**Jakou sekvenci dostaneš, když funkci range zavoláš se dvěma argumenty? A se třemi?**

for i in range(3, 7):

print(i, end=' ')

sekvence:3 4 5 6

for i in range(2, 6, 8):

print(i, end=' ')

výstup bude 2

oprava: prázdný výstup, číslo 8 je moc velké, aby mohlo generovat číslo mezi počáteční a konečnou hodnotou

for i in range(2, 6, 2):

print(i, end=' ')

výstup: 2,4

číslo 2(start) se zvýšií o step(2), inkrementuje se vždy o 2, přičemž 6(stop) není zahrnuta

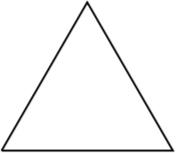
Oprava: ano omlouvám se výstup bude 2, nelze ale inkrementovat 2 o číslo (step)8, když stop 6, takže výstup jen 2

for i in range(2, 6, 8):

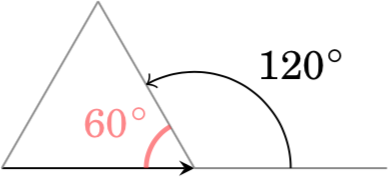
print(i, end=' ')

výstup bude 2

**Nakresli trojúhelník.**



**Poznámka:** Rovnostranný trojúhelník má vnitřní úhly 60°. Želva se ale otáčí o vedlejší úhel 180 - 60 = 120°.



from turtle import forward, left, exitonclick

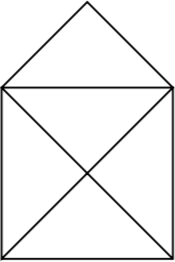
for i in range(3):

forward(100)

left(120)

exitonclick()

**Nakresli domeček!**



Jak pravil Pythagoras, délka šikmé čáry v domečku je √2-krát délka stěny. Funkci na odmocninu najdeš na taháku.

from turtle import forward, left,right, exitonclick,goto

#nakreslení trojuhelníka

for i in range(3):

forward(100) # Délka strany trojúhelníka

left(120) # Vnitřní úhel trojúhelníka který má 60stupnů

# Natočení o 90 stupňů doprava pro nakreslení čtverce

right(90)

# Nakreslení čtverce

for i in range(4):

forward(100) # Délka strany čtverce

left(90) # Vnitřní úhel čtverce, který má 90 stupňů

# Nakreslení šikmé čáry od rohu k rohu čtverce

goto(0, 0) # Návrat na počáteční bod

forward(100) # Posun na protilehlý roh čtverce

left(135) # Otočení o 135 stupňů (45 stupňů představuje jeden vnitřní úhel čtverce)

# Nakreslení čáry k protilehlému rohu čtverce

forward(141)# Délka úhlopříčky čtverce (dvojnásobek délky strany čtverce)

left(135)

forward(100)

left(135)

forward(141)

exitonclick()

--úprava--

from turtle import forward, left,right, exitonclick,goto

#nakreslení trojuhelníka

for i in range(3):

forward(100) # Délka strany trojúhelníka

left(120) # Vnitřní úhel trojúhelníka který má 60stupnů

# Natočení o 90 stupňů doprava pro nakreslení čtverce

right(90)

# Nakreslení čtverce

for i in range(4):

forward(100) # Délka strany čtverce

left(90) # Vnitřní úhel čtverce, který má 90 stupňů

# Výpočet délky úhlopříčky čtverce

delka\_strany = 100

delka\_uhlopricky = (2 \* delka\_strany \*\* 2) \*\* 0.5

# Posun na protilehlý roh čtverce

goto(0,0)

# Otočení a nakreslení střechy domečku

left(45)

forward(delka\_uhlopricky)

left(135)

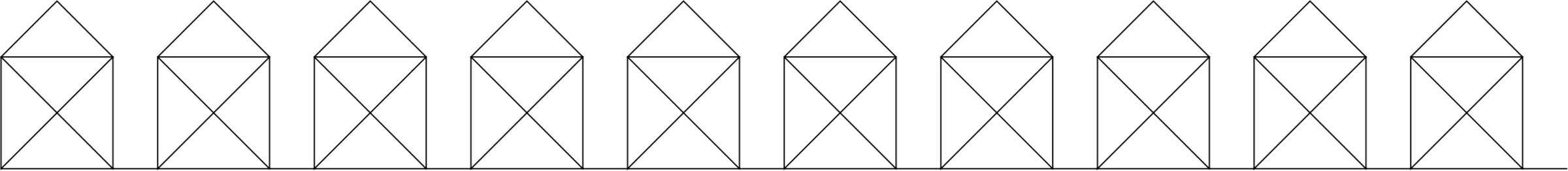
forward(100)

left(135)

forward(delka\_uhlopricky)

exitonclick()

Nakresli vesnici!



Jestli kreslení trvá dlouho a nebaví tě čekat, můžeš naimportovat a použít funkci [speed](https://docs.python.org/3/library/turtle.html#turtle.speed), která nastavuje rychlost dalšího kreslení:

from turtle import speed

speed(1) # želva jak má být – extrémně pomalá

speed(3) # pomalá želva

speed(6) # normálně rychlá želva

speed(10) # hodně rychlá želva

speed(0) # turboželva

from turtle import forward, left, right, exitonclick, goto

# Nakreslení trojúhelníka

for i in range(3):

forward(100) # Délka strany trojúhelníka

left(120) # Vnitřní úhel trojúhelníka, který má 60 stupňů

# Natočení o 90 stupňů doprava pro nakreslení čtverce

right(90)

# Nakreslení čtverce

for i in range(4):

forward(100) # Délka strany čtverce

left(90) # Vnitřní úhel čtverce, který má 90 stupňů

# Výpočet délky úhlopříčky čtverce

delka\_strany = 100

delka\_uhlopricky = (2 \* delka\_strany \*\* 2) \*\* 0.5

# Posun na protilehlý roh čtverce

goto( 0, 0)

# Otočení a nakreslení střechy domečku

left(45)

forward(delka\_uhlopricky)

left(135)

forward(100)

left(135)

forward(delka\_uhlopricky)

# Nový blok

right(45) # Narovnání želvy

forward(20) # Rovně

right(90) # Narovnat nahoru

# Definice funkce pro nakreslení jednoho domečku

def nakresli\_domecek():

for i in range(3):

forward(100)

left(90)

forward(100)

left(135)

forward(delka\_uhlopricky)

right(75)

forward(100)

right(120)

forward(100)

right(75)

forward(delka\_uhlopricky)

right(45)

forward(20)

right(90)

# forward(100) # Posun dopředu pro nový domeček

# Nakreslení 8 dalších domečků vedle sebe

for i in range(8):

nakresli\_domecek()

exitonclick()

Nakresli pravidelný n-úhelník, kde n zadá uživatel.

Vnitřní úhel pravidelného *n*-úhelníka má *180 × (1-2/n)* stupňů.

Aby byly tvary zhruba stejně veliké, použij pro *n*-úhelník délku strany např. *200/n*

from turtle import forward, left, exitonclick

# Získání počtu stran od uživatele

n = int(input("Zadej počet stran n-úhelníku: "))

# Délka strany pravidelného n-úhelníku

side\_length = 200 / n

# Výpočet vnitřního úhlu pravidelného n-úhelníku

interior\_angle = 180 \* (1 - 2 / n)

# Nakreslení pravidelného n-úhelníku

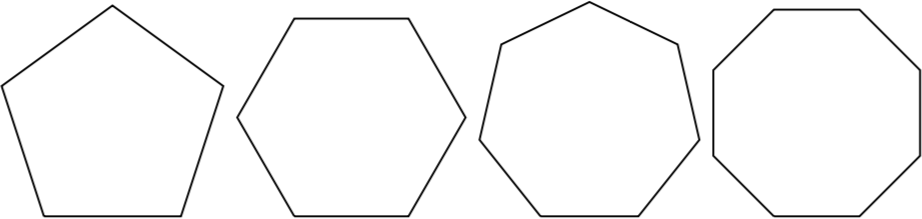
for \_ in range(n):

forward(side\_length)

left(180 - interior\_angle)

exitonclick()

Nakresli vedle sebe pětiúhelník, šestiúhelník, sedmiúhelník, osmiúhelník.



from turtle import forward, left, penup, pendown, exitonclick

# Délka strany pro každý úhelník

side\_length = 100

# Vnitřní úhel pro každý úhelník

interior\_angles = [180 \* (1 - 2 / n) for n in range(5, 9)]

# Posun želvy doleva pro lepší viditelnost

penup()

left(90)

forward(100)

left(90)

forward(200)

left(180)

pendown()

# Nakreslení pětiúhelníku

for \_ in range(5):

forward(side\_length)

left(180 - interior\_angles[0])

# Přesun na novou pozici pro šestiúhelník

forward(2 \* side\_length)

# Nakreslení šestiúhelníku

for \_ in range(6):

forward(side\_length)

left(180 - interior\_angles[1])

# Přesun na novou pozici pro sedmiúhelník

forward(2.5 \* side\_length)

# Nakreslení sedmiúhelníku

for \_ in range(7):

forward(side\_length)

left(180 - interior\_angles[2])

# Přesun na novou pozici pro osmiúhelník

forward(2.5 \* side\_length)

# Nakreslení osmiúhelníku

for \_ in range(8):

forward(side\_length)

left(180 - interior\_angles[3])

exitonclick()

--úprava--

from turtle import forward, left, penup, pendown, exitonclick

# Délka strany pro každý úhelník

side\_length = 100

# Vnitřní úhel pro každý úhelník

interior\_angles = [180 \* (1 - 2 / n) for n in range(5, 9)]

# Posun želvy doleva pro lepší viditelnost

penup()

left(180)

forward(300)

left(180)

pendown()

# Cyklus pro nakreslení polygonů

for n, angle in zip(range(5, 9), interior\_angles):

# Nakreslení polygonu

for \_ in range(n):

forward(side\_length)

left(180 - angle)

# Posun na novou pozici pro další polygon

forward(2.5 \* side\_length)

exitonclick()

Želva umí jen rovné čáry. Jde pomocí rovných čar nakreslit kolečko?

Nevíš-li, přeskoč na další projekt a pak se sem vrať.

import turtle

# Inicializace želvy

t = turtle.Turtle()

# Nakreslení kruhu

t.circle(100) # 100 určuje poloměr kruhu

# Ukončení programu po kliknutí

turtle.done()

**Nakresli pravidelný stoúhelník.**

**from turtle import forward, left, exitonclick**

**# Počet stran pravidelného stoúhelníka**

**n = 100**

**# Délka strany pravidelného stoúhelníka**

**side\_length = 10**

**# Výpočet úhlu mezi stranami pravidelného stoúhelníka**

**angle = 360 / n**

**# Nakreslení pravidelného stoúhelníka**

**for \_ in range(n):**

**forward(side\_length)**

**left(angle)**

**exitonclick()**

**Nakresli takovýto ornament:**

**from turtle import forward, left,shape, exitonclick**

**import random**

**def ornament(x):**

**shape("turtle")**

**pocetCar = x \* 5**

**posun = 20 / x**

**for i in range(pocetCar):**

**forward(i \* posun)**

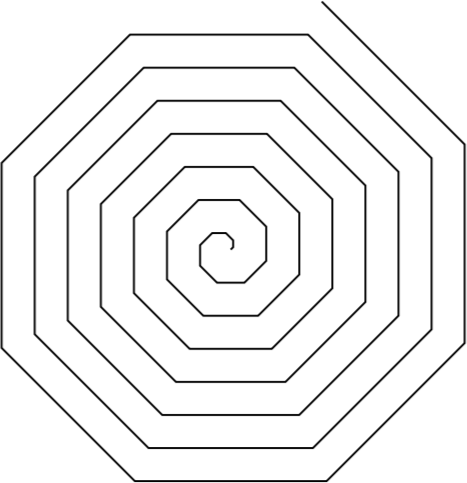
**left(360 / x)**

**x = int(input("Zadej kolikatiúhelník chceš vykreslit: "))**

**ornament(x)**

**exitonclick()**

Nakresli takovýto ornament:



**from turtle import \***

**import random**

**def ornament(x):**

**shape("turtle")**

**pocetCar = x \* 7**

**posun = 20 / x**

**for i in range(pocetCar):**

**forward(i \* posun)**

**left(360 / x)**

**x = int(input("Zadej kolikatiúhelník chceš vykreslit: "))**

**ornament(x)**

**done()**

**--úprava--**

**from turtle import forward, left,shape, exitonclick**

**import random**

**def ornament(x):**

**shape("turtle")**

**pocetCar = x \* 7**

**posun = 20 / x**

**for i in range(pocetCar):**

**forward(i \* posun)**

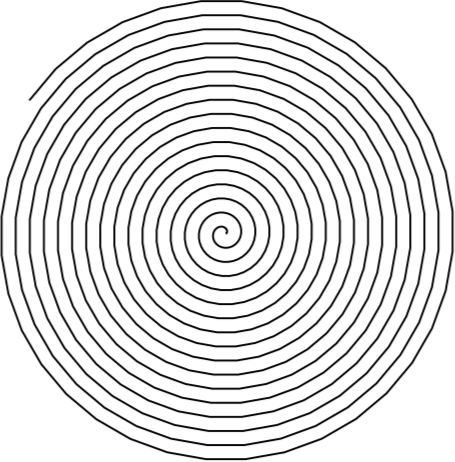
**left(360 / x)**

**x = int(input("Zadej kolikatiúhelník chceš vykreslit: "))**

**ornament(x)**

**exitonclick()**

Nakresli spirálu.



Od ornamentů ke spirále se dostaneš stejným způsobem jako od *n*-úhelníků ke kolečku.

**from turtle import forward, left,shape, exitonclick**

**import random**

**# Získání počtu stran od uživatele**

**x = 100**

**def ornament(x):**

**shape("turtle")**

**pocetCar = x \* 16**

**posun = 1 / x**

**for i in range(pocetCar):**

**forward(i \* posun)**

**left(360 / x)**

**x = int(input("Zadej kolikatiúhelník chceš vykreslit: "))**

**ornament(x)**

**exitonclick()**

Napiš program, který se zeptá na 3 čísla a zjistí, jestli je jejich součet větší než 10.

**cislo1 = float(input("Zadej první číslo: "))**

**cislo2 = float(input("Zadej druhé číslo: "))**

**cislo3 = float(input("Zadej třetí číslo: "))**

**soucet = cislo1 + cislo2 + cislo3**

**if soucet > 10:**

**print("Součet těchto čísel je větší než 10.")**

**else:**

**print("Součet těchto čísel není větší než 10.")**

**příklad:**

**(venv) C:\Users\Marta\Documents\PythonPyLadies\03>python ukol17.py**

**Zadej první číslo: 3**

**Zadej druhé číslo: 2**

**Zadej třetí číslo: 7**

**Součet těchto čísel je větší než 10.**

**(venv) C:\Users\Marta\Documents\PythonPyLadies\03>python ukol17.py**

**Zadej první číslo: 2**

**Zadej druhé číslo: 2**

**Zadej třetí číslo: 3**

**Součet těchto čísel není větší než 10.**

Napiš program, který se zeptá na 10 čísel a zjistí, jestli je jejich součet větší než 1000.

**soucet = 0**

**for i in range(10):**

**cislo = float(input(f"Zadej {i + 1}. číslo: "))**

**soucet += cislo**

**if soucet > 1000:**

**print("Součet těchto čísel je větší než 1000.")**

**else:**

**print("Součet těchto čísel není větší než 1000.")**

**příklad:**

**(venv) C:\Users\Marta\Documents\PythonPyLadies\03>python ukol18.py**

**Zadej 1. číslo: 333**

**Zadej 2. číslo: 333**

**Zadej 3. číslo: 200**

**Zadej 4. číslo: 2**

**Zadej 5. číslo: 3**

**Zadej 6. číslo: 16**

**Zadej 7. číslo: 5**

**Zadej 8. číslo: 7**

**Zadej 9. číslo: 21**

**Zadej 10. číslo: 3**

**Součet těchto čísel není větší než 1000.**

Napiš program, který postupně načte od uživatele dvě čísla a jednoznakový řetězec – buď '+', '-', '\*' nebo '/'. Program provede na číslech příslušnou operaci.

Příklad použití programu (hodnoty za dvojtečkou doplnil uživatel):

První číslo: 123

Druhé číslo: 456

Operace: +

123 + 456 = 579

# Načtení prvního čísla od uživatele

prvni\_cislo = int(input("První číslo: "))

# Načtení druhého čísla od uživatele

druhe\_cislo = int(input("Druhé číslo: "))

# Načtení operace od uživatele

operace = input("Operace (+, -, \*, /): ")

# Provedení požadované operace na zadaných číslech

if operace == '+':

vysledek = prvni\_cislo + druhe\_cislo

elif operace == '-':

vysledek = prvni\_cislo - druhe\_cislo

elif operace == '\*':

vysledek = prvni\_cislo \* druhe\_cislo

elif operace == '/':

vysledek = prvni\_cislo / druhe\_cislo

else:

print("Špatný vstup.")

# Vytisknutí výsledku

print(f"{prvni\_cislo} {operace} {druhe\_cislo} = {vysledek}")

úkol:

(venv) C:\Users\Marta\Documents\PythonPyLadies\03>python ukol19.py

První číslo: 30

Druhé číslo: 20

Operace (+, -, \*, /): +

30 + 20 = 50

--uprava--

# Načtení prvního čísla od uživatele

prvni\_cislo = int(input("První číslo: "))

# Načtení druhého čísla od uživatele

druhe\_cislo = int(input("Druhé číslo: "))

# Načtení operace od uživatele

operace = input("Operace (+, -, \*, /): ")

# Provedení požadované operace na zadaných číslech

if operace == '+':

vysledek = prvni\_cislo + druhe\_cislo

elif operace == '-':

vysledek = prvni\_cislo - druhe\_cislo

elif operace == '\*':

vysledek = prvni\_cislo \* druhe\_cislo

elif operace == '/':

if druhe\_cislo == 0:

print("Nelze dělit nulou.")

else:

vysledek = prvni\_cislo / druhe\_cislo

else:

print("Špatný vstup.")

# Vytisknutí výsledku (pokud není dělení nulou)

if operace != '/' or druhe\_cislo != 0:

print(f"{prvni\_cislo} {operace} {druhe\_cislo} = {vysledek}")

Napiš program, který se pětkrát zeptá na číslo a nejmenší zadané číslo vypíše.

Program by mělo jít snadno upravit tak, aby se ptal na deset nebo sto čísel.

Nevíš-li si rady, podívej se do materiálů k Přepisování proměnných. Místo přičítání do proměnné *celkem* zkus dávat správnou hodnotu do proměnné *zatim\_nejmensi*.

nejmensi\_cislo = None

for i in range(5):

cislo = int(input("Zadej číslo: "))

if nejmensi\_cislo is None or cislo < nejmensi\_cislo:

nejmensi\_cislo = cislo

# Výpis nejmenšího čísla

print("Nejmenší zadané číslo je:", nejmensi\_cislo)

příklad:

Zadej číslo: 1

Zadej číslo: 2

Zadej číslo: 3

Zadej číslo: 4

Zadej číslo: 5

Nejmenší zadané číslo je: 1

Naprogramuj hru *Oko bere*:

* Začínáš se *skóre* 0 bodů.
* V každém kole:
  + Počítač vypíše, kolik máš bodů.
  + Počítač se zeptá, jestli chceš pokračovat.
  + Pokud byla odpověď „ne“:
    - Hra končí.
  + Jinak:
    - Počítač „vybere kartu“ – náhodně vybere číslo od 2 do 10;
    - vybranou hodnotu vypíše;
    - přičte tuto hodnotu ke *skóre*.
  + Pokud máš víc než 21 bodů:
    - Počítač vypíše, že prohráváš;
    - hra končí.
* Po skončení hry počítač vypíše dosažené *skóre*.

Cílem hry je neprohrát a získat přitom co nejvíc bodů, ideálně 21.

from random import randrange

soucet = 0

while soucet < 21:

print('Máš', soucet, 'bodů')

odpoved = input('Otočit kartu? ')

if odpoved == 'ano':

karta = randrange(2, 11)

print('Otočil jsi', karta)

soucet = soucet + karta

elif odpoved == 'ne':

break

else:

print('Nerozumím! Odpovídej "ano", nebo "ne"')

if soucet == 21:

print('Gratuluji! Vyhrála jsi!')

elif soucet > 21:

print('Smůla!', soucet, 'bodů je moc!')

else:

print('Chybělo jen', 21 - soucet, 'bodů!')