

# Sköldpaddsprogrammering

2024/2025

# Outline

## Kommandon

- Förflyttning

- Andra kommandon

## Interaktivitet

- Här är vi

- Mål

## Input

- Ta emot text

## Datatyper

- Varför olika datatyper

- Tre datatyper

- Växla mellan datatyper

- Ta emot tal

## Interaktivitet

- Vi var här

- Användarinput

## Övningar

- Blad 1

- Interaktivitet x2

# Kommandon

## Förflyttning

Här är en lista med kommandon som flyttar på paddan:

<code>forward(x)</code>	Går x steg framåt
<code>back(x)</code>	Går x steg bakåt
<code>right(x)</code>	Roterar x grader medurs
<code>left(x)</code>	Roterar x grader moturs
<code>setposition((x,y))</code>	Placerar paddan i position (x,y)
<code>setheading(x)</code>	Roterar paddan till x grader

# Kommandon

## Andra kommandon

<code>penup()</code>	Slutar rita
<code>pendown()</code>	Börjar rita
<code>color("färg")</code>	Ändrar färgen
<code>begin_fill()</code>	
<code>end_fill()</code>	
<code>fillcolor('färg')</code>	Ändrar den inre färgen
<code>shape('turtle')</code>	Ändrar formen till en padda
<code>clear()</code>	Tömmer skärmen

# Outline

## Kommandon

- Förflyttning

- Andra kommandon

## Interaktivitet

- Här är vi

- Mål

## Input

- Ta emot text

## Datatyper

- Varför olika datatyper

- Tre datatyper

- Växla mellan datatyper

- Ta emot tal

## Interaktivitet

- Vi var här

- Användarinput

## Övningar

- Blad 1

- Interaktivitet x2

# Interaktivitet

## Statiskt program

Än så länge har vi gjort våra program *statiska* — det gör alltid samma sak när vi kör dem. Om vi vill kunna ändra hur det beter sig under körning så måste vi ta emot *input* från användaren.

Exempel med en stjärna:

```
1 spetsar = 0
2 totala_spetsar = 5
3 while spetsar < totala_spetsar:
4     forward(50)
5     left(180-180/totala_spetsar)
6     spetsar = spetsar+1
```

# Användarinput

Nu vill vi göra något sånt här:

```
1 spetsar = 0
2 totala_spetsar = # ANVÄNDAR INPUT
3 while spetsar < totala_spetsar:
4     forward(50)
5     left(180-180/totala_spetsar)
6     spetsar = spetsar+1
```

# Användarinput

Nu vill vi göra något sånt här:

```
1 spetsar = 0
2 totala_spetsar = # ANVÄNDAR INPUT
3 while spetsar < totala_spetsar:
4     forward(50)
5     left(180-180/totala_spetsar)
6     spetsar = spetsar+1
```

Här kan vi märka att vi kommer långt på att använda variabler istället för att *hårdkoda*.



# Outline

## Kommandon

- Förflyttning

- Andra kommandon

## Interaktivitet

- Här är vi

- Mål

## Input

- Ta emot text

## Datatyper

- Varför olika datatyper

- Tre datatyper

- Växla mellan datatyper

- Ta emot tal

## Interaktivitet

- Vi var här

- Användarinput

## Övningar

- Blad 1

- Interaktivitet x2

# Input

## Ta emot text

Om man vill ta emot text från användaren använder vi oss utav *funktionen* `input()`.

```
1 namn = input("Vad heter du? ") # Tar emot text
2 print("Hej", namn) # Skriver ut
```

# Input

Ta emot text

Om man vill ta emot text från användaren använder vi oss utav *funktionen* `input()`.

```
1 namn = input("Vad heter du? ") # Tar emot text
2 print("Hej", namn) # Skriver ut
```

```
1 Vad heter du? Calle
2 Hej Calle
```

# input-kommandot

```
1 namn = input("Vad heter du? ") # Tar emot text
2 print("Hej", namn) # Skriver ut
```

`input(x)` fungerar som att den skriver ut `x` i *konsolen* och pausar programmet tills användaren trycker på **Enter**. Då sparas texten man skrivit i en variabel. I det här fallet `namn`.

## input-kommandot

```
1 namn = input("Vad heter du? ") # Tar emot text
2 ålder = input("Hur gammal är du? ")
3 print("Hej", namn, ålder) # Skriver ut namn + ålder
```

```
1 Vad heter du? Calle
2 Hur gammal är du? 32
3 Hej Calle 32
```

# Outline

## Kommandon

Förflyttning

Andra kommandon

## Interaktivitet

Här är vi

Mål

## Input

Ta emot text

## Datatyper

Varför olika datatyper

Tre datatyper

Växla mellan datatyper

Ta emot tal

## Interaktivitet

Vi var här

Användarinput

## Övningar

Blad 1

Interaktivitet x2

# Datatyper

## Varför datatyper?

- ▶ Python gör skillnad på talet 3 och siffran 3 eftersom Python inte kan veta om du menar ett tal eller om du faktiskt vill skriva text. Därför måste man hålla koll på vad man vill ge sina variabler för sorts värden.
- ▶ När vi ritade ville vi att alla våra variabler skulle innehålla **tal**. När vi frågar efter någons namn vill vi spara det som text.
- ▶ Text och tal vill man kunna hantera olika. Vi vill kunna räkna med tal och vi vill kunna formatera text.

# Tre datatyper

`int`      Heltal, *integer*  
`float`    Flyttal/decimaltal  
`string`   Text

Sen finns det fler som vi kommer till senare.



# Datatyper

## Växla mellan datatyper

Om man vill ändra ett objekts datatyp skriver man så här:

```
1 a = 3 # En int
2 b = float(a) # Blir en float
3 c = str(a) # Blir en sträng
4 d = int(c) # Blir en int
```

Om man vill kontrollera ett objekts datatyp skriver man så här:

```
1 type(a)
```

## Ta emot tal med input

När man använder `input()` så sparar Python alltid värdet man skickar in som en text-sträng. För att göra om det till ett tal (`int` eller `float`) kan vi göra så här:

```
1 ålder = input("Hur gammal är du? ") # Sparar som str
2 ålder = int(ålder) # Konverterar till int
3 print("Din ålder är", ålder)
```

# Outline

## Kommandon

- Förflyttning

- Andra kommandon

## Interaktivitet

- Här är vi

- Mål

## Input

- Ta emot text

## Datatyper

- Varför olika datatyper

- Tre datatyper

- Växla mellan datatyper

- Ta emot tal

## Interaktivitet

- Vi var här

- Användarinput

## Övningar

- Blad 1

- Interaktivitet x2

# Interaktivitet

Vi var här

```
1 spetsar = 0
2 totala_spetsar = 5
3 while spetsar < totala_spetsar:
4     forward(50)
5     left(180-180/totala_spetsar)
6     spetsar = spetsar+1
```

# Interaktivitet

## Användarinput

```
1 spetsar = 0
2 totala_spetsar = input("Hur många spetsar på stjärnan? ")
3 totala_spetsar = int(totala_spetsar)
4 while spetsar < totala_spetsar:
5     forward(50)
6     left(180-180/totala_spetsar)
7     spetsar = spetsar+1
```

# Outline

## Kommandon

- Förflyttning

- Andra kommandon

## Interaktivitet

- Här är vi

- Mål

## Input

- Ta emot text

## Datatyper

- Varför olika datatyper

- Tre datatyper

- Växla mellan datatyper

- Ta emot tal

## Interaktivitet

- Vi var här

- Användarinput

## Övningar

- Blad 1

- Interaktivitet x2

# Övningar

## Blad 1

1. Fortsätt med `turtle2.py` från förra lektionen.
2. Justera den så att användaren anger antalet hörn på månghörningen.
3. Justera den så att användaren anger antalet spetsar på stjärnan.
4. Skriv ett nytt program som tar emot din ålder skriver hur många månader och hur många dagar du har levt.
5. Utveckla ditt program så att det skriver ut vilket år du är född. (Det är ok om man får fel på ett år med folk födda sent på året)
6. Utveckla ditt program så att det tar emot din grannes ålder och räknar ut er gemensamma ålder.
7. Utveckla programmet så att det tar emot en tredje ålder och skriver ut när du fyller/fyllde så mycket.

# Övningar

## Tangent-input

```
1 listen()  
2 onkey(fram, 'Up')  
3 onkey(bak, 'Down')  
4 onkey(vänster, 'Left')  
5 onkey(höger, 'Right')  
6 mainloop()
```



# Övningar

## Tangent-input

```
1 def fram():  
2     forward(10)  
3 def bak():  
4     back(10)  
5 def vänster():  
6     left(15)  
7 def höger():  
8     right(15)
```

# Övningar

## Blad 2

- ▶ Lägg till att du kan styra sköldpaddan med piltangenterna
- ▶ Lägg till att sköldpaddan kan "hoppa" framåt när man trycker på space