

# Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

- Zmienne (definiowanie)  
i typy zmiennych  
(`int`, `float`, `str`)

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

- Zmienne (definiowanie)  
i typy zmiennych  
(`int`, `float`, `str`)
- Instrukcje warunkowe

```
if warunek:  
    ...  
elif warunek:  
    ...  
else:
```

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

- Zmienne (definiowanie)  
i typy zmiennych  
(`int`, `float`, `str`)
- Instrukcje warunkowe

```
if warunek:  
    ...  
elif warunek:  
    ...  
else:
```

- Pętle

```
while warunek:
```

oraz

```
for i in range(n):
```

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

- Zmienne (definiowanie)  
i typy zmiennych  
(`int`, `float`, `str`)
- Instrukcje warunkowe

```
if warunek:  
    ...  
elif warunek:  
    ...  
else:
```

- Pętle

```
while warunek:
```

oraz

```
for i in range(n):
```

- Importowanie bibliotek:

```
from turtle import Turtle  
from turtle import *  
import turtle  
import turtle as tl
```

## Podsumowanie — *Czego się dotychczas nauczyliśmy:*

- Składnia Pythona
- Funkcje wejścia/wyjścia:

```
print("text")  
zmienna = input("Pytanie")
```

- Zmienne (definiowanie)  
i typy zmiennych  
(`int`, `float`, `str`)
- Instrukcje warunkowe

```
if warunek:  
    ...  
elif warunek:  
    ...  
else:
```

- Pętle

```
while warunek:
```

oraz

```
for i in range(n):
```

- Importowanie bibliotek:

```
from turtle import Turtle  
from turtle import *  
import turtle  
import turtle as tl
```

- Listy:

```
lista = [1,2,4,2.5,3]
```

elementy listy: `lista[i]`

Co w najbliższym czasie:

- funkcje (w pythonie)
- jeszcze o listach
- liczby zespolone
- wektory i macierze
- biblioteka [numpy](#)
- wykresy funkcji (*matematycznych*) w Pythonie



# Zadania

## 1. Wyścig żółwi :)

Zaimportuj bibliotekę `turtle`. Z biblioteki `random` zaimportuj funkcję `randrange`.

Za pomocą funkcji `turtle.Turtle()` stwórz dwa żółwie: `asia` i `zenek`. Nadaj im różne kolory (`imiezolwia.color('nazwakoloru')`) i kształt żółwia (`imiezolwia.shape('turtle')`).

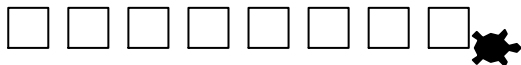
Stwórz żółwia `sedzia` (bez polskich liter!) i narysuj nim linię startu (od `(-300,-50)` do `(-300,50)`) i linię mety (od `(300,-50)` do `(300,50)`). Pamiętaj o funkcjach `.up()`, `.down()`, `.goto(x,y)`.

Każ sędziemu napisać "META" przy mecie (funkcja `.write('napis')`).

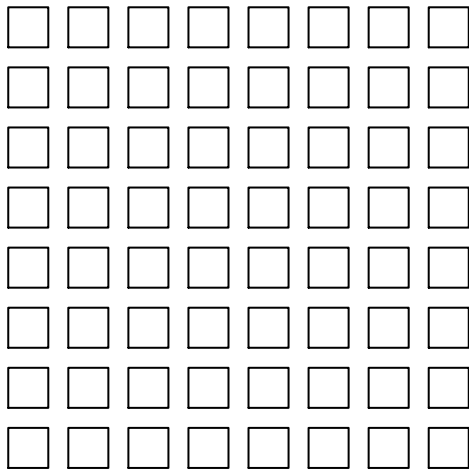
Ustaw żółwie `asia` i `zenek` na linii startu (`.goto`). Następnie przesuwaj każdego z zawodników o losową liczbę pikseli (`randrange(...)`) tak długo, aż któryś z nich nie dojdzie do mety.

Żółw, który ma większą x-ową współrzędną `imiezolwia.xcor()`, wygrywa. Sędzia zmienia kolor na kolor żółwia, który wygrał i wypisuje jego imię.

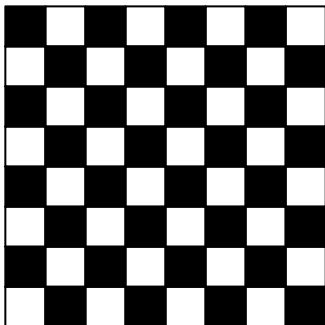
2. Narysuj linię 8 kwadratów o boku 20 w odstępie 10:



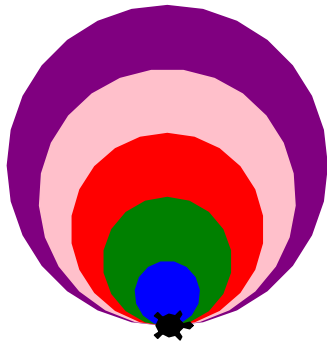
3. Narysuj „szachownicę” 8 na 8 kwadratów o boku 20 w odstępie 10:



4. Narysuj szachownicę 8 na 8 kwadratów o boku 20



5. Narysuj „pawie oko”



## Następujący kod rysuje układ współrzędnych

```
import turtle
turtle.speed(0)
xmax=180
ymax=180
for i in range(2):
    turtle.fd(-xmax)
    turtle.fd(2*xmax)
    turtle.rt(135)
    turtle.fd(10)
    turtle.bk(10)
    turtle.rt(90)
    turtle.fd(10)
    turtle.bk(10)
    turtle.rt(135)
    turtle.home()
    turtle.lt(90)
turtle.up()
```

```
turtle.home()
for x in [-150,-100,-50,50,100,150]:
    turtle.goto(x,0)
    turtle.down()
    turtle.write(x//50)
    turtle.setheading(-90)
    turtle.fd(10)
    turtle.up()
for y in [-150,-100,-50,0,50,100,150]:
    turtle.goto(4,y)
    turtle.write(y//50)
    turtle.goto(0,y)
    turtle.down()
    turtle.setheading(180)
    turtle.fd(10)
    turtle.up()
turtle.ht()
```

6. Narysuj wektory  $[1, 3]$  i  $[-2, -1.5]$

7. Napisz funkcję `wektor(x,y)`, która rysuje wektor o współrzędnych  $(x, y)$