer

Polymorfizmus

```
1. Class RodicovskaTrieda:  
2. atributy...  
3. def metoda1(self):  
4. ...  
5. def metoda2(self):  
6. ...
```

```
1. Class  
DetskaTrieda(RodicovskaTrieda):  
2. atributy...  
3. # Metoda sa predefinuje podľa  
potrieb potomka->polymorfizmus  
4. def metoda1(self):  
5. ...  
6. # Nové metódy a atribúty  
7. matributy  
8. metody
```

Princíp identity a funkcia id()

Úplna identita:

- Objekt.atribut/metoda
- Kto/Čo . Čo chcem dostať/vykonat'
- Karol.zaplat()
- Karol.vek

Anonymná identita:

- Neuvediem meno objektu . Čo chcem dostať/vykonat'
- 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**



Používame 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

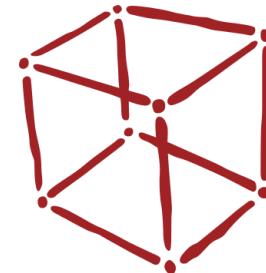
```
meno = 'laco'  
'a' in meno  
'x' in meno  
  
rodina = ['mama', 'otec', 'brat']  
'sestra' in rodina  
'mama' in rodina
```

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

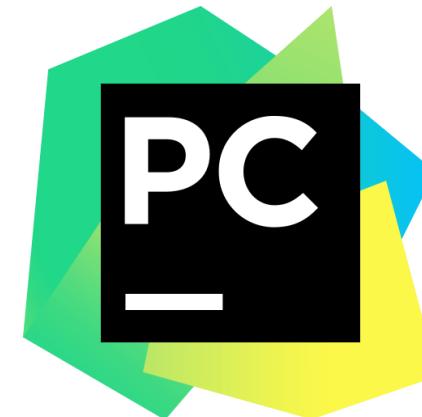
```
for jedlo in obchod:  
    print "chcem " + jedlo
```

is je $\text{id}(a) == \text{id}(b)$

Aké IDE mám použiť?



NetBeans



WxPython

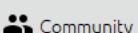
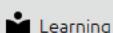


Visual Studio



Chceme úplne všetko!



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

Applications on

base (root)

Channels

Refresh



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](#)

Microsoft Store

python

Filtre

"python"

Všetky oddelenia Aplikácie Hry

Python 3.9

Aplikácie • Developer tools

★★★★★ 4

Nainštalované

Python 3.8

Aplikácie • Developer tools

Bezplatné

Python 3.10

Aplikácie • Developer tools

Bezplatné

Python 3.7

Aplikácie • Developer tools

★★★★★ 1

Bezplatné

Learn Django and Python by GoLearningBus

Aplikácie • Books & reference

Bezplatné

WiBit.Net :: Programming in Python

Aplikácie • Education

0,99 €

计算机二级 Python 考试试题库

Aplicacie • Education

Bezplatné

Data Science with Python

Aplikácie • Education

Python Programs

Aplikácie • Education

Learn Python

Aplikácie • Books & reference

Introduction to Python Programming by...

Aplikácie • Books & reference

Python Playground

Aplikácie • Developer tools

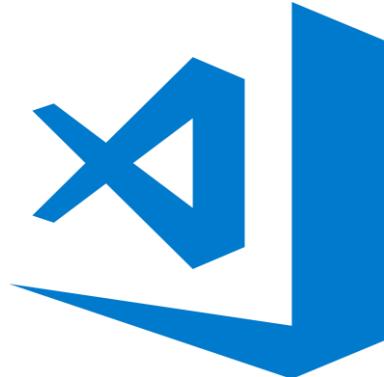
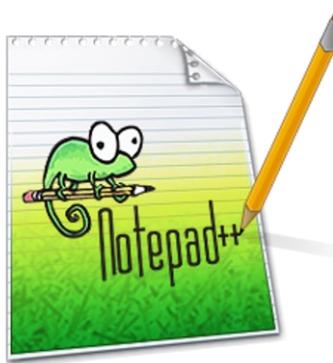
Python Programming Manual For Absolute...

Aplikácie • Utilities & tools

Python User Tutorial

Aplikácie • Utilities & tools

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.

Search bar

- > 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

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

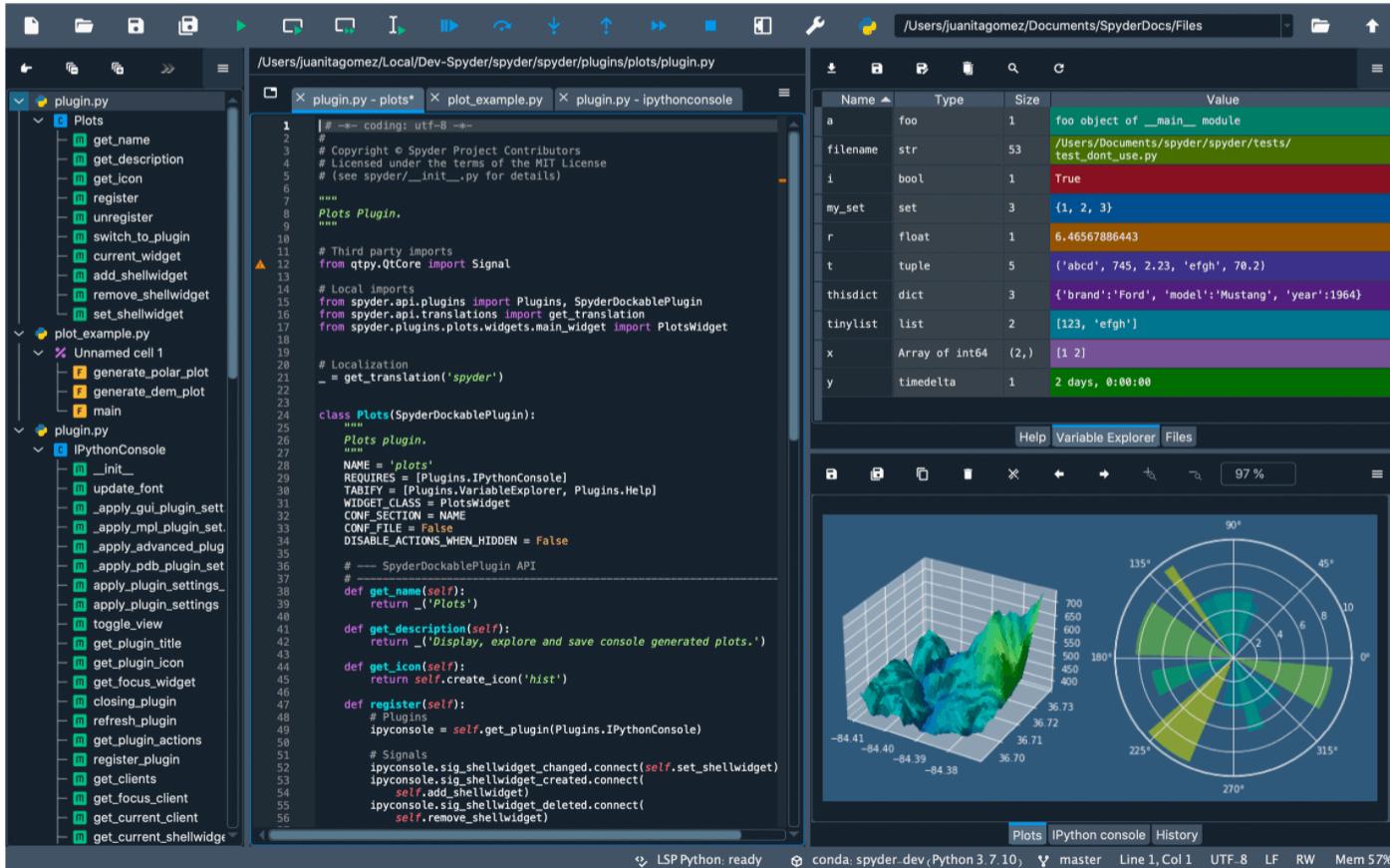
> React

...

No live templates are selected



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. It includes imports for `QtCore`, `Signal`, and various Spyder API modules. The class `Plots` inherits from `SpyderDockablePlugin` and implements methods like `get_name`, `get_description`, `get_icon`, `register`, etc.
- 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 shown in the bottom right:
 - A 3D surface plot of a function over a grid.
 - A polar plot with radial and angular axes.
- Status Bar:** Shows "LSP Python: ready", "conda: spyder.dev:Python 3.7.10", and memory usage "Mem 57%".

VERSION

Spyder 5

Search ...



WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

▶ WORKSHOPS

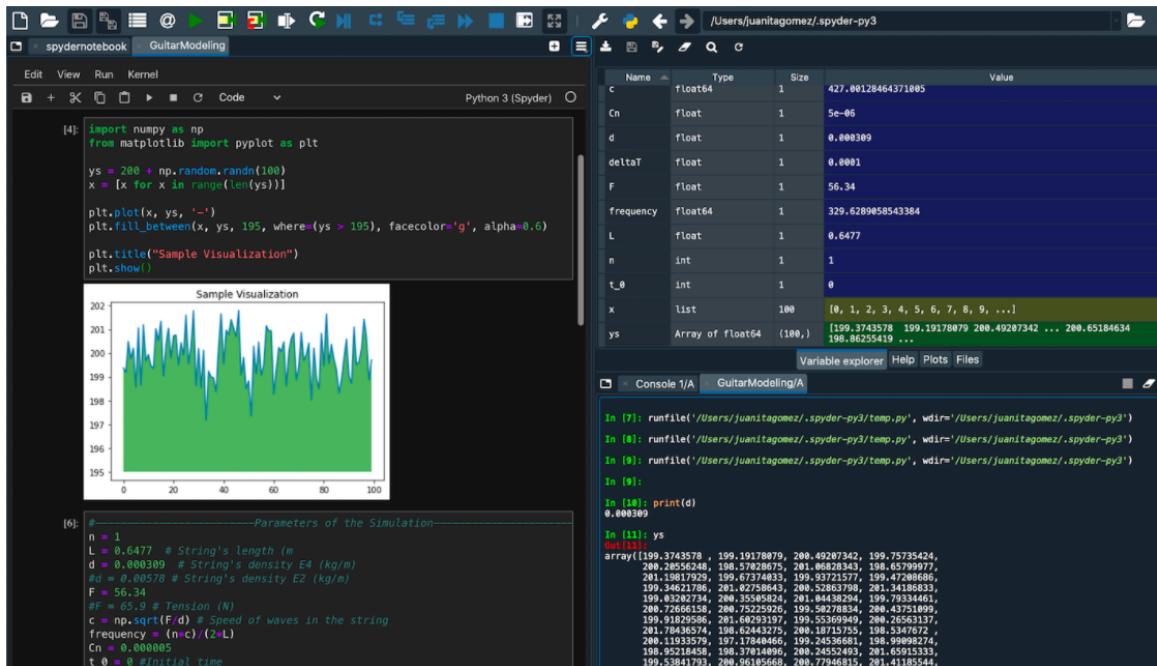
FAQ

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"]

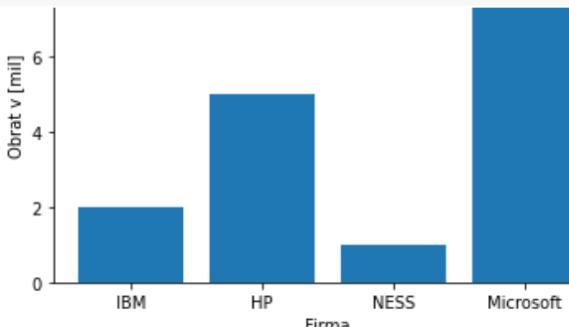
print(type(y))
print(y)

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...

Value

Name Type Size

Edit View Run Kernel

Python 3 (Spyder)

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.
Je k dispozicii False
```

Editor Notebook

IPython console History

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

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()	replace()	utcoffset()
platform	Current platform	isalnum() *	split(sep)	isoformat()	dst()
stdin, stdout, stderr	File objects for I/O	isalpha() *	splines()	__str__()	tzname()
version_info	Python version info	isdigit() *	startswith(sub)	strftime(format)	
winver	Version number	islower() *	strip()		
sys.argv for \$ python foo.py bar -c qux --h		isspace() *	swapcase() *	Date Formatting (strftime and strptime)	
sys.argv[0]	foo.py	istitle() *	title() *	%a	Abbreviated weekday (Sun)
sys.argv[1]	bar	isupper() *	translate(table)	%A	Weekday (Sunday)
sys.argv[2]	-c	join()	upper() *	%b	Abbreviated month name (Jan)
sys.argv[3]	qux	ljust(width)	zfill(width)	%B	Month name (January)
sys.argv[4]	--h	lower()		%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*)
				%U	Week number ¹ (00 to 53)
				%W	Week number ² (0 to 6)
				%x	Date
				%X	Time
				%y	Year without century (00 to 99)
				%Y	Year (2008)
				%Z	Time zone (GMT)
				%%	A literal "%" character (%)
os Variables		List Methods		File Methods	
altsep	Alternative sep	append(item)	pop(position)	close()	readlines(size)
curdir	Current dir string	count(item)	remove(item)	flush()	seek(offset)
defpath	Default search path	extend(list)	reverse()	fileno()	tell()
devnull	Path of null device	index(item)	sort()	isatty()	truncate(size)
extsep	Extension separator	insert(position, item)		next()	write(string)
linesep	Line separator			read(size)	writelines(list)
name	Name of OS			readline(size)	
pardir	Parent dir string				
pathsep	Patch separator				
sep	Path separator				
Note		Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"			
Class Special Methods		Indexes and Slices (of a=[0,1,2,3,4,5])		1.	
__new__(cls)	__lt__(self, other)	len(a)	6	Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.	
__init__(self, args)	__le__(self, other)	a[0]	0	2.	0 is Sunday, 6 is Saturday.
__del__(self)	__gt__(self, other)	a[5]	5	3.	Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.
__repr__(self)	__ge__(self, other)	a[-1]	5	4.	This is not a mistake. Range takes account of leap and double-leap seconds.
__str__(self)	__eq__(self, other)	a[-2]	4		
__cmp__(self, other)	__ne__(self, other)	a[1:]	[1,2,3,4,5]		
__index__(self)	__nonzero__(self)	a[:5]	[0,1,2,3,4]		
__hash__(self)		a[:-2]	[0,1,2,3]		
__getattr__(self, name)		a[1:3]	[1,2]		
__getattribute__(self, name)		a[1:-1]	[1,2,3,4]		
__setattr__(self, name, attr)		b=a[:]	Shallow copy of a		
__delattr__(self, name)					
__call__(self, args, kwargs)					

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				
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 ¹	Y					Y	Y	Y			Y ²				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		
NetBeans*	Y	F	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	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					
SciTE	Y	F ¹		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 code editor with the following code:

```
import unittest
class Robot(object):
    def __init__(self, x, y):
        self.x = x
        self.y = y
class Test(unittest.TestCase):
    def test_robots(self):
        robot = Robot(x=0, y=0)
        robot.walk(x=1, y=1)
```

A cursor is positioned at the end of the word "walk". A context menu is open, listing options: "Create walk method at Robot", "Assign to field (self.walk)", and "Assign to local (walk)".

PyDev development

PyDev is open source and depends on **you**! You can help PyDev in many ways. You may be in the form of bug fixes, answers to questions, documentation, new features... Another option is financial support (PayPal):

\$7 \$20 \$45 \$100

Corporate sponsorship is also available for your company.

Search PyDev-related content

Google™ Custom Search

 Favorite at:
eclipse marketplace



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()
```

Isinstance a issubclass

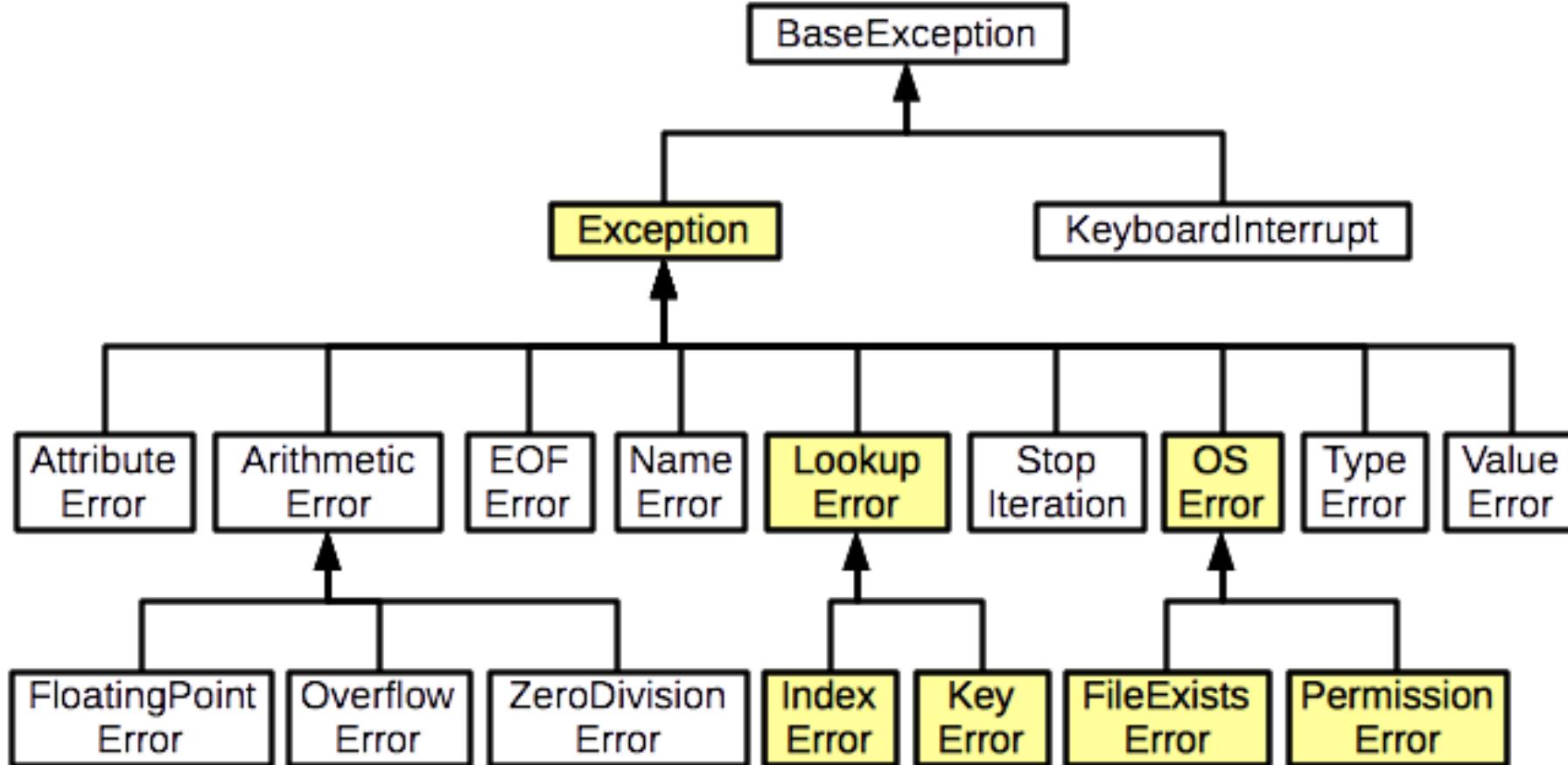
Boolean notácia:

- **is**
- **has**
- **can**

- Je Laco Objektom triedy str?
- Je trieda sama sebe podtriedou?

```
>>> isinstance("Laco", str)
True
>>> isinstance("Laco", int)
False
>>> isinstance(5, int)
True
>>> isinstance(5L, int)
False
>>> issubclass(int, object)
True
>>> issubclass(Exception, object)
True
>>> issubclass(ArithmetricError, Exception)
True
>>> issubclass(SyntaxError, Exception)
True
```

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

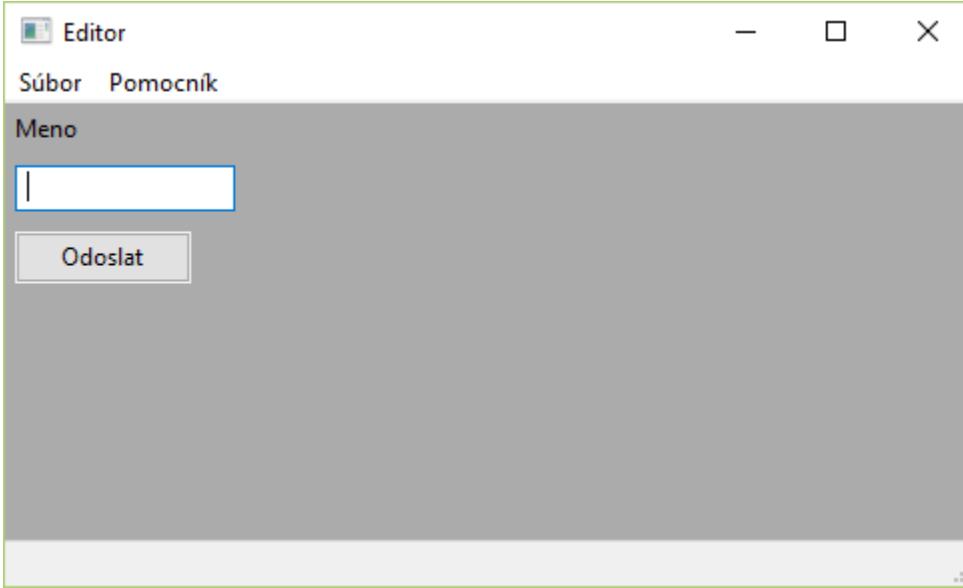
- **`__dict__`**: Dictionary containing the class's namespace.
- **`__doc__`**: Class documentation string or None, if undefined.
- **`__name__`**: Class name.
- **`__module__`**: Module name in which the class is defined. This attribute is "`__main__`" in interactive mode.
- **`__bases__`**: A possibly 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 links for Home, Trees, Index, and Help, along with the version "wxPython 2.8.9.2" and options for frames or 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>
bool	<code>Command(self, winid)</code>
bool	<code>Create(self, parent, id, pos, size, style, name)</code> Create the GUI part of the Window for 2-phase creation mode.
StatusBar	<code>CreateStatusBar(self, number, style, winid, name)</code>
wxToolBar	<code>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

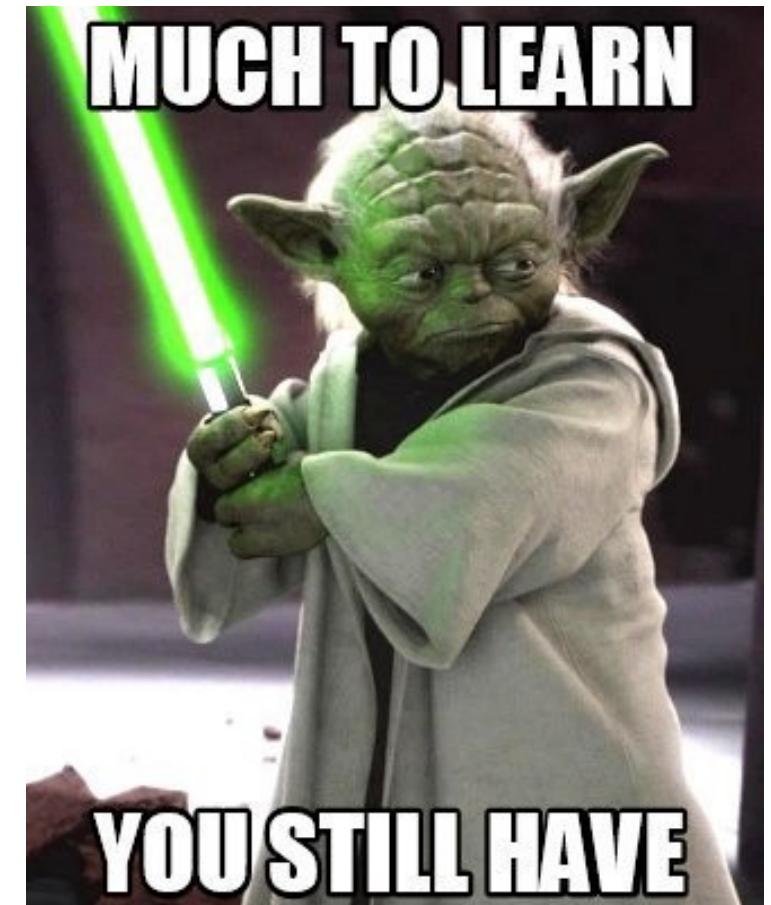
- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

YouTube tutoriály

Packt Publishing

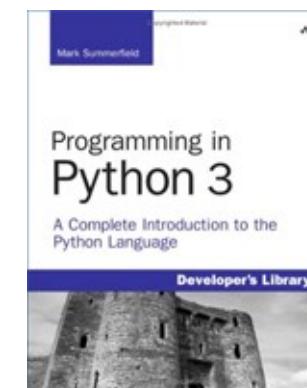
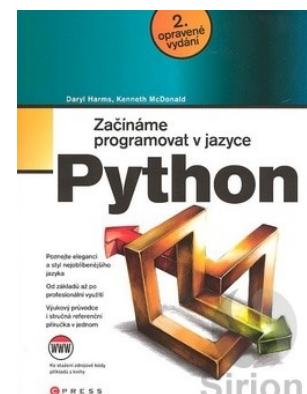
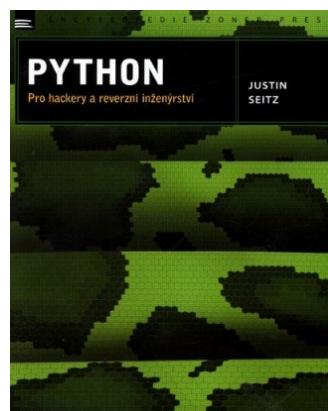
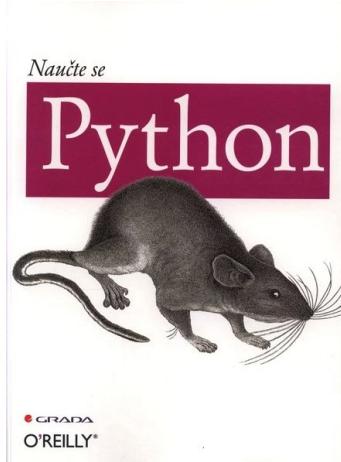
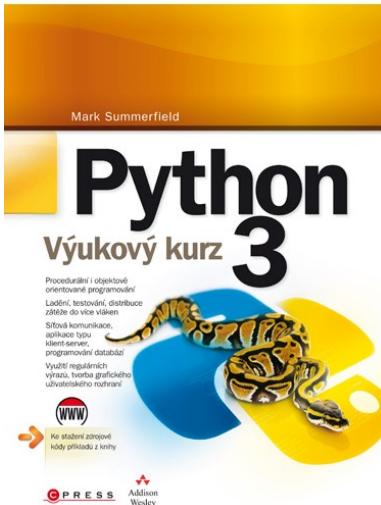
Čo ti odporúčam si pozrieť?

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>



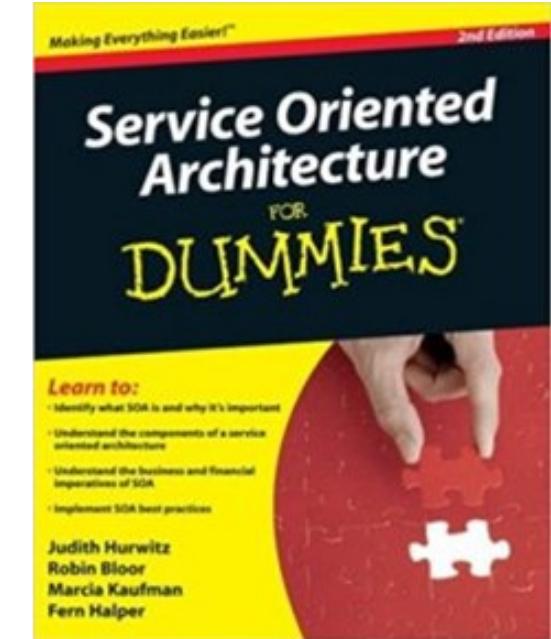
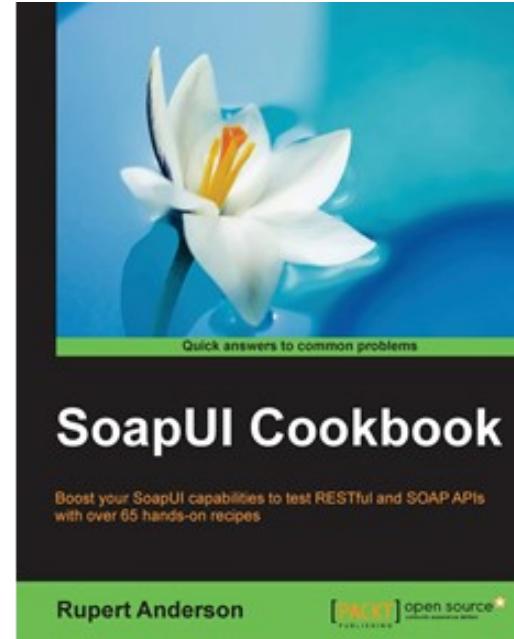
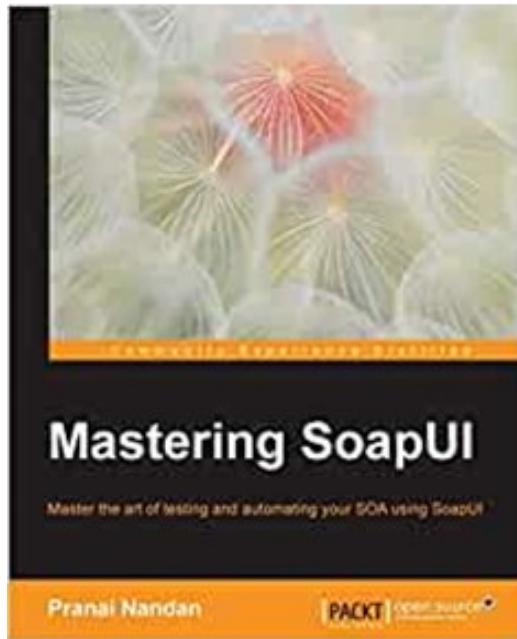
Šup do záložiek

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

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

[Popular](#) [Name](#) [New](#)

Efektívne používanie klávesnice

Špeciálne znaky, kde ich nájst' na klávesnici

The diagram shows a standard QWERTY keyboard layout with a focus on the top row and function keys F1-F12. Special characters are highlighted in various colors (e.g., red, green, blue) across these keys.

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

Retázce

! € # & # & & € !

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

...

NOVÁ AKTIVITA



Roland Mondek

10 h

POZVAŤ 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



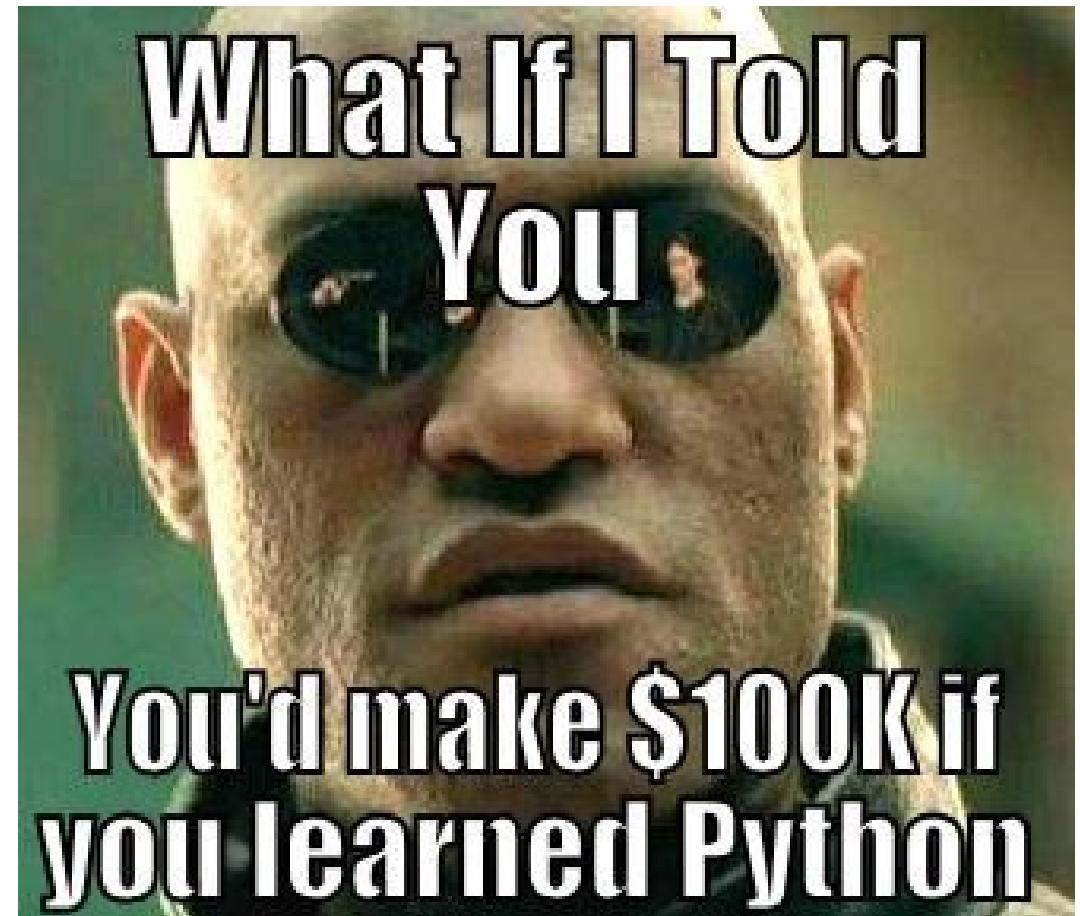
Vytvoriť novú skupinu

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



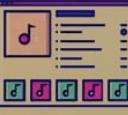
Quote Gener. Number guessing Dice Simulation YT downloader

Mid



Password Manag. Mario Party Web Crawler Email Autom.

Pro



Music Player Face Detection Twitter Clone Twitter Bot

Ako skončíme?

1. Pridaj si ma na LinkedIn

- 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 dashboard. At the top, there is a navigation bar with the LinkedIn logo, the word "SlideShare", a search bar, and a profile picture. Below the navigation bar, there are three main tabs: "Home", "Explore", and "Analytics". Under the "Home" tab, there are three sections: "My Uploads", "My Comments", and "Analytics". The "My Uploads" section is currently selected. It displays two presentations:

- Automatizácia a optimalizácia Firemné procesy**
by Ing. Mgr. Miroslav Reiter, DSc., N
mir.reiter@reiter.sk
1 month ago, 11 slides
Views: 113 | Likes: 0 | Comments: 0 | Downloads: 0
- Najväčšie faily na sociálnych sietach a Google reklame**
by Ing. Mgr. Miroslav Reiter, DSc., N
mir.reiter@reiter.sk
6 months ago, 63 slides
Views: 19 | Likes: 0 | Comments: 0 | 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

* Viac informácií na www.iamremarkable.sk



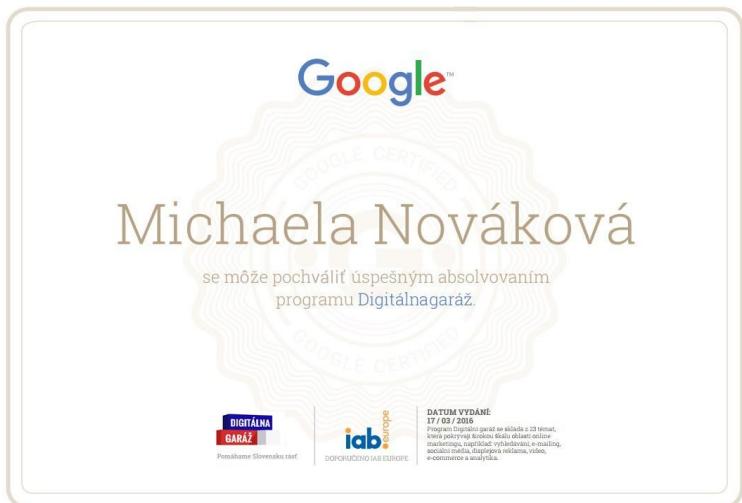
Digitálna garáž Online platforma na výuku digitálneho marketingu a mäkkých zručností

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

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

ZÁLOŽKA ZDIELAŤ

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€

Prístup 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