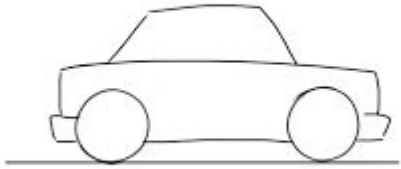


ZÁKLADY PROGRAMOVÁNÍ V JAZYCE PYTHON

LEKCE 2 - PROMĚNNÉ A STRUKTURY

TŘÍDA, INSTANCE (OBJEKT)



Třída: Auto

Metody:

- brm_brm()
- Tu_tůůů()

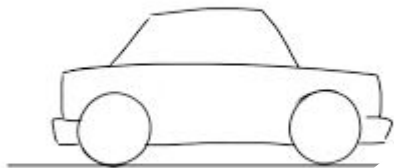


Instance(objekt): Blesk_McQueen

Metody:

- brm_brm()
- Tu_tůůů()

TŘÍDA, INSTANCE (OBJEKT)

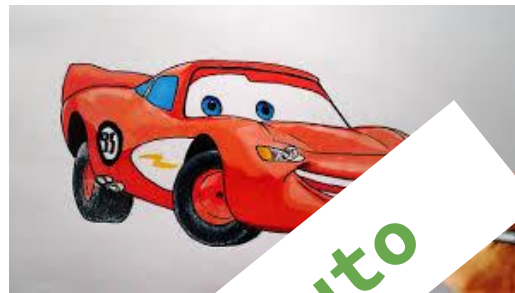


Třída: Auto

Metody:

- brm()
- tuůů()

Abstrakce auta



Instance (objekt): `objekt_McQueen`

Metody:

- brm()
- tuůů()

Konkrétní auto

NĚKTERÉ ZÁKLADNÍ TYPY TŘÍD

int	celá čísla	Číslo bez desetinné tečky	10
float	čísla s plovoucí tečkou	Číslo s desetinnou tečkou popř exponentem	7.82e2 (čti: $7,82 \cdot 10^2$)
string	textový řetězec	znaky	“Je 28. února”
list	Očíslovaný seznam	Jakékoli objekty indexované int	[1,7.82e2, “Ahoj”]
dictionary	Slovník	Jakékoli objekty indexované str	{‘věk’: 7, ‘barva’: ‘zelená’ }

NĚKTERÉ SPECIÁLNÍ TŘÍDY

Některé speciální objekty jsou vždy vytvořeny jenom jednou a vždycky mají stejnou adresu v paměti (v rámci jednoho spuštění)

None - objekt reprezentující "nic"

True, False - dva objekty typu **bool**

=

není

“rovná se”

KRABÍČKY

Co se děje když napíšu:

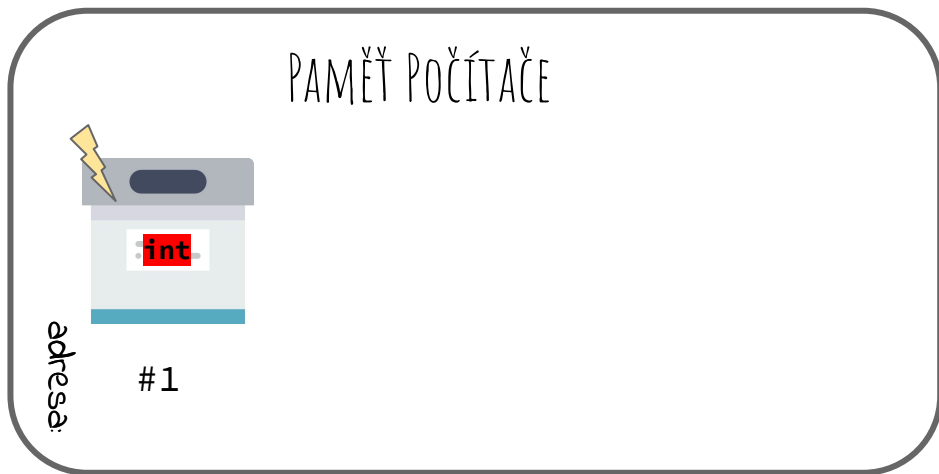
`a = 10`

`???`

KRABÍČKY

Co se děje když napíšu:

```
a = 10 + 12.5
```



1) Python se pokusí vyhodnotit pravou stranu od =

V našem případě si řekne: co tím programátor myslel?

Asi chtěl vytvořit nový objekt do kterého schová číslo 10

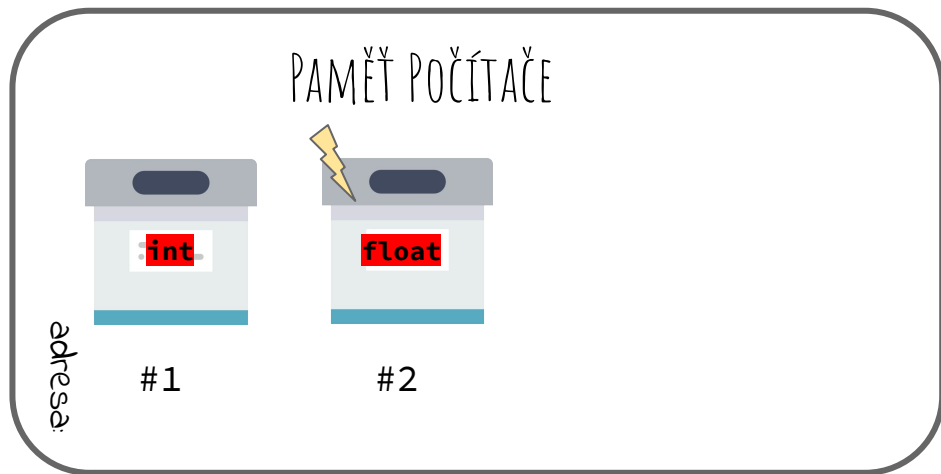
Tak to bude třeba objekt **int** pro celočíselné hodnoty

=> Python vytvoří nový objekt **int** s hodnotou 10

KRABÍČKY

Co se děje když napíšu:

```
a = 10 + 12.5
```



2) Je na pravé straně ještě něco? ANO

Je tam plus, jenže plus co???

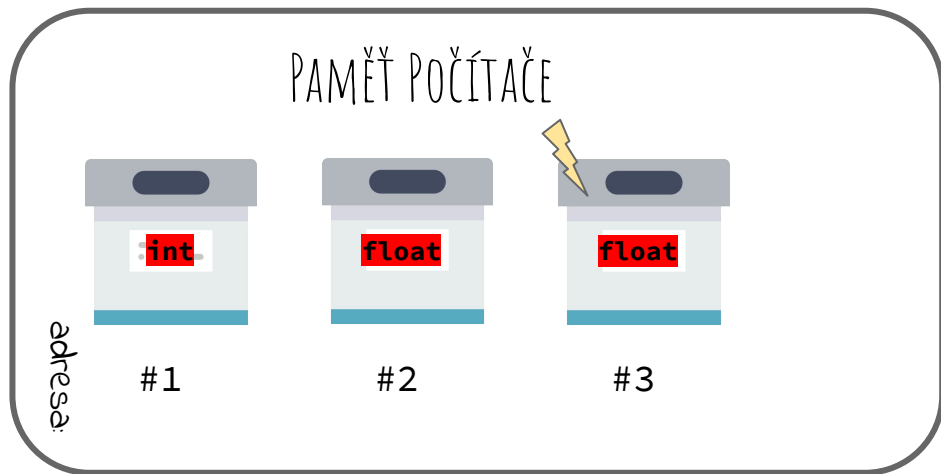
12.5... chmm... programátor asi myslí
číslo s desetinnou čárkou = **float**

=> Python vytvoří nový objekt
float s hodnotou 12.5

KRABÍČKY

Co se děje když napíšu:

```
a = 10 + 12.5
```



3) Je na pravé straně ještě něco? NE

Tak teď +

Python koukne jestli je definována funkce + mezi objekty **int** a **float**

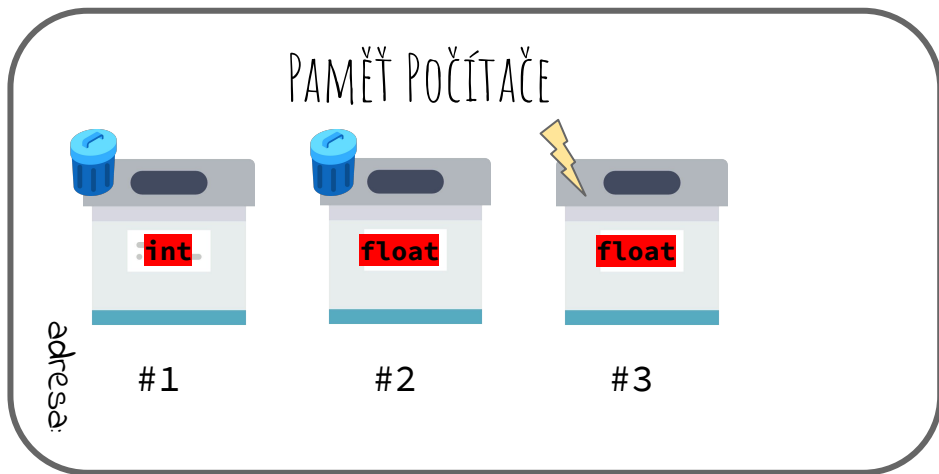
Hurá je! Tak ji použijem!

Funkce vrátí nový objekt typu **float**

KRABÍČKY

Co se děje když napíšu:

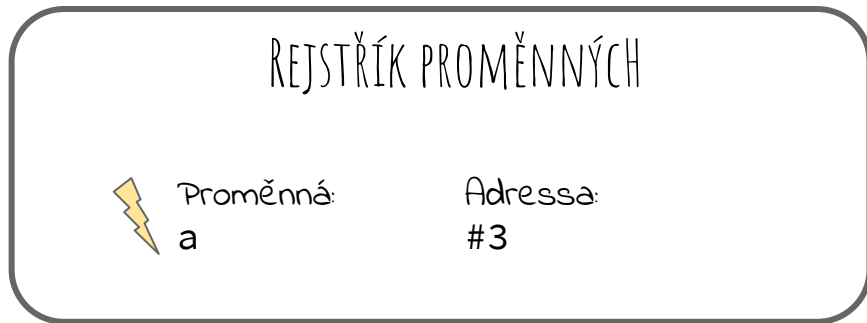
```
a = 10 + 12.5
```



- 4) Je na pravé straně ještě něco k řešení? NE

Adresa v paměti je přiřazena k proměnné v rejstříku proměnných

Ostatní objekty které nejsou nadále používané jsou vymazány z paměti



FUNKCE PRINT A TYPE

- **print(var)**
 - vypíše obsah proměnné `var`
- **type(var)**
 - vrátí typ proměnné `var`

Vyzkoušejte!

PŘÍKLAD

```
a = 12  
print(type(a))  
print(a)
```

Co se stalo tady?

```
b = str(a)  
print(type(b))  
print(b)
```

```
a = b  
print(type(a))  
print(a)
```

`=` není rovná se

```
a = int(a)
```

OPERACE S ČÍSLY

Operátor	Popis funkce	Příklad
+ Sčítání		$10 + 4 = 14$
- Odčítání		$10 - 4 = 6$
* Násobení		$10 * 4 = 40$
/ Dělení		$10 / 4 = 2.5$
% Modulo	Zbytek po dělení	$10 \% 4 = 2$
** Exponent	Umocnění	$10**4 = 10000$
//	Celočíselné dělení, zaokrouhuje dolů	$10//4 = 2$ $-10//4 = -3,$

Pozor: ^ je bitový operátor, ne umocňování:

```
print(2^4) #je 6...
```

RELAČNÍ OPERÁTORY

Operátor	Popis funkce	Příklad
<code>==</code>	Se rovná	<code>1 == 2</code> is False
<code>!=</code>	Se nerovná	<code>1 != 2</code> is True
<code>></code> (<code><</code>)	Větší (menší) než	<code>1 > 2</code> is False <code>1 < 2</code> is True
<code>>=</code> (<code><=</code>)	Větší (menší) nebo rovno	<code>2 >= 2</code> is True

Pozor: `=>` nebo `=<` nefungují.

OPERÁTORY IDENTITY A 'MEMBERSHIP' OPERÁTORY

Operátor	Popis funkce	Příklad
<code>is</code>	Porovnání jestli jsou obě proměnné cíli na stejný objekt	<code>type(2) is int is True</code>
<code>is not</code>	Porovnání jestli jsou obě proměnné cíli na jiný objekt	<code>type(2) is not int is False</code>
<code>in</code>	Vyhodnotí jestli je něco v něčem	<code>"Ahoj" in "Ahoj Karle" is True</code>

LOGICKÉ OPERÁTORY

Operátor	Popis funkce	Příklad
<code>and</code>	Logické “a”	<code>True and True is True</code> <code>True and False is False</code> <code>False and True is False</code> <code>False and False is False</code>
<code>or</code>	Logické “nebo”	<code>True or True is True</code> <code>True or False is True</code> <code>False or True is True</code> <code>False or False is False</code>
<code>not</code>	Logická “negace”	<code>not True is False</code> <code>not False is True</code>

POŘADÍ VYHODNOCOVÁNÍ

Dřív

() Věci v závorkách se vyhodnotí

nejdříve

**

*, /, %, //

+, -

<=, <, >, >=

==, !=

is, is not, in, not in

not, or, and

Později

Výrazy napravo se vyhodnotí dřív.

Příklad:

$x = y = 3$



PRÁCE S LOGICKÝMI OPERÁTORY

```
maji_chleba = False
penize = 50 # Kc
karta = False
cena_1_strycle = 32.90 # Kc

koupit_10_chlebu = (
    maji_chleba
    and ( (penize >= 10 * cena_1_strycle) or karta )
)
```

Úkol: upravte podmínky tak, aby se dalo koupit 10 chlebů.
Pozn.: Doporučujeme zkoušet po částech...

POŘÁDEK V JUPYTER NOTEBOOKU

jupyter

Untitled1

Klikněte sem a přejmenujte si notebook



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



In []:

- Run Cells
- Run Cells and Select Below
- Run Cells and Insert Below
- Run All
- Run All Above
- Run All Below
- Cell Type
 - Code
 - Markdown
 - Raw NBConvert
- Current Outputs
- All Output

Klikněte sem a budete moci vkládat nadpisy a formátovaný text.

```
# Hlavní nadpis
```

Hlavní nadpis

```
## Nadpis 2. úrovně
```

Nadpis 2. úrovně

PYTHON JAKO KALKULAČKA



Zadání:

Kolik energie je třeba k tomu, abyste se dostali z Koutů nad Desnou (562 m.n.m.) na vrchol Pradědu (1492 m.n.m.)? Kolik gramů hořké čokolády tuto energii obsahuje? Pro výpočet se aproximujte dokonalou koulí o hmotnosti 50 kg, která se valí bez jakéhokoliv odporu.

Nápověda:

- pro dokonalou kouli valící se bez odporu platí $E = m g \Delta h$
- <https://www.kaloricketabulky.cz/jemna-horka-cokolada-70-kakaa>

PYTHON JAKO KALKULAČKA

Změna zadání:

Jejda! Dali jsme si preventivně celou tabulku, ale měli jsme si dát jen **20 g**. Teď se kvůli nepromyšlené konzumaci čokolády naše hmotnost zdvojnásobila. Kolik energie a kolik čokolády to bude teď, když jsme dokonalou koulí o hmotnosti **100 kg**?

Použijte kód, který už máte, jen upravte hodnotu odpovídající proměnné!

PYTHON JAKO KALKULAČKA

Další změna zadání:

Jak by to vyšlo na Marsu? (`g_Mars = 0.38 * g_Earth`)

STRING - ŘETĚZEC ZNAKŮ

t	k	o	J	e	ž	u	r	á	t	k	o
-3	-2	-1	0	1	2	3	4	5	6	7	8

a = 'Ježurátko'

a[0] ?

a[:] ?

a[2:] ?

a[:2] ?

a[1:6] ?

a[2:-2] ?

a[-4:] ?

a[0:14] ?

a[::2] ?

a[1:8:2] ?

a[:: -1] ?

b = 2, c = 7

a[b:c] ?

b = 1, c = 2**2

a[b:c] ?

b = 1.1, c = 4

a[b:c] ?

CVIČENÍ - PŘEVOD ČÍSLA NA ŘETĚZEC

Opravte následující blok kódu a faktickou chybu (má být 4-):

```
a = 3 #tady neopravujte  
b = "Ježura má " + a + "-hlavý penis." #opravte tady  
print(b)
```

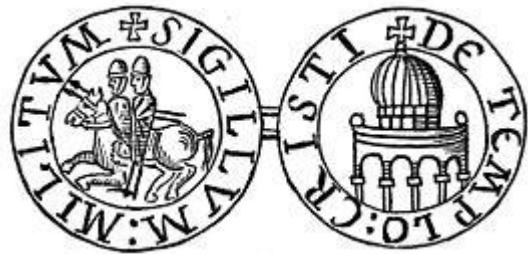
Nápověda: vzpomeňte si na funkci str().

ŠIFRA (HURVÍNEK'S CODE)

```
sifrovane_sdeleni_podle_Hurvinka = """Kčodčoyčožčo  
čotčoočohčolčoečo čovčoyčolčoučoščotčoičotčoečo,čo  
čojčosčotčoečo čončoečojčolčoečopčoščoičo"""
```

vyluštěte pomocí vhodného indexování []

TAJEMSTVÍ TEMPLÁŘŮ



Zakódujte následující řetězec znaků: rozdělte ho na dva podřetězce; první bude obsahovat liché a druhý sudé znaky. Ty pak spojte.

veta = 'Krutopřísně tajné sdělení obsahující tajemství k ovládnutí světa'



PONORKOVÁ ŠIFRA

Použijte předchozí metodu tentokráte s krokem 3 (tzn. každé třetí písmeno - dostanete tři stringy které sečtete.)

sifra = 'Kškon lívi,ik š tntoířl!ei!'

Tentokráte dostanete po rozkódování nejčastější zprávu, kterou si posílaly ponorky za II. světové války.



ZARÁMOVÁNÍ ŘETĚZCE

Napište skript, který vypíše řetězec ve správně velkém rámečku.

Příklad:

`s = "Ježurátko v rámečku"` $\xrightarrow{\text{Váš skript}}$

```
+-----+
| Ježurátko v rámečku |
+-----+
```

K řešení budete potřebovat:

```
len("aha")          # délka řetězce, výsledek je 3.
3*"la"             # násobení řetězců, výsledek je "lalala".
print()            # už znáte
```

Dnešní lekci Vám přinesla ježura, Váš oblíbený pojídač
mravenců.

