

**IB111**

**Základy programování**

**František Lachman**

**[lachmanfrantisek@mail.muni.cz](mailto:lachmanfrantisek@mail.muni.cz)**

**cvičení 2**

**26. září 2017**

# Osnova

- kontrolní otázky, opakování z minulého cvičení
- základní konstrukce
- příklady
- zadání domácí úlohy

# Kontrolní otázky (I.)

- Jak v Pythonu deklaruujeme proměnné?
- Jak se určuje typ proměnné?
- Jak zjistíme typ proměnné?
- Může se změnit typ proměnné?
- Jak lze přetypovat číslo na řetězec?

# Kontrolní otázky (II.)

- Jaký je rozdíl mezi `x = 3` a `x == 3`?
- Co znamená `x += 1`?
- Jak zapisujeme operaci "umocňování"?

# Kontrolní otázky (III.)

- Jaký je rozdíl mezi cykly `for` a `while`?
- Proč při programování používáme funkce?
- Jak zapisujeme definici funkce?
- Jak zapíšeme parametry s defaultní hodnotou?

# Kontrolní otázky (IV.)

- Jak v Pythonu vyznačujeme blok kódu?
- Co znamená [PEP8](#)?
- Jaké jsou příklady konvencí, které nejsou „povinné“, ale je velmi vhodné je dodržovat?
- Jak do zdrojového kódu zapisujeme komentáře?

# Základní struktury

# Proměnné

```
my_cool_variable = 1
print(my_cool_variable) # 1
second_variable = my_cool_variable
print(second_variable) # 1
second_variable = 2
print(my_cool_variable, second_variable) # 1 2
```



# Cyklus

```
for i in range(10):  
    print(i, end=" ")  
print()  
# 0 1 2 3 4 5 6 7 8 9
```

```
for i in range(5, 9):  
    print(i, end=" ")  
print()  
# 5 6 7 8
```

```
for i in range(1, 10, 2):  
    print(i, end=" ")  
print()  
# 1 3 5 7 9
```

# Range X seznam>>>

- `range(n)` - objekt "sekvenčního" typu
- `list(range(n))` - seznam vygenerovaný z prvků `range(n)`

```
>>> range(10)
range(0, 10)
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

# Funkce

```
def numbers(n):  
    for i in range(1, n + 1):  
        print(i, end=" ")  
    print()
```

```
>>> numbers(5)  
1 2 3 4 5
```

```
>>> numbers(10)  
1 2 3 4 5 6 7 8 9 10
```

# Příklady - 2.1. Posloupnosti

- 2.1.1. Sudá čísla

Napište funkci `even_numbers(n)`, která vypíše prvních  $n$  sudých čísel větších než 0.

- 2.1.2. Mocniny

Napište funkci `powers(base, n)`, která vypíše prvních  $n$  mocnin o daném základu.

- 2.1.4. Fibonacci

Napište funkci `fibonacci(n)`, která vypíše prvních  $n$  prvků Fibonacciho posloupnosti.

# Výpis bez přechodu na nový řádek

```
print(text, end=" ")
```

```
>>> print("1"); print("2")
```

```
1  
2
```

```
>>> print("1", end=" "); print("2")
```

```
1 2
```

# Příklady - 2.3. Textová grafika

- 2.3.1. Vyplněný čtverec

Napište funkci `square(n)`, která v textové grafice vykreslí vyplněný čtverec o straně  $n$ .

```
>>> square(5)
```

```
# # # # #
```

```
# # # # #
```

```
# # # # #
```

```
# # # # #
```

```
# # # # #
```

# Příklady - 2.3. Textová grafika

- 2.3.3. Pyramida

Napište funkci `pyramid(n)`, která v textové grafice vykreslí pyramidu o velikosti  $n$ .

```
>>> pyramid(5)
      #
     # # #
    # # # # #
   # # # # # # #
  # # # # # # # #
 # # # # # # # # #
```





# Příklady - 2.2. Tabulky

- 2.2.1. Násobilka

Napište funkci `table_products(n)`, která vypíše tabulku s daným počtem řádků a sloupců (+ popisný řádek a sloupce), kde v každé buňce se nachází součin čísla řádku a čísla sloupce.

```
>>> table_products(5)
      1 2 3 4 5
      - - - - -
1 | 1 2 3 4 5
2 | 2 4 6 8 10
3 | 3 6 9 12 15
4 | 4 8 12 16 20
5 | 5 10 15 20 25
```

# Příklady - 2.2. Tabulky

- 2.2.3. Sčítání `table_additions(n)`:

```
>>> table_additions(5)
```

```
      1 2 3 4 5
      - - - - -
1 | 2 3 4 5 6
2 | 3 4 5 6 7
3 | 4 5 6 7 8
4 | 5 6 7 8 9
5 | 6 7 8 9 10
```

- maximum, mocniny, zbytek po dělení, ...

# 1. Domácí úloha

- Deadline: 3. 10. 2017 23:59
- Odevzdání do odevzdáárny v ISe:  
Skupina 20 [Lachman]/Domácí úkol 1
- Jeden soubor `UČO.py` s hlavičkou:

```
"""  
uco: 123456  
jmeno: Jméno Příjmení  
hw: 1  
"""
```

# 1. Domácí úloha (1)

Napište funkce `alternating_multiples(n)` vypisující  
alternující násobky daného čísla:

```
>>> alternating_multiples(2)
0 -2 4 -6 8 -10 12 -14 16 -18 20
```

```
>>> alternating_multiples(3)
0 -3 6 -9 12 -15 18 -21 24 -27 30
```

# 1. Domácí úloha (2)

- Navrhněte funkci `crossing(n, length)` vykreslující přechod o `n` bílých pruzích délky `length`.

```
>>> crossing(4, 8)
# # # # # # # # # #
#                               #
# # # # # # # # # #
#                               #
# # # # # # # # # #
#                               #
# # # # # # # # # #
#                               #
# # # # # # # # # #
```

- Mezi vodorovnými `#` jsou mezery.

# 1. Domácí úloha (3)

- Napiště funkci `mocniny(m, n)`, která vypíše tabulku pro `m` čísel s prvními `n` mocninami.

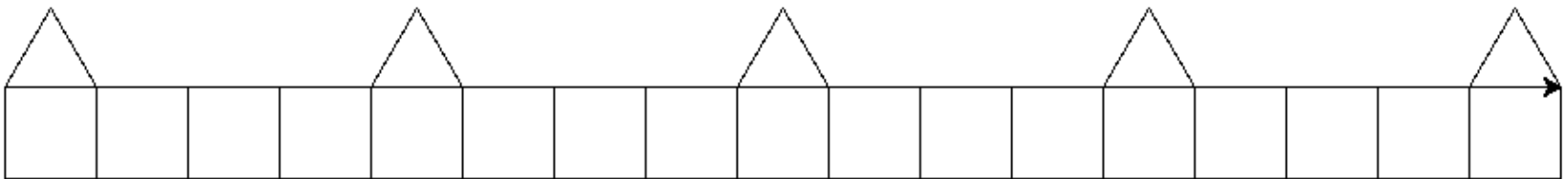
```
def mocniny(m, n):  
    pass
```

```
>>> mocniny(5, 3)  
    1 2 3 4 5  
    - - - - -  
1 | 1 2 3 4 5  
2 | 1 4 9 16 25  
3 | 1 8 27 64 125
```

# 1. Domácí úloha (4) 🐢

- Navrhněte funkci `castle(towers, length, space)`, která pomocí želví grafiky nakreslí hrad, který má daný počet věží (`towers`) a mezer mezi nimi (`space`). Hranu základních čtverců a trojúhelníků lze definovat parametrem `length`.

```
>>> castle(5, 50, 3)
```



# 1. Domácí úloha (5)

- Nakreslete pomocí želví grafiky hada
  - ideálně krajtů (=python..;-)

```
def draw_python()  
    pass
```

- Mělo by jít poznat, že se jedná o hada.
- Za pěkné provedení budou bonusové body.