Programování 1: Funkce

Petr Čermák
cermak@mag.mff.cuni.cz

Katedra Fyziky Kondenzovaných Látek
MFF UK Praha

2020
forked from https://gitlab.kam.mff.cuni.cz/mj/prm1

Domácí úkol

Vypsat prvočísla do N
U většiny z vás selhali testy 4 a 5 (5-6 ciferná čísla)
Proč?

- cyklus 1-N není problém
- vnořený cyklus 1-N už je problém

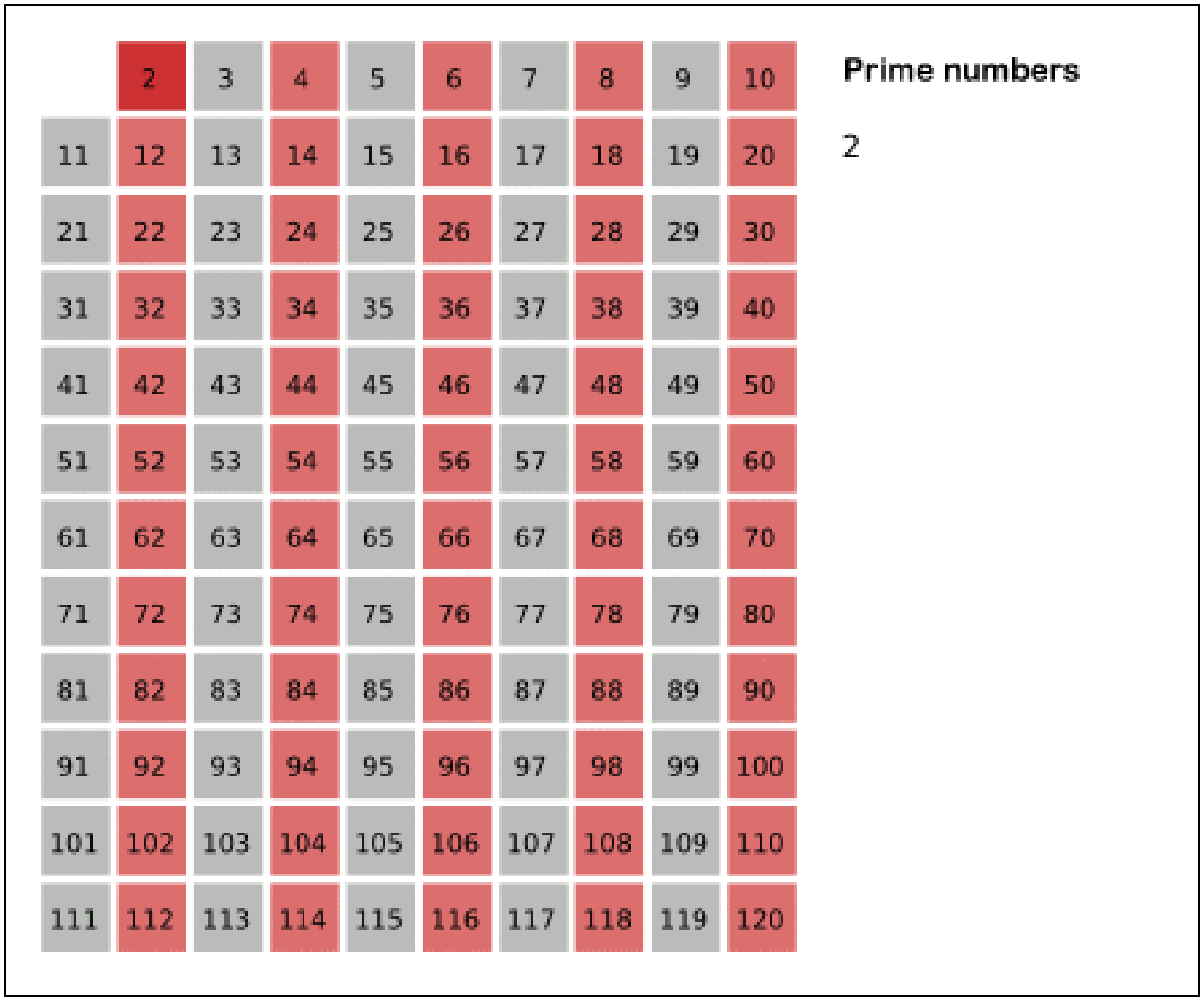
Jak optimalizovat řešení?

Nejčastější kód

```
1  def prvocisla(n):
2      out=[]
3      for i in range(2, n+1):
4          for d in range(2, i):
5              if i % d ==0:
6                  break
7          else:
8              out.append(i)
9      return out
```

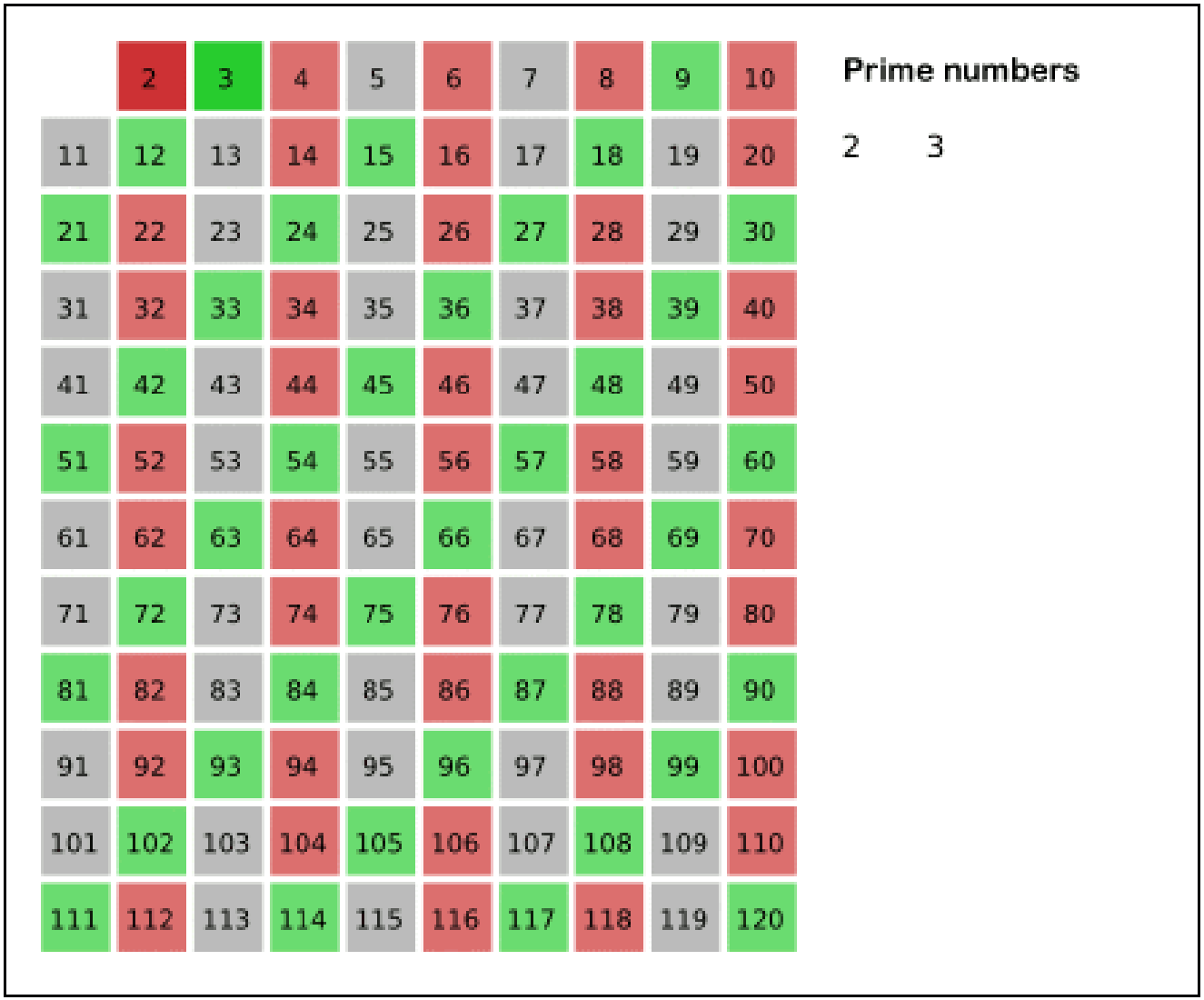
Erastothenovo síto

Jak optimalizovat řešení?



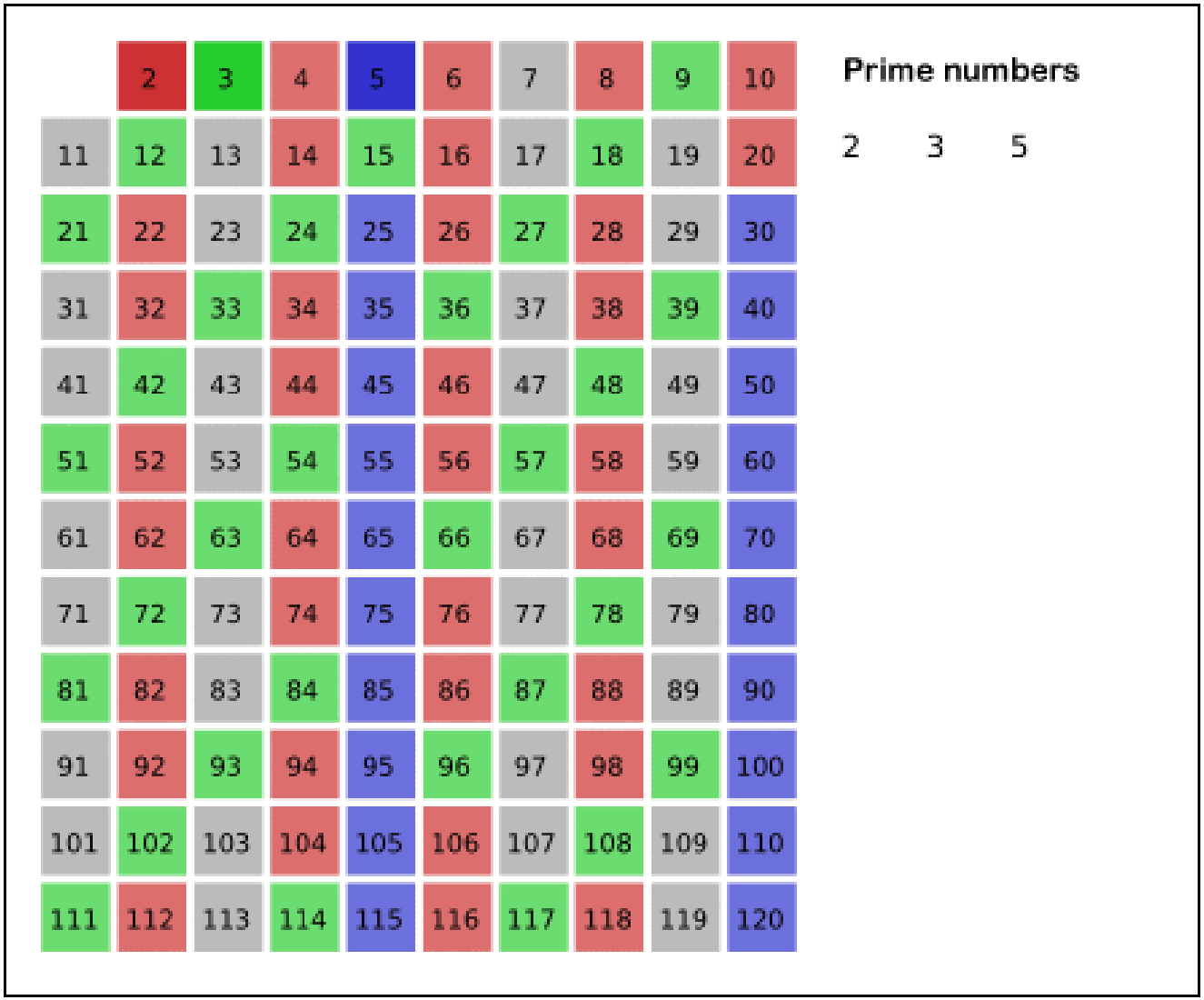
Erastothenovo síto

Jak optimalizovat řešení?



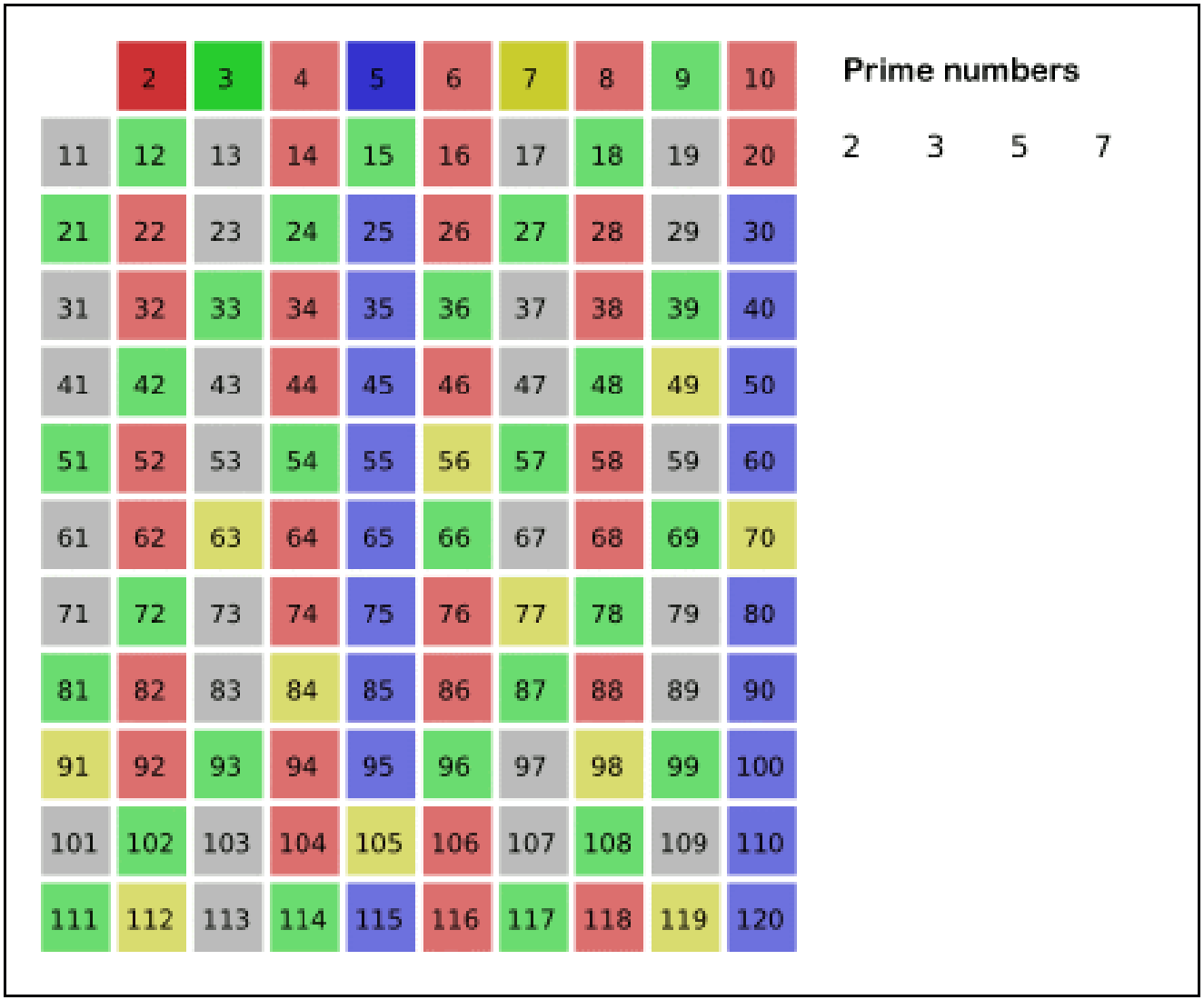
Erastothenovo síto

Jak optimalizovat řešení?



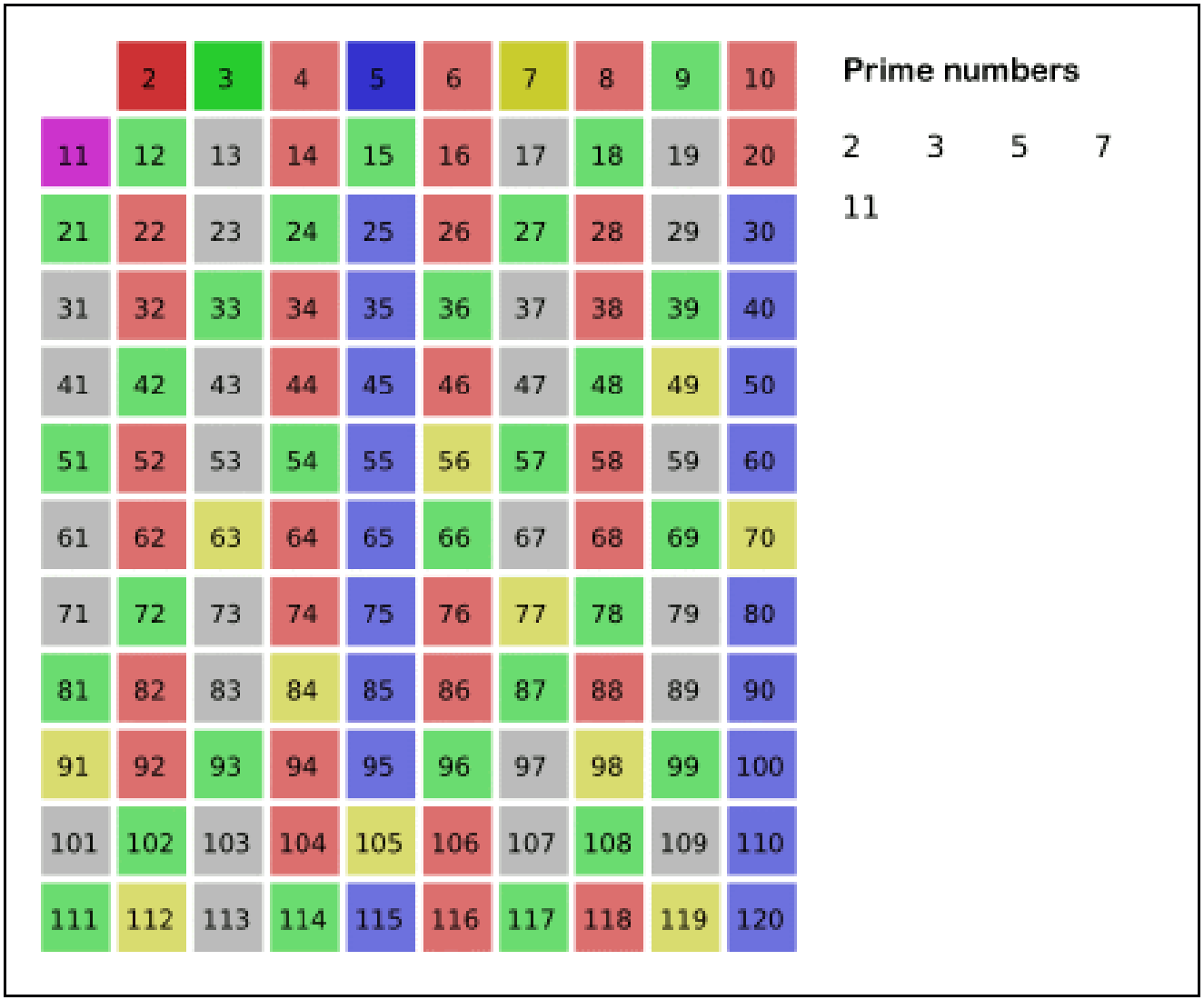
Erastothenovo síto

Jak optimalizovat řešení?



Erastothenovo síto

Jak optimalizovat řešení?



Definice funkce

```
1 def stekej():
2     print("Haf!")
3
4 stekej()
5 stekej()
```

- Opakující se část programu stačí napsat jednou a spustit vícekrát.
- Také jsme tím část programu pojmenovali, takže je při čtení jasné, co má dělat.

Funkce s parametrem

```
1 def stekej(n):
2     for i in range(n):
3         print("Haf!")
4
5 stekej(5)
```

- Funkci můžeme předat parametr.
- Když zavoláme `stekej(5)`, funkce se spustí s proměnnou `n` nastavenou na `5`.
- Proměnné `n` a `i` existují jen lokálně uvnitř funkce, zvenčí nejsou vidět.

Funkce s výsledkem

```
1 def minimum(a, b):
2     if a < b:
3         return a
4     else:
5         return b
6
7 print(minimum(3, 5))
```

- Příkaz **return** ukončí vykonávání funkce a vrátí výsledek.
- Můžeme použít i **return** bez výsledku (vrátí **None**).
- Výsledek funkce nemusíme použít.

Ještě o parametrech

```
1  # Nepovinné parametry
2
3  def stekej(n=1, zvuk="Haf!"):
4      for i in range(n):
5          print(zvuk)
6
7
8  # Různé způsoby volání:
9
10 stekej()
11 stekej(5)
12 stekej(5, "HAF!")
13 stekej(5, zvuk="HAF!")
14 stekej(zvuk="HAF!")
```

Viditelnost proměnných: chyták

```
1 zvuk = "Kuku!"
2 kolik_hodin = 0
3
4 def zakukej():
5     print(zvuk)
6     kolik_hodin += 1
```

- Funkce vidí globální proměnnou **zvuk**
- Proměnná **kolik_hodin** je ale lokální, protože funkce do ní přiřazuje.

Viditelnost proměnných: řešení

```
1  zvuk = "Kuku!"
2  kolik_hodin = 0
3
4  def zakukej():
5      global kolik_hodin
6      print(zvuk)
7      kolik_hodin += 1
```

Úkoly na hodinu

Napište funkci `je_sude(n)`, která vrátí True - False

Napište funkci `kolik_sudych(list)`, která vrátí počet sudých čísel (použijte předchozí funkci)

Napište funkci `vyber_suda(list)`, která list jen se sudými čísly

Napište funkci `kvadraticka(a, b, c)`, která vrací list (0-2) reálných řešení.