

Programování 1: Podmínky a cykly

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>

Řídicí příkazy: podmínky

if *podmínka*:

příkazy

else:

příkazy

Větev **else** můžeme vynechat.

Pokud bychom chtěli vynechat *příkazy*, můžeme místo nich napsat **pass**. To je prázdný příkaz.

Řídicí příkazy: řetězení podmínek

Abychom nemuseli psát:

if *podmínka:*

příkazy

else:

if *podmínka:*

příkazy

else:

příkazy

Existuje zkratka:

if *podmínka:*

příkazy

elif *podmínka:*

příkazy

else:

příkazy

Řídicí příkazy: cykly

while *podmínka:*
příkazy

break – vyskočí z cyklu

continue – přejde na další průchod cyklem

while True – nekonečný cyklus

Příklad: testování prvočísel

```
1 #!/usr/bin/env python3
2 # Otestuje, zda číslo je prvočíslem
3
4 n = int(input())
5 d = 2
6 mam_delitele = False
7
8 while d < n:
9     if n%d == 0:
10        print(f"Číslo {n} je dělitelné {d}")
11        mam_delitele = True
12        break
13    d += 1
14
15 if not mam_delitele:
16    print(f"Číslo {n} je prvočíslo")
```

Příklad: jak funguje else za cyklem

```
1 #!/usr/bin/env python3
2 # Otestuje, zda číslo je prvočíslem (2. pokus)
3
4 n = int(input())
5 d = 2
6
7 while d < n:
8     if n%d == 0:
9         print(f"Číslo {n} je dělitelné {d}")
10        break
11    d += 1
12 else:
13    print("Číslo {n} je prvočíslo")
```

Příklad: sčítáme posloupnost čísel

```
1  #!/usr/bin/env python3
2  # Načítá čísla ze vstupu ukončená -1,
3  # vypíše jejich součet
4
5  s = 0
6
7  n = int(input())
8  while n != -1:
9      s += n
10     n = int(input())
11
12 print(s)
```

Příklad: sčítáme posloupnost čísel

```
1  #!/usr/bin/env python3
2  # Načítá čísla ze vstupu ukončená -1,
3  # vypíše jejich součet
4
5  s = 0
6
7  while True:
8      n = int(input())
9      if n == -1:
10         break
11         s += n
12
13  print(s)
```


Příklad: sčítáme posloupnost čísel v Pythonu 3.8

Novinka v Pythonu 3.8: „mroží“ operátor (assignment expressions / přiřazovací výrazy).

:= přiřadí hodnotu do proměnné a zároveň hodnoty nabývá

```
1  #!/usr/bin/env python3
2  # Načítá čísla ze vstupu ukončená -1,
3  # vypíše jejich součet
4  # používáme mrože (walrus operator)
5  # nutný python 3.8
6
7  s = 0
8
9  while (n := int(input())) != -1:
10     s += n
11
12 print(s)
```

Úkol na zbytek hodiny

Spočítejte, kolik má zadané číslo cifer.