



## Algebra

### LÄRARE

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

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

### NSYFTE

Syftet med övningen är att eleven ska

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

### TIDSÅTGÅNG

En lektion à 60 min.

### KOSTNAD

Ingen kostnad

### UTRUSTNING

Datorer eller lärplattor och webbsidan Repl.it <https://repl.it>

### REDOVISNING

Eleven lämnar in sin kod och sina konstverk till läraren.

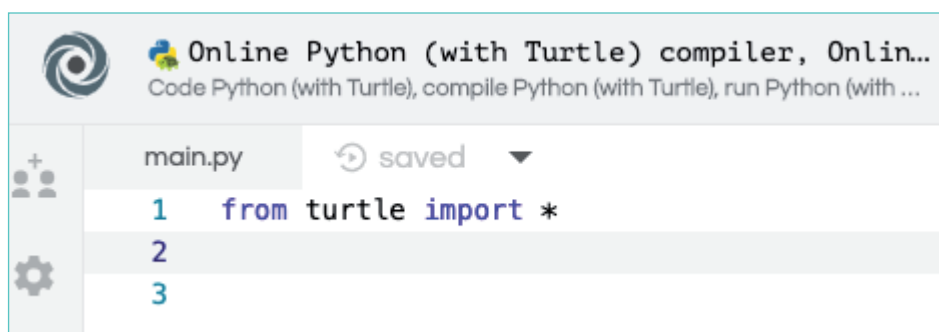
## FALLGROPAR

Testa gärna övningarna själv först så får du en bild av vilka problem eleverna kan stöta på.

Vanliga fel är till exempel att man stavat fel, blandat ihop kommatecken och punkt eller att man glömt citattecken (citationstecken), kolon eller parenteser. Men måste också skriva programmet i ”rätt ordning”, alltså i den ordning programmet ska utföras. Datorn kan inte tänka själv och avgöra vilken ordning som är logisk.

Håll reda på indragen! All kod som ingår i en loop måste starta en bit in. Glöm inte kolon

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



The screenshot shows the top part of an online Python compiler. The title bar reads "Online Python (with Turtle) compiler, Onlin...". Below the title bar, there is a file name "main.py" and a "saved" status. The code editor contains the following code:

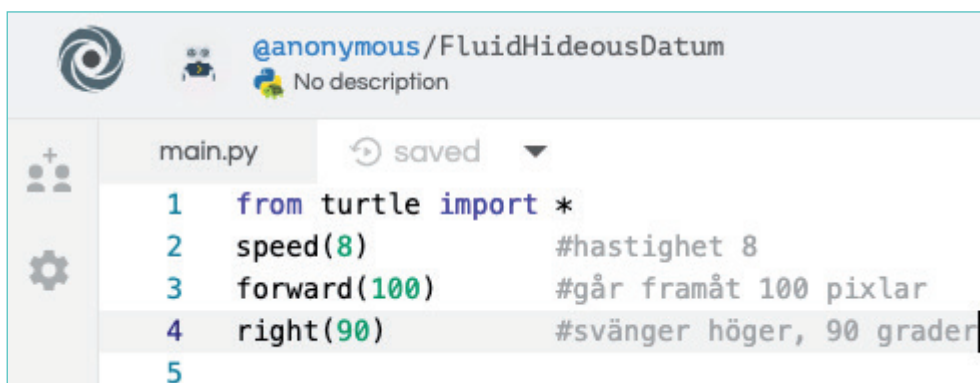
```
1 from turtle import *
2
3
```

Det är inte alltid lätt att tolka de felmeddelanden som dyker upp under ”console”. Alltid brukar det vara någon elev som ”knäcker koden”. Låt eleverna hjälpa varandra, när de försökt själva först.

## PEDAGOGISKA TIPS

Man kan skriva in anteckningar i sin kod, utan att det påverkar själva programmet. Det kan man göra genom att använda #. Det är användbart om eleven ska lämna in ett program som hen gjort och vill visa att hen förstått de olika delarna.

Då kan det ut så här:

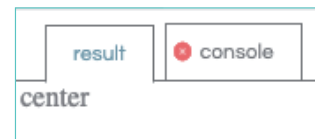


The screenshot shows the top part of an online Python compiler. The title bar reads "@anonymous/FluidHideousDatum" and "No description". Below the title bar, there is a file name "main.py" and a "saved" status. The code editor contains the following code:

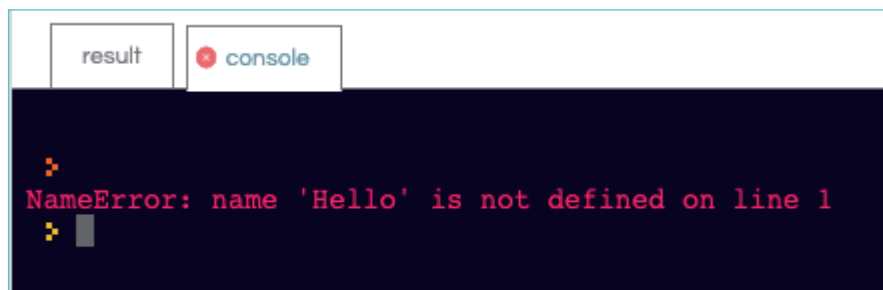
```
1 from turtle import *
2 speed(8) #hastighet 8
3 forward(100) #går framåt 100 pixlar
4 right(90) #svänger höger, 90 grader
5
```

Eftersom det inte går att spara kod om man inte har något konto, kan eleven kopiera sin kod (markera text, ctrl/cmd C) och sedan klistra in (ctrl/cmd V) den i ett separat dokument som hen lämnar in till läraren.

Ibland när man trycker på "Run" så händer ingenting. Istället lyser en röd markering vid fliken "console".



Om man klickar på "console" får man fram ett felmeddelande som visar i vilken rad felet uppstått. Ofta är det felstavning, kolon som saknas eller att man glömt citationstecken.



På hemsidan Stackoverflow kan man få viss hjälp: <https://stackoverflow.com/>

Samarbeta med ämnet bild och skapa konst att smycka klassrummet och korridoren med.

I Bilaga 1 finns tips på användbar kod när man arbetar med Python och Turtle.

### FÖRMÅGOR

- formulera och lösa problem med hjälp av matematik samt värdera valda strategier och metoder,
- använda och analysera matematiska begrepp och samband mellan begrepp,
- välja och använda lämpliga matematiska metoder för att göra beräkningar och lösa rutinuppgifter,
- använda matematikens uttrycksformer för att samtala om, argumentera och redogöra för frågeställningar, beräkningar och slutsatser.

### CENTRALT INNEHÅLL

Hur algoritmer kan skapas och användas vid programmering. Programmering i olika programmeringsmiljöer.

Hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning.

Avbildning och konstruktion av geometriska objekt, såväl med som utan digitala verktyg.

## KUNSKAPSKRAV

	E	C	A
<b>Problem-lösning 1, P<sub>1</sub></b>	Eleven kan lösa olika problem i bekanta situationer på ett i <b>huvudsak</b> fungerande sätt genom att välja och använda strategier och metoder med <b>viss</b> anpassning till problemets karaktär samt <b>bidra till att formulera</b> enkla matematiska modeller som kan tillämpas i sammanhanget.	Eleven kan lösa olika problem i bekanta situationer på ett <b>relativt väl</b> fungerande sätt genom att välja och använda strategier och metoder med <b>förhållandevis god</b> anpassning till problemets karaktär samt <b>formulera</b> enkla matematiska modeller som <b>efter någon bearbetning</b> kan tillämpas i sammanhanget.	Eleven kan lösa olika problem i bekanta situationer på ett <b>väl</b> fungerande sätt genom att välja och använda strategier och metoder med <b>god</b> anpassning till problemets karaktär samt <b>formulera</b> enkla matematiska modeller som kan tillämpas i sammanhanget.
<b>Problem-lösning 2, P<sub>2</sub></b>	Eleven för <b>enkla och till viss del</b> underbyggda resonemang om val av tillvägagångssätt och om resultatens rimlighet i förhållande till problemsituationen samt kan <b>bidra till</b> att ge <b>något</b> förslag på alternativt tillvägagångssätt.	Eleven för <b>utvecklade och relativt väl</b> underbyggda resonemang om tillvägagångssätt och om resultatens rimlighet i förhållande till problemsituationen samt kan ge <b>något</b> förslag på alternativt tillvägagångssätt.	Eleven för <b>välutvecklade</b> och <b>väl</b> underbyggda resonemang om tillvägagångssätt och om resultatens rimlighet i förhållande till problemsituationen samt kan ge <b>förslag</b> på alternativa tillvägagångssätt.
<b>Begrepp 3 B<sub>3</sub></b>	I beskrivningar av matematiska begrepp kan eleven växla mellan olika uttrycksformer samt föra <b>enkla</b> resonemang kring hur begreppen relaterar till varandra.	I beskrivningar av matematiska begrepp kan eleven växla mellan olika uttrycksformer samt föra <b>utvecklade</b> resonemang kring hur begreppen relaterar till varandra.	I beskrivningar av matematiska begrepp kan eleven växla mellan olika uttrycksformer samt föra <b>välutvecklade</b> resonemang kring hur begreppen relaterar till varandra.
<b>Metod</b>	Eleven kan välja och använda <b>i huvudsak fungerande</b> matematiska metoder med <b>viss</b> anpassning till sammanhanget för att göra beräkningar och lösa rutinuppgifter inom aritmetik, algebra, geometri, sannolikhet, statistik samt samband och förändring med <b>tillfredställande</b> resultat.	Eleven kan välja och använda <b>ändamålsenliga</b> matematiska metoder med <b>relativt god</b> anpassning till sammanhanget för att göra beräkningar och lösa rutinuppgifter inom aritmetik, algebra, geometri, sannolikhet, statistik samt samband och förändring med <b>gott</b> resultat.	Eleven kan välja och använda <b>ändamålsenliga och effektiva</b> matematiska metoder med <b>god</b> anpassning till sammanhanget för att göra beräkningar och lösa rutinuppgifter inom aritmetik, algebra, geometri, sannolikhet, statistik samt samband och förändring med <b>mycket gott</b> resultat.
<b>Kommunikation</b>	Eleven kan redogöra för och samtala om tillvägagångssätt på ett i <b>huvudsak fungerande</b> sätt och använder då symboler, algebraiska uttryck, formler, grafer, funktioner och andra matematiska uttrycksformer med <b>viss</b> anpassning till syfte och sammanhang.	Eleven kan redogöra för och samtala om tillvägagångssätt på ett <b>ändamålsenligt</b> sätt och använder då symboler, algebraiska uttryck, formler, grafer, funktioner och andra matematiska uttrycksformer med <b>förhållandevis god</b> anpassning till syfte och sammanhang.	Eleven kan redogöra för och samtala om tillvägagångssätt på ett <b>ändamålsenligt och effektivt</b> sätt och använder då symboler, algebraiska uttryck, formler, grafer, funktioner och andra matematiska uttrycksformer med <b>god</b> anpassning till syfte och sammanhang.

## BEDÖMNING

För- måga	Nivå	Kommentar för bedömning	Testas i uppgift
<b>P1</b>			
	<b>E</b>	Eleven löser enklare problem med hjälp av kod, genom att t ex testa sig fram och kombinera olika delar av koden för att mönster och "blommor" ska ritas upp.	<i>del 2: B</i>
	<b>C</b>	Eleven löser problem med hjälp av kod och har en tanke om vad som ska ritas upp och hur koden då borde formuleras. Eleven testar koden och försöker omformulera den vid behov.	<i>del 2: B och D</i>
	<b>A</b>	Eleven visar säkerhet i sitt problemlösande genom att ha en tydlig och genomtänkt plan för hur koden ska formuleras innan hen testar den.	<i>del 2: B, C och D</i>
<b>P2</b>			
	<b>E</b>	Eleven kan diskutera tillvägagångssätt och göra enklare felsökningar.	<i>del 2: B</i>
	<b>C</b>	Eleven diskuterar och analyserar tillvägagångssätt samt gör vissa felsökningar.	<i>del 2: B, C och D</i>
	<b>A</b>	Eleven är systematisk i sitt felsökande och kan ge förslag på alternativa tillvägagångssätt.	<i>del 2: B, C och D</i>
<b>B3</b>			
	<b>E</b>	Eleven kan beskriva några delar av sin kod.	<i>del 2: hela uppgiften</i>
	<b>C</b>	Eleven kan förklara stora delar av sin kod.	<i>del 2: hela uppgiften</i>
	<b>A</b>	Eleven kan förklara de olika delarna av sin kod och beskriver ingående vad de olika delarna innebär och vad skillnaden är mellan dem.	<i>del 2: hela uppgiften</i>
<b>M</b>			
	<b>E</b>	Eleven kan skriva given kod med och utan loopar samt testa den.	<i>del 2: A1-2, A4-8, C1-2</i>
	<b>C</b>	Eleven kan ändra koden i loopen för att rita upp tolv cirklar.	<i>del 2: A10</i>
		Eleven kan ändra koden för att sex cirklar i olika färger ska ritas upp.	<i>del 2: C4</i>
	<b>A</b>	Eleven kan skriva egen kod och med loopar och testa den.	<i>del 2: D</i>
<b>K</b>			
	<b>E</b>	Eleven kan presentera sin kod.	<i>del 2: A och B</i>
	<b>C</b>	Eleven kan presentera och beskriva sin kod.	<i>del 2: B, C och D</i>
	<b>A</b>	Eleven kan presentera en fullständig och korrekt kod.	<i>del 2: B, C och D</i>

## BILAGA 1: Användbar kod

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 <code>(1,2,3)</code>
<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
<pre>for i in range(4):     forward(100)     right(90)</pre>	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>