



Algebra

ELEV

Det finns många olika programmeringsspråk. Ett av dem är Python, som du ska få bekanta dig med i den här uppgiften.

I den här övningen kommer du få rita med Turtle. Du kommer att få testa, tolka och skriva egen kod samt lösa problem och skapa konst med hjälp av kod.

SYFTE

Syftet med övningen är att du ska

- få bekanta dig med programmeringsspråket Python3.
- lära dig att använda en editor för att skriva program.
- rita med Turtle.
- testa, tolka och skriva egen kod.
- lösa problem med hjälp av kod.

REDOVISNING/BEDÖMNING

Du lämnar in din kod till din lärare.

DEL 1: Python, en introduktion

För att skriva och köra program i Python 3 på din dator kan du ladda ned och installera Python tillsammans med en så kallad editor. Allt detta finns gratis på webbplatsen www.python.org.

Installera aldrig Python eller andra program om du inte först har tillåtelse från datorns ägare.

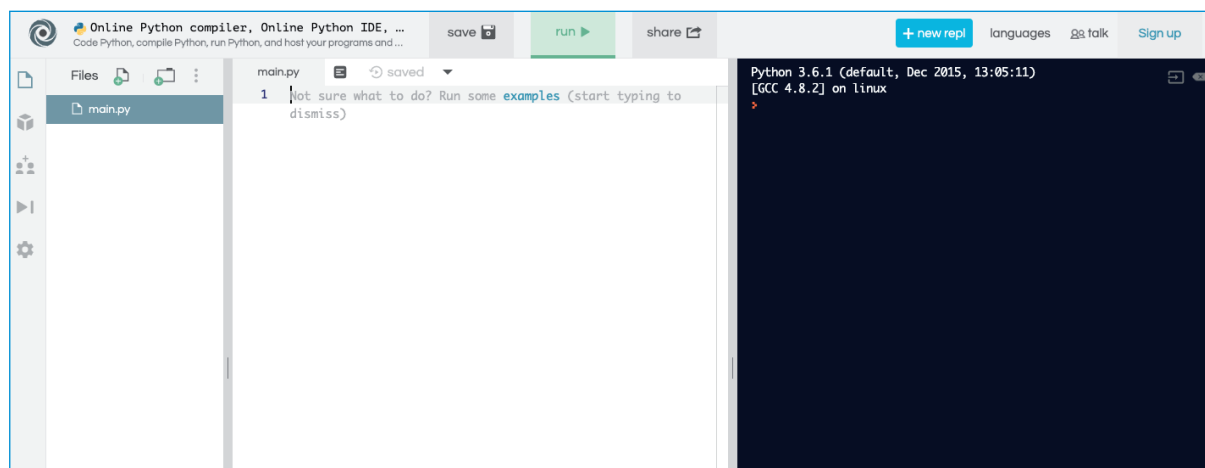
Om man inte vill eller kan installera något på sin dator, finns det webbsidor där man kan arbeta med Python direkt i webbläsaren. Exempel på sådana sidor är repl.it och trinket.io. Du kan använda dem båda genom att skaffa ett konto. Då kan du även spara dina projekt. Fördelen med repl.it är att du kan använda editorn utan konto också.

repl.it/languages/python3

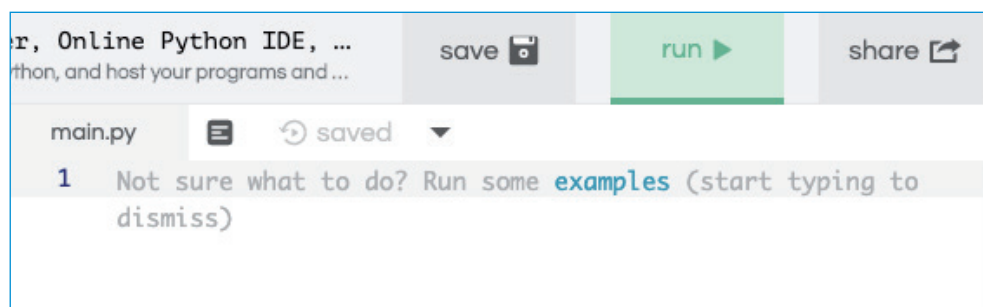
Repl.it:

Gå in på repl.it/languages/python3.

Då öppnas följande fönster:



I mitten finns själva *editorn*. Där skriver du ditt program. När du är färdig klickar du på ”Run”.



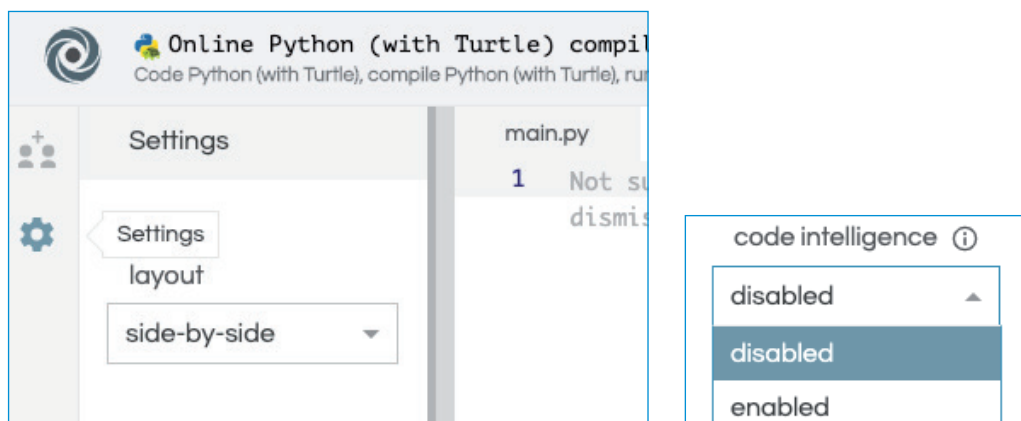
Då körs/visas ditt program i det högra fönstret.

För att kunna använda Turtle behöver du börja med att klicka på ”+new repl” uppe i det högra hörnet.

Välj ”Python (with turtle)” i listan som öppnas och klicka sedan på ”Create repl”.

I editorn finns en inbyggd funktion som innebär att den gissar vad du tänker skriva och ger förslag. När man är ovan programmerare är den funktionen mest förvirrande så vi börjar med att stänga av den:

Välj settings (kugghjulet) och sedan "Code intelligence". Markera "disabled".



DEL 2: Python with turtle, skapa konst

RITA CIRKLAR - UPPGIFT A

1. Skriv in koden:

```
from turtle import *  
circle(100)
```

2. Kör koden genom att klicka på "Run", Vad händer?
3. Vad betyder din kod? Var står värdet på variabeln "100" för, tror du? Testa din hypotes genom att byta ut den mot ett annat värde.
4. Lägg till en rad i din kod:

```
from turtle import *  
circle(100)  
circle(50)
```

5. Kör koden och tolka den. Vad betyder din kod?
6. Ändra koden till:

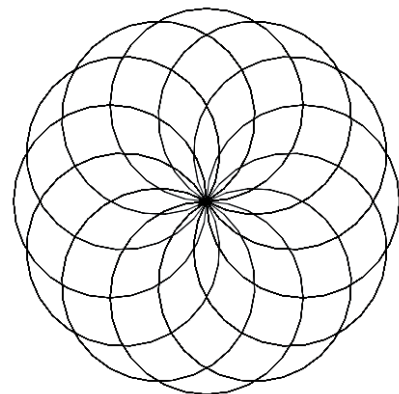
```
from turtle import *  
circle(100)  
right(180)  
circle(50)
```

7. Kör koden och tolka den. Vad betyder din kod?
8. Lägg till en loop:

```
from turtle import *  
for i in range(6):  
    circle(100)  
    right(60)
```

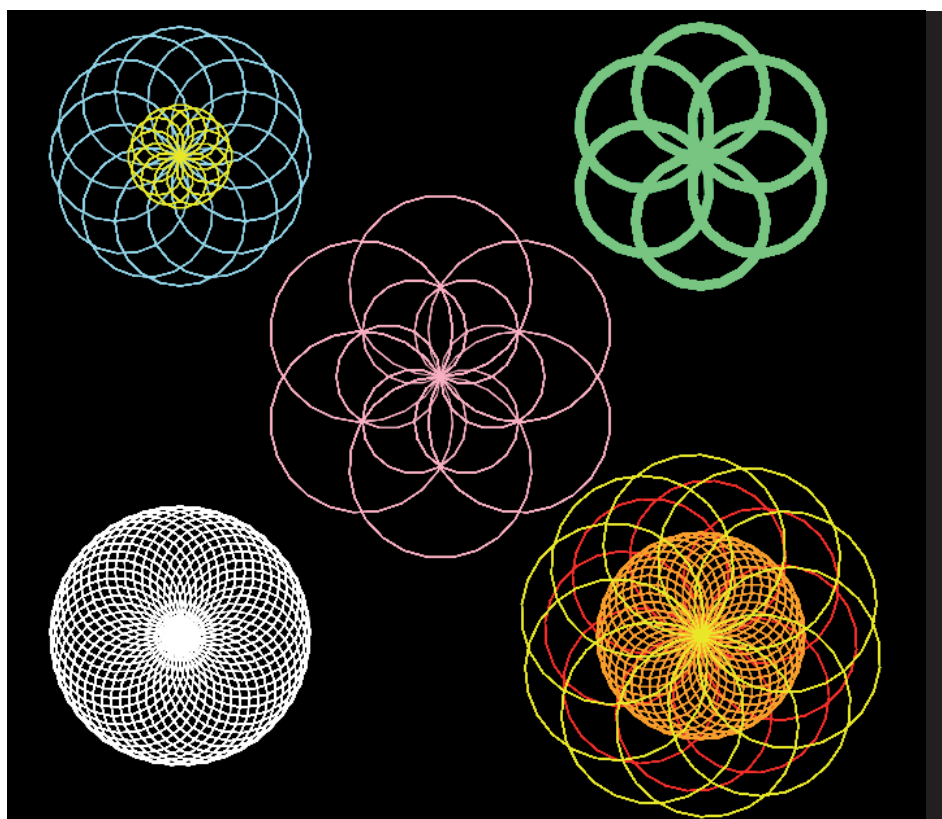
NOTERA att de två sista raderna ska börja en bit in (använd tab-tangenten). Det är för att definiera vad som ingår i loopen.

9. Kör koden och tolka den. Vad betyder din kod?
Vad står "for i in range(6):" för?
Vad står "right(60)" för?
10. Hur ska du ändra din kod, om du istället skulle vilja rita upp den här figuren (med 12 cirklar)?



RITA CIRKLAR PÅ EGEN HAND - UPPGIFT B

1. Skapa mönster på egen hand. Hur gör man till exempel för att rita upp flera "blommor" i samma fönster? Inspiration:



Kod som kan vara användbar:

KOD	FÖRKLARING OCH EXEMPEL
<code>speed ()</code>	Ändrar rit-hastigheten Ex: <code>speed(12)</code> är snabbare än <code>speed(6)</code> <code>speed(0)</code> är det snabbaste och kan vara användbar om man snabbt vill testa sin kod.
<code>Screen () . bgcolor ()</code>	Ändrar färgen på bakgrunden Ex: <code>Screen().bgcolor('black')</code>
<code>pensize ()</code>	Anger tjockleken på pennan. Ex: <code>pensize(3)</code>
<code>forward ()</code>	Får turtle att förflytta sig framåt en viss sträcka (antal pixlar). Ex: <code>forward(100)</code>
<code>backward ()</code>	Får turtle att förflytta sig bakåt en viss sträcka (antal pixlar). Ex: <code>backward(100)</code>
<code>right ()</code>	Får turtle att vrida sig åt höger ett visst gradantal. Ex: <code>right(90)</code>
<code>left ()</code>	Får turtle att vrida sig åt vänster ett visst gradantal. Ex: <code>left(45)</code>
<code>penup ()</code>	Lyfter "pennan"
<code>pendown ()</code>	Sätter ned pennan

<code>color()</code>	Ändrar färgen på turtles penna Ex: <code>color("red")</code> För exempel på olika färger man kan använda: Sök på "Python colors" så får du upp exempel på färgkartor. Färgerna kan även skrivas som färgkoder, RGB, i formen (1,2,3)
<code>goto()</code>	Förflyttar turtle till en viss position på skärmen. Anges i x-led och y-led. Turtle startar alltid i Origo (0,0) Ex: <code>goto(100, -100)</code>
<code>hideturtle()</code>	Gömmer turtle
<code>for i in range(4): forward(100) right(90)</code>	Loop som skapar en kvadrat med sidan 100 pixlar
<code>begin_fill()</code>	Markerar starten för en polygon (månghörning) man vill fylla med färg. Ex: <code>begin_fill()</code>
<code>end_fill()</code>	Markerar slutet för en polygon (månghörning) man vill fylla med färg. Ex: <code>end_fill()</code>

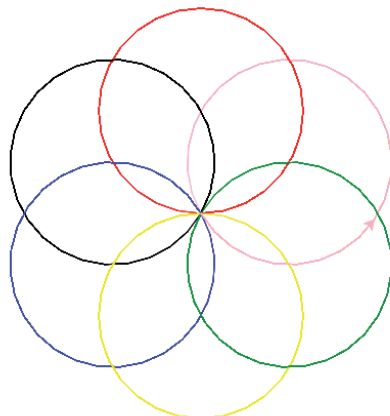
2. Visa ditt resultat för en kompis. Jämför kod, inspireras av varandra och skapa något nytt tillsammans.

RITA CIRKLAR MED FÄRGBYTEN - UPPGIFT C

1. Öppna ett nytt fönster och skriv in koden:

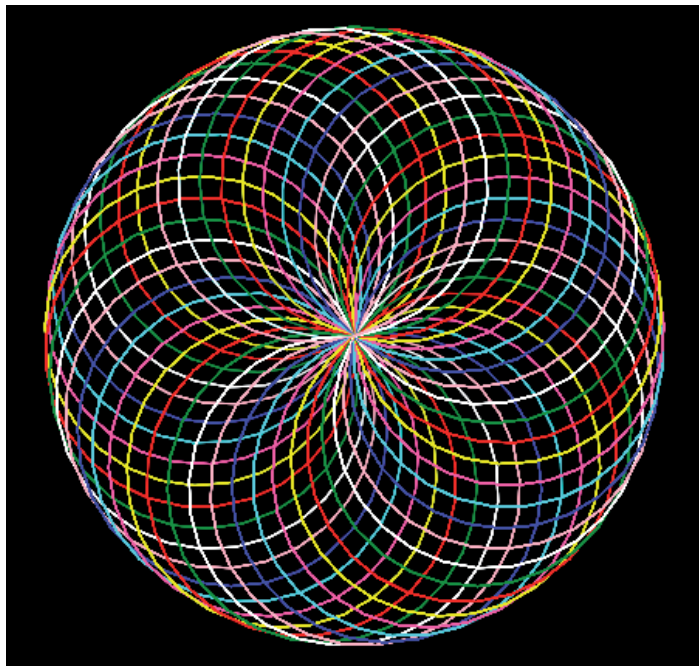
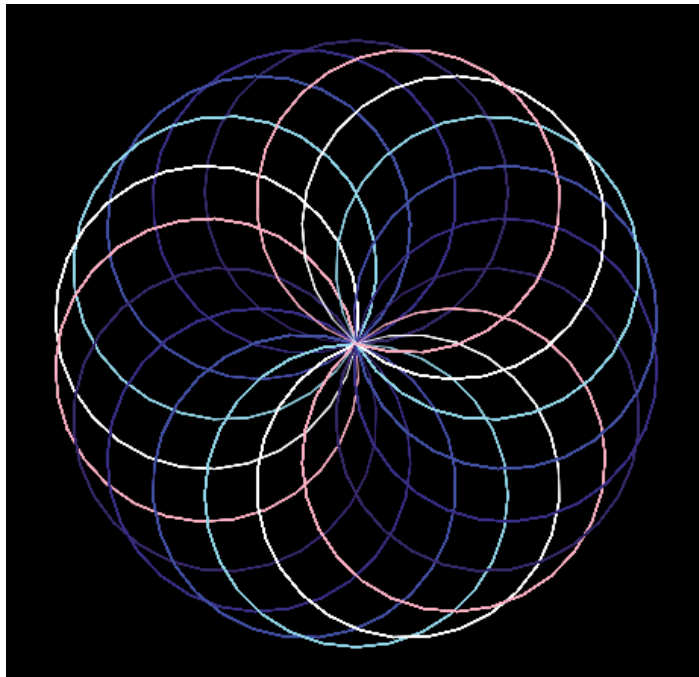
```
from turtle import *
speed(12)
for colours in ("red", "black", "blue"):
    color(colours)
    circle(100)
    left(120)
```

2. Testa din kod genom att klicka på "Run". Vad händer?
3. Tolka din kod. Vad betyder de olika delarna?
4. Ändra koden så cirklar i sex olika färger ritas upp.



UTMANING - UPPGIFT D

1. Arbeta vidare med din kod och försök rita en bild som liknar någon av de här:



2. Finns det flera olika sätt att lösa problemet på? Diskutera och jämför med en kompis.