# Rekurze

**Teorie: Rekurze je programovací technika, při níž je určitá procedura nebo funkce znovu volána dříve, než je dokončeno její předchozí volání.**

Obecný zápis rekurze:

**def funkce(promenna)**

**if (ukoncovaci podmínka):**

**return promenna**

**funkce (zmenena\_promenna)**

Jednoduchý příklad (vyzkoušejte i bez příkazu if, pak dojde k zacyklení)

def rekurze(hodnota):

if hodnota <= 10:

print(hodnota)

rekurze(hodnota+1)

rekurze(0)

**Příklad na rekurzi: výpočet faktoriálu.**

def factorial(x):

if x == 0:

return 1

elif x > 0:

return x \* factorial(x - 1)

print(factorial(5))

print("pro kontrolu:", 1 \* 2 \* 3 \* 4 \* 5)

Při rekurzi si musíte dát pozor, aby se někdy ukončila. Jinak program upadne na přetečení zásobníku. Příkaz Return 1 způsobí vždy ukončení funkce.

**Příklad: Fibonacciho posloupnost**

Jedná se o matematickou posloupnost, která začíná čísly 0 a 1. Každé další číslo je poté součet dvou předchozích čísel. Číslům v této posloupnosti se říká Fibonacciho čísla.

**Začátek posloupnosti tedy vypadá takto:**

****

def fib(n):

if n == 0:

return 0

elif n == 1:

return 1

return fib(n-1)+fib(n-2)

for i in range(10):

print(fib(i))

**Příklad potápěč, alias hromada lokálních proměnných**

def pruzkum(hloubka):

print(f'Rozhlížím se v hloubce {hloubka} m')

if hloubka >= 30:

print('Už toho bylo dost!')

else:

print(f'Zanořuju se (z {hloubka} m)')

pruzkum(hloubka + 10)

print(f'Vynořuju se (na {hloubka} m)')

pruzkum(0)

Na předchozím příkladu je vidět zajímavé chování lokálních proměnných. Když je potápěč „na dně“, jakou hodnotu má proměnná hloubka?

Tahle otázka je chyták. Která proměnná hloubka? Když je program „na dně“, existují čtyři různé lokální proměnné hloubka: každé zanoření, každé **zavolání funkce pruzkum, má vlastní proměnnou.**

Podobně jako když máte globální a lokální proměnnou stejného jména, každá funkce „vidí“ jen tu svoji proměnnou. Ale když se potápěč vynoří a volání funkce se ukončí, tato proměnná přestane existovat. A „volající“ funkce vidí svoji proměnnou, ve které je stále původní hodnota.