

```

py = "Kurz Phyton 00P Princípy"
a = py
# Cele
b = 2
# Realne
c = 5.58
d = True
e = None
# Complex
f = 5 + 2j
# Bajty
g = b"Ahoj"
h = memoryview(bytes(4))
import datetime
i = datetime.datetime.now()
print(f"Dnes je: {i}")

# Zoznam (list) []
j = ["Patricia", "Miroslav", "Maria", "Emilia"]

# tice (tuple) ()
k = (5, 8, 9, 6)

# Slovník (dict) {kluc:hodnota} {key:value}
l = {"programator": "Jaro", "analytik": "Monika"}

# Mnozina (set) {hodnoty/kluce}
l = {"programator", "analytik"}

# Zoznam objektov /Premennych
%who

# Zoznam objektov /Prehľad
%whos

# Zoznam oddeleny ciarkou
%who_ls

```

Dnes je: 2023-03-22 13:03:48.788746

DataloreMetric	SQLError	a	b	c	d	datetime
g	h	i	j	json	k	l
py					logging	os

Variable	Type	Data/Info
----------	------	-----------

DataloreMetric	type	<class '__main__.DataloreMetric'>
SQLError	type	<class '__main__.SQLError'>
a	str	Kurz Phyton 00P Princípy
b	int	2
c	float	5.58
d	bool	True
datetime	module	<module 'datetime' from '<...>b/python3.8/datetime.py'>

e	NoneType	None
f	complex	(5+2j)
g	bytes	b'Ahoj'
h	memoryview	<memory at 0x7fbe0b5f5940>
i	datetime	2023-03-22 13:03:48.788746
j	list	n=4

```
[ 'DataloreMetric',
  'SQLError',
  'a',
  'b',
  'c',
  'd',
  'datetime',
  'e',
  'f',
  'g',
  'h',
  'i',
  'j',
  'json',
  'k',
  'l',
  'logging',
  'os',
  'py']
```

2. Objekty Objekt = Podstatné meno Objekty sú súborom dát a metód, ktoré fungujú na základe údajov, ktoré uchovávajú. Dáta si môžete predstaviť ako premenné a metódy vytvárajú funkcionálnosť.

Príklady Objekty =

- strom lipa,
- zalohova faktura,
- kniha coelho,
- film kocur v cizmach,
- kurz python oop

Z čoho je objekt zložený/čo obsahuje:

1. Stav/Atributy/Vlastnosti/Datové Zložky/Data: ...
2. Chovanie/Správanie - Def/Funkcie/Metódy - Slovesa: ...

1.Příklad objekt = Zapisník Journal

Atributy/Datové zložky (podst.m., prídav.m, číslovky, kombinácia)

farba = čierna sirka, vyska = 20, 30 pocet_listov = 80

Chovanie/Spravanie/Metody (def) - (slovesa)

2.Příklad

Vlastnosti/Atributy

vek vyska hmotnost druh

Chovanie/Spravanie

rastie () plodí () produkuje kyslík ()

```
"ahoj svet".capitalize()  
"ahoj svet".isnumeric()  
True.real
```

1

III. Triedy (Classes)

V OOP píšete triedy, ktoré reprezentujú veci a situácie v reálnom svete. Trieda slúži ako plán (blueprint) na vytváranie objektov.

Vytvárate objekty založené na triedach.

- Manual/navod ako vytvorit objekty
- Trieda (abstraktne) - blueprinty
- Objekt (konkrétne) - instance triedy

Kuchár Trieda (posstatne meno - abstrakcia)

1. Napíšem recept/vytvorím/napíšem triedu
2. Najdem si recept/použijeme existujúcu triedu, triedy/stiahnem
3. Modifikujem to čo najdem/upravím existujúce

```
# Trieda - návod tvorby objektov/recept/plan - abstraktne napr. pes  
# Objekt - konkrétny - dvojročný hrubosrstý jazvecík
```

```
# Trieda  
class Pes:  
    pass  
  
# Objekt / instancia triedy  
# názov/meno = dátový typ  
Luigi = Pes()  
Mario = Pes()  
Yoshi = Pes()  
Rexo = Pes()
```

```
class Pes:
    # 1. Vlastnosti (Property), premenne, Data/Datove Zlozky
    meno = ""
    rasa = str()
    je_lenivy = False
    vek = 0
    hmotnost = 0.2
    pohlavia = ["samec", "samica"]

    # 2. Metody (def), chovanie
    def lezi(self):
        print("Lezi a je mu dobre...")

    def spi(self):
        print("Spi...")

    def vypis_info_vsetky(self):
        print(f"Ako sa vola --> {self.meno}")
        print(f"Kolko ma rokov --> {self.vek}")
        print(f"Aka je to rasa --> {self.rasa}")

rexik = Pes()
print(f"rexik je lenivy: {rexik.je_lenivy}")
print(f"rexik meno: {rexik.meno}")

rexik.meno = "rexik domaci"
print(f"rexik meno: {rexik.meno}")
print(f"rexik rasa: {rexik.rasa}")

rexik.rasa = "nemecky ovciak"
print(f"rexik rasa: {rexik.rasa}")

lassie = Pes()
lassie.meno = "Lassie"
lassie.rasa = "Kolia"
lassie.vek = 3

lassie.lezi()
lassie.spi()
lassie.vypis_info_vsetky()

misa = Pes()
misa.meno = "Miska"
misa.rasa = "Jazvecik"
misa.vek = 2
misa.vypis_info_vsetky()
```

```
rexik je lenivy: False
rexik meno:
rexik meno: rexik domaci
rexik rasa:
rexik rasa: nemecky ovciak
```

```
Lezi a je mu dobre...  
Spi...  
Ako sa vola --> Lassie  
Kolko ma rokov --> 3  
Aka je to rasa --> Kolia  
Ako sa vola --> Miska  
Kolko ma rokov --> 2  
Aka je to rasa --> Jazvecik
```

IV. Objekty, id, str

```

class Kniha:
    """ DocString - Dokumentacny retazec
        Eshop - Phyton Programovacie Knihy
        @author Adam Sangala
        @version 1.0
        @example obj_kniha = Kniha()
    """

    # 1.Property/Vlastnosti
    nazov = str()
    cena = float()
    pocet_stran = ()
    ISBN = str()
    autor = str()
    zaner = list()
    # 2. Chovanie/Metody
    def vypis_info(self):
        print(f"Nazov --> {self.nazov}")
        print(f"Autor --> {self.autor}")
        print(f"Pocet stran --> {self.pocet_stran}")

    def __str__(self):
        return f"Nazov objektu: {self.nazov} a pocet stran je: {self.pocet_stran}"

kniha1 = Kniha()
kniha2 = Kniha()
kniha3 = Kniha()
#redekларacia
kniha2 = kniha3

print(f"kniha1: {id(kniha1)}")
print(f"kniha2: {id(kniha2)}")
print(f"kniha3: {id(kniha3)}")

meno = "Adam"
vek = 35
print(f"meno: {id(meno)}")
print(f"vek: {id(vek)}")

# == operator porovnanania - equality operator
# is identity operator - operator
print(f"Tieto objekty kniha1 a kniha2 su rovnako {kniha1 == kniha2}")
print(f"Tieto objekty kniha1 a kniha2 su rovnake: {id(kniha1) == id(kniha2)}")
print(f"Tieto objekty kniha1 a kniha2 su rovnake: {kniha1 is kniha2}")
print(f"Tieto objekty kniha1 a kniha2 su rovnake: {kniha2 is kniha3}")

kniha1.nazov = "Head First Python"
kniha1.vypis_info()

kniha2.nazov = "Head First Python 2. edition"
kniha2.vypis_info()

```

```
kniha3.nazov = "Head First Python 2. edition"
kniha3.vypis_info()
```

```
print(f"Co som zac: {kniha1}")
print(f"Akého som typu: {kniha1}")
```

```
print(f"Co som zac: {kniha2}")
print(f"Akého som typu: {kniha2}")
```

```
print(dir(kniha1))
print(kniha1.__doc__)
```

```
kniha5 = Kniha()
kniha5.nazov = "OCA/OCP Java"
kniha5.autor = "Kathy Sierra"
print(kniha5)
```

```
kniha1: 140454210017024
kniha2: 140454210017648
kniha3: 140454210017648
meno: 140454210098160
vek: 140454276171136
Tieto objekty kniha1 a kniha2 su rovnako False
Tieto objekty kniha1 a kniha2 su rovnake: False
Tieto objekty kniha1 a kniha2 su rovnake: False
Tieto objekty kniha1 a kniha2 su rovnake: True
Nazov --> Head First Python
Autor -->
Pocet stran --> ()
Nazov --> Head First Python 2. edition
Autor -->
Pocet stran --> ()
Nazov --> Head First Python 2. edition
Autor -->
Pocet stran --> ()
Co som zac: Nazov objektu: Head First Python a pocet stran je: ()
Akého som typu: Nazov objektu: Head First Python a pocet stran je: ()
```

V. Konstruktory (init) a del + datetime


```

from datetime import datetime

class Byt:
    cena = float()
    adresa = str()
    vymera = float()
    lokalita = str()
    poschodie = int()
    datum_vystavby = datetime.now()

    def __init__(self):
        print("Hura som v konstruktore, objekt je vytvoreny...")
        self.cena = 200_000
        self.lokalita = "Bratislava Nove Mesto"

    def __del__(self):
        print("Hura som v destruktor, objekt je zruseny...")
        print(f"Dňa: {datetime.now()} sme vymazali objekt: {self.__class__}")

# Nazov objektu (byt1) = Volanie konstruktora, Typ/Trieda
byt1 = Byt()
byt2 = Byt()

del byt1, byt2
del Byt

```

```

Hura som v konstruktore, objekt je vytvoreny...
Hura som v konstruktore, objekt je vytvoreny...
Hura som v destruktor..., objekt je zruseny
2023-03-22 13:09:38.832582 sme vymazali objekt <class '__main__.Byt'>
Hura som v destruktor, objekt je zruseny...
Dňa: 2023-03-22 13:09:38.832744 sme vymazali objekt: <class '__main__.Byt'>
Hura som v destruktor, objekt je zruseny...
Dňa: 2023-03-22 13:09:38.833053 sme vymazali objekt: <class '__main__.Byt'>

```

```
!pip install prettytable
```

```
import prettytable
```

```

tabulka_zakaznici = prettytable.PrettyTable(["Meno", "Priezvisko", "Email", "Kont
print(tabulka_zakaznici)

```

```

Requirement already satisfied: prettytable in /opt/python/envs/default/lib/pyth
Requirement already satisfied: wcwidth in /opt/python/envs/default/lib/python3.
+-----+-----+-----+-----+
| Meno | Priezvisko | Email | Kontakt |

```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
#!/pip install prettytable

from prettytable import PrettyTable

tabulka_zakaznici = PrettyTable(["Meno", "Priezvisko", "Email", "Kontakt"])
print(tabulka_zakaznici)

tabulka_zakaznici.add_row(["Mario", "Sangala", "abc@nieco.sk", 123])
tabulka_zakaznici.add_row(["Ivana", "Sangala", "abc@nieco.sk", 456])
tabulka_zakaznici.add_row(["Monika", "Selicka", "abc@nieco.sk", 852])

print(tabulka_zakaznici)
```

```
+-----+-----+-----+-----+
| Meno | Priezvisko | Email | Kontakt |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|  Meno  | Priezvisko |   Email   | Kontakt |
+-----+-----+-----+-----+
| Mario  | Sangala    | abc@nieco.sk | 123    |
| Ivana  | Sangala    | abc@nieco.sk | 456    |
| Monika | Selicka    | abc@nieco.sk | 852    |
+-----+-----+-----+-----+
```

```
#modul copy

import copy
zoznam1 = list("ADAM")
# shallow copy (plytka kopia)
zoznam2 = copy.copy(zoznam1)
# deep copy (hlboka kopia)
zoznam3 = copy.deepcopy(zoznam1)

print(f"je zoznam1 a zoznam2 rovnaky: {zoznam1 is zoznam2}")
print(f"je zoznam3 a zoznam2 rovnaky: {zoznam3 is zoznam2}")
print(f"je zoznam1 a zoznam3 rovnaky: {zoznam1 is zoznam3}")
```