



## Algebra

### ELEV

Det finns många olika programmeringsspråk. Ett av dem är Python, som du ska få bekanta dig med i de här uppgifterna när du ritar med Python och Turtle. Du kommer att få testa, tolka och skriva egen kod samt använda loopar för att rita stjärnor och bestämma stjärnornas placering på skärmen.

Uppgiften bygger vidare på Nivå ETT och uppgifterna i ”Programmering och digital kompetens”, Algebra, Matematik Y. Se även ”Lathund – Python with turtle” för fler tips.

### SYFTE

Syftet med övningen är att du ska

- få bekanta dig med programmeringsspråket Python.
- lära dig att använda en editor för att skriva program.
- få erfarenhet av att rita med Turtle.
- testa, tolka och skriva egen kod.
- förstå varför och hur man använder sig av loopar.
- få förståelse för koordinatsystemets uppbyggnad och förstå hur koordinater för en viss punkt anges.
- få erfarenhet av kommandot `goto(x,y)`.
- få ökad förståelse för vinklar genom att låta Turtle rotera ett visst antal grader med kommandot `left()` respektive `right()`.

### REDOVISNING/BEDÖMNING

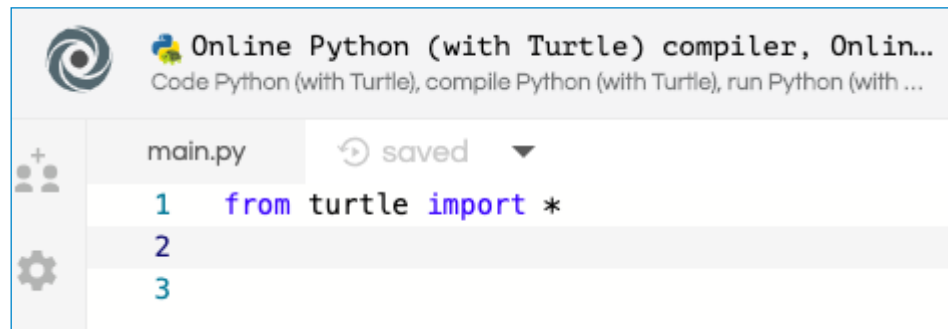
Du lämnar in din kod och dina svar till din lärare.

## DEL 1: Python with turtle, rita stjärnor av kvadrater

1. Börja med att gå in på [repl.it](https://repl.it) och välj ”start coding” och sedan ”Python (with Turtle)” och ”Create Repl”.

För en djupare genomgång av editorn repl.it och dess funktioner, se ”Lathund – Python with Turtle”.

2. Börja med att importera biblioteket Turtle genom att skriva ”`from turtle import *`” i fönstret till vänster.



Observera: Så länge du arbetar med Turtle i Python måste den första raden stå kvar.

3. Byt rad genom att trycka på ”Enter”.  
Skriv sedan in följande kod:

```
for i in range(4):  
    forward(100)  
    right(90)
```

Notera att de två sista raderna ska starta längre in för att visa vad som ingår i loopen ”`for i in range(4):` ”.

Dit kommer du genom att trycka på ”tab”.

4. Klicka på ”Run”. Vad händer?
5. Vad betyder de olika delarna av din kod?
6. Ändra koden till:

```
for i in range(2):  
    for i in range(4):  
        forward(100)  
        right(90)
```

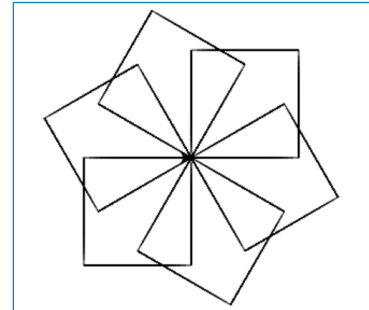
7. Klicka på ”Run”. Vad händer?
8. Vad betyder de olika delarna av din kod?

9. Ändra den första loopen till `for i in range(6):` och lägg till `left(60)` i slutet av den andra loopen.

Håll reda på indragen!

```
for i in range(6):
    for i in range(4):
        forward(100)
        left(90)
    left(60)
```

Om du fått till koden rätt, har du en bild som ser ut så här:



10. För att snygga till ”stjärnan” kan du fylla den med färg genom att lägga till kommandot `begin_fill()` respektive `end_fill()`

Gör det och kör programmet.

```
begin_fill()
for i in range(6):
    for i in range(4):
        forward(100)
        left(90)
    left(60)
end_fill()
```

11. Fortsätt på egen hand och gör förändringar vad gäller storlek, antal uddar på stjärnan och färg.



**TIPS:**

Om du ändrar antalet uddar genom att rita 10 kvadrater istället för 6, behöver du även ändra den sista vridningen turtle gör, från `left(60)` till `left(36)`. Kan du se sambandet?

Tycker du turtle ritar för långsamt? Ändra hastigheten genom att lägga till raden `speed(0)` innan loopen.

```
3 speed(0)
4 begin_fill()
5 for i in range(6):
6     for i in range(4):
```

Färg ändrar du genom att lägga till, till exempel raden `color("red")`

Notera att `color('red')` ger samma resultat. Det spelar ingen roll om man använder dubbla eller enkla citationstecken.

```
3 speed(0)
4 color('red')
5 begin_fill()
6 for i in range(6):
```

## DEL 2: Python with turtle – rita stjärnor som man gör med penna och papper

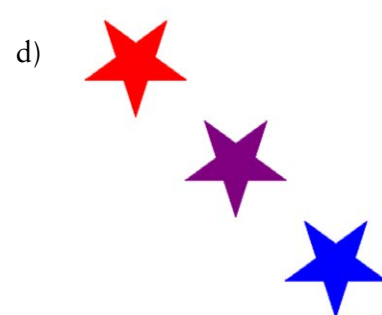
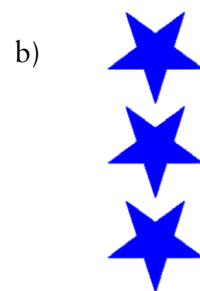
1. Skriv in koden

```
from turtle import *
for i in range(5):
    forward(100)
    left(144)
```

2. Kör programmet. Tolka koden.

3. Rita en ny stjärna och fyll den med färgen blå. Avsluta med `hideturtle()` för att gömma turtle.

4. Använd kommandot `goto(x,y)` samt `penup()` respektive `pendown()` för att åstadkomma följande mönster:



5. Planera och skapa ett eget mönster.