

# .se

Måns Jonasson

# Barnhack!

– kom igång med Scratch del I

VÄLKOMMEN TILL KOM IGÅNG MED SCRATCH!	3
<i>Om den här kursen</i>	3
<i>Vad är Scratch?</i>	3
GRUNDERNA I SCRATCH – VAD ÄR VAD?	3
BLOCK – SCRIPT I OLIKA KATEGORIER	5
<i>Rörelse</i>	5
<i>Utseende</i>	6
<i>Ljud</i>	6
<i>Penna</i>	6
<i>Data</i>	6
<i>Händelser</i>	6
<i>Kontroll</i>	6
<i>Känna av</i>	6
<i>Operatorer</i>	6
<i>Fler block</i>	6
DITT FÖRSTA PROGRAM	7
<i>Ditt första script – låt katten springa</i>	8
<i>Loopar – så gör du saker flera gånger</i>	10
<i>Flera script till samma sprite</i>	13
<i>Ljud – låt katten jama</i>	14
<i>Variabler – håll något i minnet</i>	14
<i>Att räkna upp eller ner variabler</i>	17
<i>Scenen kan också ha script</i>	17
<i>Initiering – hur ska du ha det när du börjar?</i>	19
LJUD – ALLT BLIR ROLIGARE MED MUSIK	20
<i>Ladda in ljud från det medföljande biblioteket</i>	21
VILLKOR – KÖR BARA KODEN OM NÅGOT STÄMMER	22
BAKGRUNDER – SCENENS DEKOR	26

## Barnhack! Kom igång med programmering

Det här dokumentet om att komma igång med Scratch är extramaterial som hör ihop med Internetguiden "Barnhack! Kom igång med programmering" av Anders Thoesson. Hela boken, och mer extramaterial, finns att ladda ned kostnadsfritt här: [www.iis.se/guider](http://www.iis.se/guider)



## Välkommen till Kom igång med Scratch!

Målet med den här kursen är att du ska få en första inblick i hur det är att programmera egna spel och andra program. Idag är nästan alla barn i Sverige vana vid att använda datorer, surfplattor och mobiltelefoner. De kan installera appar, spela spel, skriva skolarbeten, använda webbläsare och söka på nätet. Det är dock inte så många barn som fått en inblick i hur det är att själv instruera sina smarta prylar och få dem att göra vad som helst.

När hemdatorerna blev vanliga på 1980-talet uppstod en generation av ”sovrumsprogrammerare” – vanliga barn och ungdomar som lärde sig att programmera sina datorer, dels eftersom det inte fanns samma utbud av färdiga program och spel som idag, men antagligen främst för att alla hemdatorer kom förprogrammerade med BASIC – ett enkelt programmeringsspråk som lockade många till att börja experimentera med programmering på ett enkelt sätt.

Dagens PC-datorer har oftast inte med någon programmeringsmiljö installerad från början och många av enheterna som våra barn använder är av säkerhetsskäl slutna plattformar, vilket gör det närmast omöjligt för en vanlig användare att börja skapa egna program.

### Om den här kursen

Den här instruktionsboken är skriven för den som är absolut nybörjare när det kommer till programmering. I skrivande stund är Scratch 2.0 i betaversion, vilket tyvärr gör alla delar av Scratch inte riktigt funkar som de ska och en hel del är inte översatt till svenska än. Instruktionsboken kommer att uppdateras löpande för att anpassas till nya funktioner i Scratch. Den här instruktionsboken är ett så kallat XL-material som hör ihop med Internetguiden ”Barnhack! – programmera med barn” som du kan ladda ned och läsa från: [www.iis.se/guider](http://www.iis.se/guider)

### Vad är Scratch?

Scratch är en mjukvara framtagen av MIT (Massachusetts Institute of Technology) för att introducera programmering som koncept för barn och ungdomar. Eftersom det är lätt att komma igång med Scratch är det dock också många vuxna som gillar att göra egna program med hjälp av Scratch.

Den som börjar leka med Scratch lär sig snabbt grunderna i programmering som koncept. Du använder dig av objekt, variabler, loopar, villkor och mycket annat som blir en solid grund att bygga vidare på om du blir sugen på att fortsätta din

programmeringskarriär efter att ha börjat med Scratch. Det finns stora mängder färdiga spel och program till Scratch att titta på, lära sig av och börja ändra i.

Barn som använder Scratch blir snabbt kompetenta "mini-programmerare" och kan redan efter en stund göra egna spel och program. Men kanske allra viktigast är att barnen lär sig att datorerna tar sina instruktioner från människor, och att alla som vill kan lära sig hur.

## Grunderna i Scratch – vad är vad?

Scratch började som ett Java-program, men inför version 2.0 byggde man om hela programmet till Adobe Flash. Den senaste versionen används alltså i din webbläsaren genom att surfa till:

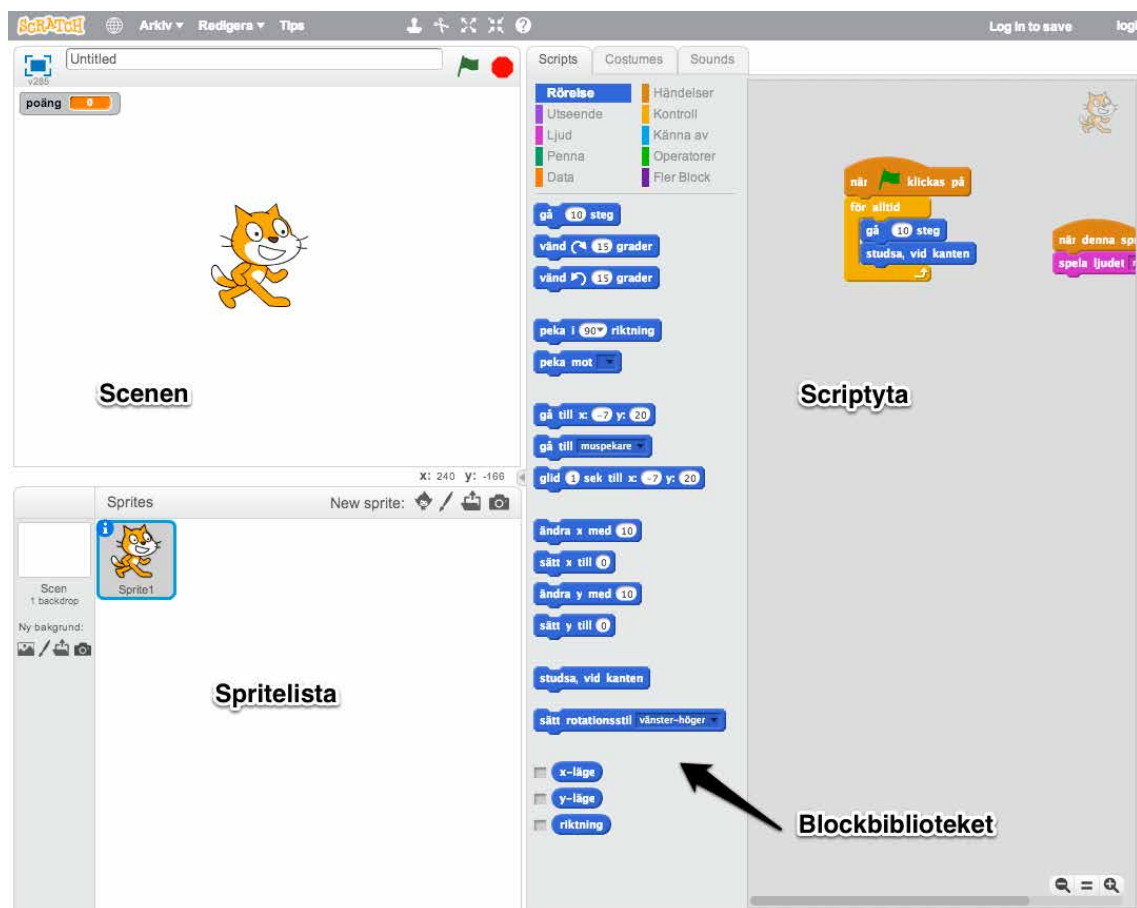
<http://scratch.mit.edu>

Det du möts av där är ett antal färdiga Scratch-program som du kan titta på för att se vad som går att göra med Scratch, men det du är intresserad av är så klart att börja skapa ditt eget program. Klicka därför på "Skapa" högst upp i menyn (Om webbplatsen är på engelska när du kommer in står det "Create" där istället).

Det som händer nu är att editorn i Scratch laddas över hela webbläsarens yta. Du går igenom de olika delarna av editorn först av allt.

Scratch finns översatt till en mängd olika språk. Tyvärr kan det smyga sig in lite engelska trots det, eftersom version 2.0 fortfarande är i teststadiet. Börja med att växla språk i editorn till svenska genom att klicka på "globen" i menyraden och välja svenska där.

Miljön i Scratch är indelad i några olika områden. Överst till vänster har du en yta som kallas "scen". Under den visas de "sprites" som du har att arbeta med. Längst till höger visas de "script" som är kopplade till en vald sprite, och i mitten har du ett bibliotek med de olika script som finns att använda, indelade i avdelningar med olika typer av "block".



Editorn i Scratch 2.0

Man kan tänka på Scratch som en teater. Teatern har en scen där allting utspelar sig. Scenen kan byta dekor (bakgrund) och på scenen ställer sig en eller flera skådespelare (sprites) för att agera. Skådespelarna agerar enligt sitt manus (script).

I Scratch är det du som är regissör, rollbesättare, manusförfattare, scenograf och kostymör! Du bestämmer vilken pjäs som ska spelas, vilka som ska vara med i den, vad skådespelarna ska säga, hur scenen ska se ut och vad skådespelarna ska ha på sig.

För att datorn ska kunna utföra dina instruktioner och programmet ska köras som du tänkt dig behöver du koppla ihop dina sprites med olika block till ett script och placera dem på scenen. Du kan se vilka script som är kopplade till en sprite genom att antingen klicka på spriten i din spritelista eller genom att dubbelklicka

på en sprite som är placerad på scenen. När du först börjar använda en sprite har den inga script kopplade till sig, så det ser tomt ut i ytan till höger. För att instruera en sprite drar du block från spritebiblioteket till scriptytan till höger, vilket kopplar ett block till just den spriten. Mer om det senare, för nu är det dags att sluta snacka och börja göra lite program!

Nu kastar du dig rakt in i Scratch, och du kanske börjar undra över några av de saker som som går igenom i det här kapitlet. Bli inte orolig för det, utan försök ändå att bara följa kapitlet från början till slut. Du kommer att lära dig några av de grundläggande sakerna kring hur Scratch fungerar, och du kommer att gå in på detaljerna senare.

## Block – script i olika kategorier

Överst i mitten, ovanför blockbiblioteket, ser du tre flikar – ”Script”, ”Klädslar” och ”Ljud”. Under dessa ligger tio kategorirubriker med olika färgkoder.



*De tio olika scriptkategorierna i Scratch*

Du kommer nu att titta lite närmare på de olika kategorierna av scriptblock, men du börjar med att titta på hur själva blocken ser ut:



*Blocket som rör en sprite ett antal steg*

Förutom färgen på blocket ovan så ser du att det har en liten inbuktning på ovansidan och en utbuktning på undersidan. Det här är Scratch sätt att visa dig hur de olika blocken kan sättas ihop, precis som pusselbitar. Just det här blocket kan du alltså sätta ihop med ett block från ovansidan och ett annat block från undersidan. På så sätt kan du styra i vilken ordning instruktionerna ska utföras, för Scratch läser ett script uppifrån och ner. Det block du sätter under detta kommer att läsas efter att Scratch gjort klart det som ditt block ska göra.

### RÖRELSE

Nu kan du titta på kategorierna. Först ligger ”rörelse”. Här hittar du de block som används för att flytta en sprite på scenen. Det allra första blocket, som du tittade på nyss, flyttar en sprite 10 steg framåt. Du ser att siffran ”10” ligger i en

vit oval. Det betyder att du kan klicka där och ändra siffran från 10 till vad du vill – och Scratch kommer att flytta denna sprite så många steg som du angett när blocket körs. Här finns en mängd olika block men du tittar på fler av dem lite längre fram.

#### UTSEENDE

Under ”utseende” ligger lila block som styr hur en sprite eller scenen ska se ut. Här finns till exempel möjligheten att gömma sprites ett tag, eller att ändra bakgrund på scenen.

#### LJUD

Under ”ljud” ligger förstås de block som låter Scratch spela upp ljud av olika slag. Om en sprite ska skrika ”Aj” när den träffas av något är det med hjälp av de rosa ljudblocken du gör det.

#### PENNA

De gröna ”penna”-blocken låter oss rita på scenen. Du kan göra ett eget ritprogram i Scratch om du vill och de här blocken hjälper till.

#### DATA

Under ”data” är det nästan helt tomt när du börjar med ett nytt projekt i Scratch. Men här finns ett par knappar, nämligen ”Skapa en variabel” och ”Skapa en lista” som du kommer att använda senare för att skapa små virtuella lådor att spara saker i. Mer om det senare.

#### HÄNDELSE

Den här kategorin är en av de viktigaste, för det är genom att koppla händelseblock till dina sprites som du kan få dem att reagera när du till exempel klickar på dem med muspekaren eller trycker ner en tangent på tangentbordet.

#### KONTROLL

De gula kontrollblocken är också väldigt viktiga. Här kan du skapa loopar, villkor och annat som låter dig styra hur och när dina sprites ska agera.

#### KÄNNA AV

De här ljusblå blocken har flera olika former, vilket du lär dig mer om lite senare. Här finns block som låter en sprite reagera om den till exempel rör vid en annan sprite, vilket är väldigt användbart till exempel när du gör spel och vill att något ska hända om två sprites krockar med varandra.



**OPERATORER**

Här ligger allt som har med matte att göra, tillsammans med en hel del andra gröna block som hanterar beräkningar och logik. Ibland behöver du till exempel kunna jämföra två tal med varandra, och då är det med operatorer du kan göra det.

**FLER BLOCK**

Den här kategorin är överkurs för tillfället. I Scratch 2.0 kan man göra helt egna block, men funktionen är inte riktigt färdig än, så du hoppar över den till en annan gång.

## Ditt första program – Du säger hej till katten Scratch



*Katten Scratch*

Katten är bara en av väldigt många olika figurer som följer med programmet, men eftersom han står där mitt på scenen redan när editorn öppnas kan du ju börja med att se till att han har något att göra.

I den här övningen kommer du använda ett par olika block för att få katten att springa över skärmen och dessutom jama när du klickar på honom med muspekaren. Kanske inte världens roligaste program, men du kommer att lära dig ett par grundläggande saker innan du börjar lära dig mer om de olika kategorierna av scriptblock.



*Infoknappen finns på varje sprite*

Varje sprite kan finnas på två ställen när den laddats in i Scratch. Katten som är laddad från början står dels mitt på scenen, dels återfinns han längre ner på skärmen, i spritelistan där alla dina laddade sprites alltid ligger. I övre vänstra hörnet av varje sprite i listan finns en blå rund info-knapp. Här klickar du för att ställa olika egenskaper för en sprite, så klicka på den nu för att se alternativen för katten, eller "Sprites" som den heter just nu.

*Inställningar för sprite*

I rutan som visas nu kan du se några olika detaljer för just den sprite du klickade på, en instans av katten Scratch. Först och främst går det att döpa en sprite till något som går lättare att känna igen, om du vill. Under namnet finns några värden: x, y och riktning. X och y är spritens position på scenen, och från allra första början när katten står mitt på scenen är dess position X:0 och Y:0. Du kommer att lära dig mer om positioner i koordinater senare, men nästa värde, ”riktning”, är viktigt redan nu.

Riktningen säger åt vilket håll en sprite pekar. Istället för att säga att en sprite pekar åt exempelvis höger eller neråt används grader. Noll grader är rakt uppåt, så 90° betyder att den här spritens riktning just nu är rakt åt höger på scenen, vilket också visas av den lilla ratten till höger om siffran. Prova att klicka på ratten och dra med musen åt olika håll medan du håller ner musknappen. Du ser dels att siffran ändras mellan noll och 360 grader, men du ser också att den katt som visas på scenen snurrar runt för att följa med din inställning på ratten. Ställ tillbaka riktningen till 90 grader just nu, för du vill att katten ska börja gå åt höger när han sätter igång.

På nästa rad finns något som heter ”rotationsstil”. Det här berättar för editorn hur spriten ska tillåtas att rotera. Som standard kan en sprite rotera i alla riktningar, vilket du såg alldeles nyss när du drog i riktningratten. Här finns tre inställningar: ”full rotation”, ”vänster-höger” och ”ingen rotation” som bara ser ut som en prick. Klicka på pricken för att bestämma att spriten inte får rotera alls. Testa nu igen att dra i riktningratten ovanför och du ser att även om riktningssiffran ändras så står katten still och snurrar inte alls. Det är för att du ställt in denna sprite på att inte roteras alls. Ställ in ”rotationsstil” på ”vänster-höger” nu genom att klicka på mittenalternativet som ser ut som ett rakt streck med två pilspetsar och se till att riktningen ovanför är på 90 grader igen om du ställde om den för att testa nyss. Nu har du ställt in din sprite på att bara få vända sig rakt åt höger eller vänster, inga andra håll.

Den sista rutan i inställningspanelen för vår sprite heter ”kan dras på scenen” och den bestämmer huruvida en sprite kan flyttas med hjälp av musen när programmet körs senare. Du låter den vara urkryssad för katten, du vill ju flytta honom med hjälp av script, inte med musen.

För att komma tillbaka till spritelistan, klicka på den lilla runda blå knappen med en vänsterpil precis till vänster om katten (se bilden ovan).

### Ditt första script – låt katten springa

För att katten ska kunna agera måste du ge den ett manus – ett script. När du klickar på en sprite i spritelistan så visas just den spritens script i den stora grå ytan till höger på skärmen som du kallar "spritescript". Där är tomt just nu eftersom katten inte fått några block kopplade till sig än, men du ser en liten utsuddad miniversion av spriten längst upp till höger för att påminna oss om vilken sprite du jobbar med för tillfället.

Nu ska du få katten att springa över skärmen! För att göra detta behöver du koppla ett händelseblock och ett rörelseblock till katten.

Om du tittar till vänster i editorn, i det övre högra hörnet av scenen, så ser du en grön flagga och en röd cirkel där. Det här är Scratch start- och stoppknappar.



Scratch start- och stoppknappar

Klicka nu på "Händelser" överst i blockbiblioteket för att lista alla orange block som har med händelser att göra. För att få saker att börja hända kan du använda det allra första händelseblocket: "När [grön flagga] klickas på".



När [grön flagga] klickas på

Det här blocket är ett av startblocken i Scratch, för som du ser finns det ingen inbuktning ovanpå, så det går inte att haka det här blocket på undersidan av andra block. Däremot finns det en utbuktning på undersidan, vilket är en inbjudan till att lägga till andra block efter.

När du kopplar det här händelseblocket till en sprite så gör du det möjligt att låta spriten agera så fort du klickar på knappen med den gröna flaggan på ovanför scenen. För att koppla blocket till din kattsprite drar du helt enkelt blocket från biblioteket in i spritescript-ytan till höger. Du kan släppa det var som helst, men just nu kan du lägga det någonstans i övre delen av scriptet.

Nu har du berättat för Scratch att du vill att katten, "Sprite", ska agera när du klickar på startknappen. Men om du klickar på startknappen nu händer ingenting, för du har ju inte berättat vad som ska hända när du klickar på den. För att se till att något händer måste du välja ett annat block och haka på det på undersidan av ditt händelseblock.

Klicka på kategorin "Rörelse" i blockbiblioteket för att lista alla blå rörelseblock. Det allra första blocket står det "Gå (10) steg" på.



Rörelseblock: gå (10) steg

Varje gång det här blocket körs kommer den sprite som det är kopplat till att röra sig ett antal steg i den riktning som spriten pekar. Din katt pekar ju just nu mot 90 grader, rakt åt höger, så om det här blocket körs kommer katten att gå tio steg åt höger. Det står "10" i rutan från början, men det går att ändra till vad du vill. Det struntar du i just nu, för tio steg är lagom.

Klicka och dra det blå blocket till scriptytan, och håll koll på vad som händer när du börjar närma dig ditt orange händelseblock som redan ligger där.



Scratch visar att blocken kommer att passa ihop

Ser du att när du för det blå blocket mot underkanten av ditt orange händelseblock så lyser kanten upp? Det här är Scratch sätt att visa för dig att de här blocken kan sitta ihop. Om du släpper det blå blocket medan den vita kanten lyser kommer Scratch att sätta ihop blocken till ett script. Släpp det blå blocket så att det passar ihop med det orange händelseblocket. Om du missar gör det inget, du kan ta tag i det blå blocket igen och passa in det bättre, tills det fastnar där det ska.



Two block bildar ett script

Grattis! Du har skapat ditt allra första script. Testa nu att klicka på startknappen. Katten hoppar tio steg åt höger, men stannar där. Scratch gör alltså precis som du sagt åt det – när knappen med den gröna flaggan klickas på så går katten tio steg framåt. Men du ville ju att katten skulle springa fram över skärmen, så nu måste du berätta för Scratch att du vill att katten ska fortsätta att springa, inte bara hoppa fram tio steg en enda gång när du klickar på startknappen.

### Loopar – så gör du saker flera gånger

Att se till så att ett eller flera block körs flera gånger i rad automatiskt kräver att du skapar en "loop". Det finns olika typer av loopar, eller framförallt finns det olika sätt att kontrollera hur looparna startar och slutar, men för tillfället ska du skapa en oändlig loop – något som ska utföras under hela tiden vårt program körs, från det att du trycker på startknappen tills du trycker på stoppknappen.



*Två block bildar ett script*

Klicka på den brandgula kategorin "Kontroll" i blockbiblioteket. Du ser en mängd brandgula block som alla handlar om olika sätt att kontrollera flödet i ett script. Nu ska du använda ett block som skapar en loop, och det ser lite annorlunda ut. Det har fortfarande en inbuktning på ovansidan, så det går att haka på efter andra block. Men det har ingen utbuktning på undersidan, så det går inte att haka på fler block efter. Varför då? Jo, när Scratch kommer till det här blocket så kommer det att börja köra alla block som ligger inuti detta block, om och om igen i evighet. Eller åtminstone tills du trycker på startknappen.



*Blocket "För alltid"*

Som du ser har det här blocket alltså en öppning på mitten, vilket gör att du kan lägga ett eller flera block inuti detta. De block som ligger inuti körs i loop, så länge loopen körs.

Det du vill göra är ju att se till så att katten springer över skärmen hela tiden så fort du trycker på startknappen. Du ska alltså lägga det blå blocket "gå 10 steg" inne i en loop, så att Scratch förstår att du vill att blocket "gå 10 steg" ska köras om och om igen. För att göra detta måste du först plocka isär det blå blocket "gå 10 steg" från det orange blocket "När [grön flagga] klickas på". Att ta isär block är lika lätt som att sätta ihop dem. Ta tag i det blå blocket och dra det lite åt sidan. Du kan lägga det bredvid det orange blocket i scriptytan, bara de inte sitter ihop längre.

Nu går det bra att dra in vårt brandgula loop-block "för alltid" och sätta ihop det med det orange "När [grön flagga] klickas på". Gör som tidigare, dra in det nya blocket tills du ser den vita lysande kanten och släpp så blocken sätts ihop.



*Startblocket och loopblocket sitter ihop, men det blå blocket är lagt åt sidan så länge.*

Nu kan du ta tag i det blå blocket "gå 10 steg" igen och dra in det i loopen. Se till att du släpper det när det blir en vit kant vid utbuktningen inuti loopen "för alltid", se bilden:



*Släpp blocket inuti loopen*

Nu har du skapat din första loop och i den har du lagt blocket "gå 10 steg". Det du gjort är att du berättat för Scratch att du vill att blocket "gå 10 steg" ska köras om och om igen, "för alltid", när startknappen klickats.



*Din första loop*

Testa att klicka på den gröna flaggan nu och se hur katten springer iväg hela vägen till den högra kanten av scenen och stannar allra längst ut, när han nästan försvunnit ur bild. Scratch har gjort som du sagt, men det blev inte riktigt bra ändå, för när katten kom till kanten av scenen tog det stopp. Varför då?

Scratch låter inte dina sprites försvinna bort från scenen, för då skulle du inte kunna ta tag i dem med musen och dra dem dit du vill senare. Därför struntar Scratch i din instruktion "gå 10 steg" när katten inte kan komma längre framåt. Det du behöver göra nu är att få katten att vända håll när han kommer till kanten. Som tur är har Scratch ett färdigt blått block för att göra just det, och det ligger också under kategorin "Rörelse". Blocket du ska använda nu heter "studsas, vid kanten".

A blue Scratch block with the text "studsas, vid kanten" in white.

Blocket "studsas, vid kanten"

Vad det här blocket gör är att helt enkelt låta en sprite studsas när det når fram till kanten av scenen. Att en sprite studsas betyder egentligen att den vänder 180 grader om. Eftersom din katt hade riktningen 90 grader, rakt åt höger, till att börja med kommer den nu att få riktningen -90 grader, eller rakt åt vänster när den studsas. Det här kanske låter rörigt, men egentligen behöver du inte bry dig om antalet grader på riktningen just nu, så länge du minns att "studsas, vid kanten" låter din sprite vända helt om i motsatt riktning när den träffar kanten.

För att Scratch ska kunna utföra studsningen behöver du ta tag i det blå blocket "studsas, vid kanten" och lägga det inuti loopen. Varför inne i loopen? Jo, för om du lägger blocket utanför loopen kommer Scratch bara att köra blocket en enda gång, och det är direkt efter det att du klickat på startknappen. Om katten inte råkar befinna sig vid scenkanten just då kommer den inte att studsas, och när du sen flyttar på katten med "gå 10 steg" inuti loopen finns där inget block som studsar katten när det behövs.

Ta alltså tag i det blå blocket "studsas, vid kanten" och lägg det inuti loopen. Det spelar ingen roll om du lägger blocket före eller efter det blå blocket "gå 10 steg", men lägg det efter just nu. Ditt färdiga script ska nu se ut så här:



Ditt första script är färdigt!



Klicka på startknappen en gång till. Nu börjar katten gå fram och tillbaka, fram och tillbaka på scenen. Varje gång den är på väg att springa bort vid scenkanten så kommer den att studsas och börja gå åt andra hållet.

Om du tittar på ditt script medan katten springer så ser du att hela scriptet liksom glöder, och omgärdas av en gul färg. Det är Scratch som visar dig att det scriptet körs för tillfället. När du börjar göra större program kommer du kanske att ha flera script som ligger bredvid varandra och körs vid olika tillfällen. Markeringen runt det script som körs är ett bra sätt att hålla koll på vad det är som händer för tillfället i Scratch.



Scriptet glöder när det körs

Okej, katten springer fram och tillbaka på scenen, men finns det något du kan göra för att det hela ska kännas lite mer som ett spel? Du kanske ska ta och göra så att katten jamar till när man klickar på den?

#### Flera script till samma sprite

Det finns inget som hindrar att du skapar flera script till samma sprite. Nu har du ett script som gör så att katten springer. Men om du vill att katten ska jama när du klickar på den kan du skapa ett till script till katten och lägga det bredvid ditt första script. Scriptytan är ganska stor, så det finns plats för många script om du vill skapa flera.

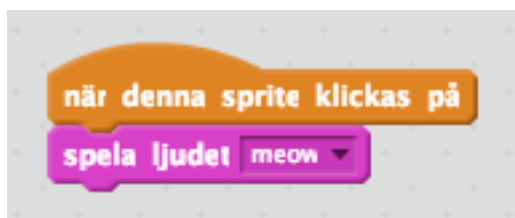
Klicka på kategorin "Händelser" igen. Här ligger ditt gamla bekanta orange block "När [grön flagga] klickas på". Men här ligger också ett annat händelsblock, nämligen "När denna sprite klickas på". Det här blocket vill du ha nu, för detta block körs bara när du klickar på spriten med musen. Dra in det orange blocket "När denna sprite klickas på" och lägg den någonstans i scriptytan, en bit ifrån ditt första script.



Du sätter igång med ett till script bredvid ditt första.

### Ljud – låt katten jama

Det här kommer att bli ett mycket enklare script, för det enda du vill göra är att spela upp ett "mjau" när du klickar på spriten. Klicka på den rosa kategorin "Ljud" och leta efter blocket "spela ljudet [meow]". Dra in det rosa blocket "spela ljudet [meow]" och haka fast det under det orange blocket "när denna sprite klickas på". Som du ser ligger ordet "meow" i en liten ruta med en pil. Det finns flera block som har samma utseende, och gemensamt för dem är att den lilla rutan låter dig välja mellan flera alternativ, om det finns. Scratch börjar med att bara ha ett ljud laddat, nämligen "meow", så just nu går det inte att välja något annat ljud, men det passar ju bra eftersom du vill att katten ska jama.



Ditt andra script är färdigt

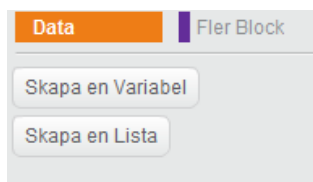
Ditt andra script är redan färdigt. Klicka på startknappen och försök att träffa katten med ett klick med muspekaren medan den springer på scenen. När du träffar katten så jamar den!

### Variabler – håll något i minnet

Det här börjar ju likna ett spel! Men vore det inte lite roligare om du fick en poäng varje gång du fick katten att jama? För att räkna poäng måste du låta Scratch hålla en siffra i minnet, nämligen själva poängen. Du vill att poängen ska börja på noll och sen räknas upp med ett varje gång du träffar katten.

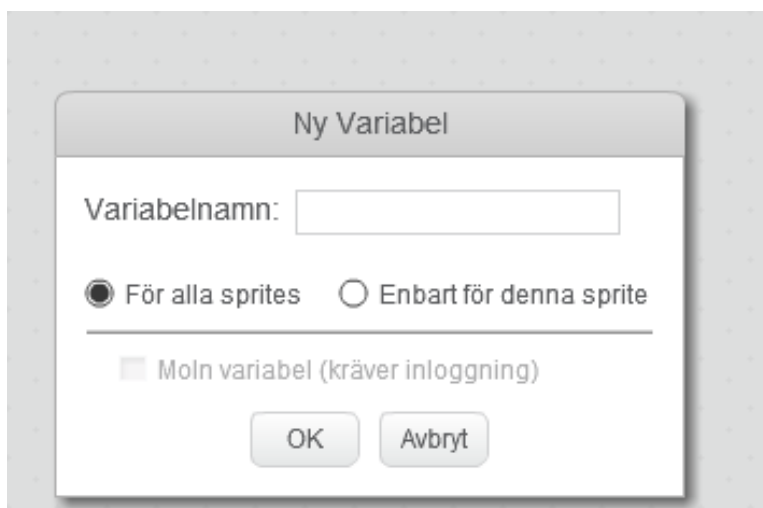
För att ett program ska kunna hålla saker i minnet finns något som kallas "variabler". Variabler kan man tänka på som ett slags lådor som du kan lägga olika saker i. När du lagt något i lådan kan du syssla med annat ett tag, men när du tittar i lådan igen ligger dina saker kvar. Du kan lägga till eller ta bort saker ur lådan, och Scratch håller ordning på innehållet åt dig. En sån låda blir utmärkt att använda för att hålla koll på poängräkningen i ditt allra första spel.

Scratch kan hantera massor av variabler samtidigt. Just nu behöver du bara använda en variabel, och för att använda den måste du först skapa den. Klicka på kategorin "Data" så ser du att här finns inga block just nu, men däremot ett par knappar.



Knapparna för att skapa variabler

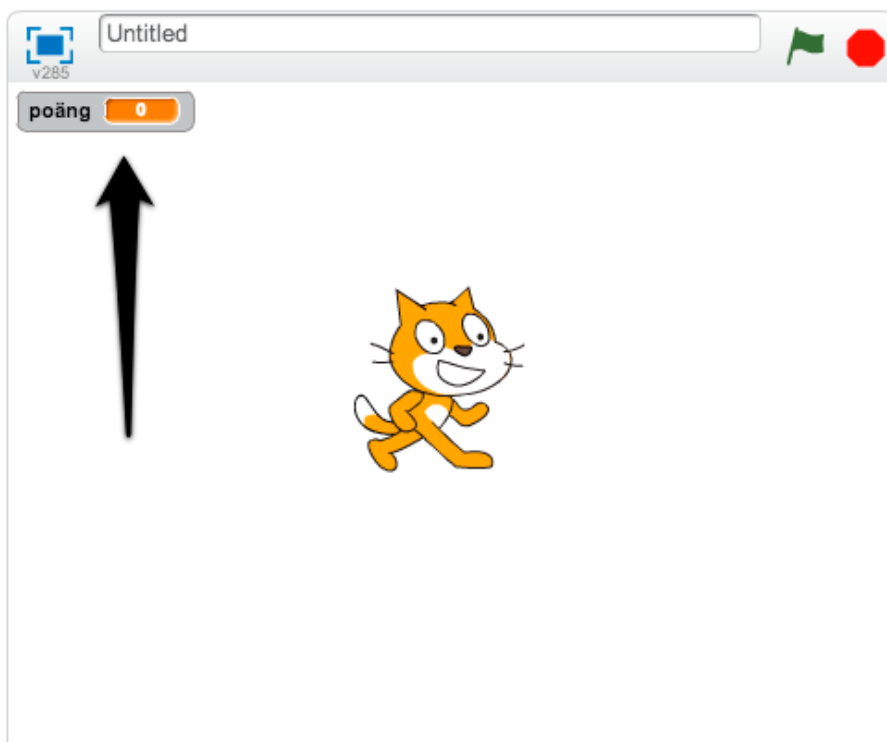
Du vill skapa en ny variabel, så klicka på "Make a variable". Genast blir det lite svårare, för nu tycker Scratch att du ska svara på ett par frågor:



Variabelfrågor

Först av allt vill Scratch att du döper vår nya variabel. Det är enda sättet som Scratch kan hålla koll på olika lådor senare, så skriv in "poäng" i fältet för variabelnamn. Nästa val handlar om vilka sprites som ska få använda den här variabeln. För tillfället spelar det ingen roll, så låt det valet vara och klicka på "OK" för att skapa variabeln "poäng".

Genast händer två saker. Det första du ser är att det nu plötsligt dök upp ett antal orange block som alla har med din nya variabel att göra. Dessutom dök en ny liten ruta upp längst upp till vänster på scenen:



Din nya variabel  
visas på scenen

Det här är ju perfekt för dig, för du vill ju visa poängen hela tiden medan spelet pågår. Men om du tittar i blockbiblioteket så ser du att vår nya variabel ligger överst i en orange oval, och att det finns en liten kryssruta till vänster om den:



*Variabeln du skapat.*

Testa att kryssa ur rutan till vänster om din variabel så ser du att poängrutan nu försvinner från scenen. Kryssa i rutan igen så kommer den tillbaka på samma ställe igen. På detta sätt kan du själv välja om dina variabler ska visas på scenen eller inte. Man kan använda variabler på många olika sätt. En del variabler, som din nya poängvariabel, är viktiga för den som använder programmet att ha koll på. Den som spelar ditt nya spel vill ju veta hur många poäng den har. Men många variabler behövs bara inuti själva programmet för att hålla koll på olika saker. Då är det onödigt att visa dem för användaren, och du kommer att göra mer av det längre fram.

#### Att räkna upp eller ner variabler

Nu har du alltså en låda att spara spelarens poäng i, men om du sätter igång programmet nu så ser du att poängen aldrig räknas upp oavsett hur många gånger du klickar på katten. För att Scratch ska veta när poängen ska räknas upp måste du berätta när och hur det ska göras.

Du har ju redan ett script som håller koll på när katten träffas av muspekaren. Det scriptet gör så att Scratch spelar upp ljudet "meow" när man klickar på katten. I det scriptet blir det perfekt att också lägga in ett block som räknar upp poängen.

Under "Data" ligger nu ett antal nya block som alla har att göra med din variabel. Bland dem ser du det orange blocket "ändra [poäng] med 1". Ta det blocket och dra in det till vårt script, under det rosa blocket "spela ljudet [meow]".



*Poänguppräknning på plats*

Det du nu gjort är att säga åt Scratch att "när denna sprite klickas på", "spela ljudet [meow]" och "ändra [poäng] med 1". Det där "ändra [poäng] med 1" betyder

att du ska ta det tal som för tillfället ligger i variabeln "poäng" och lägga till ett. Som du ser ligger ettan i en vit boll, och det betyder att du kan skriva in vilken siffra som helst där. Om du vill att man ska få tio eller tusen poäng varje gång man klickar på katten går det att lägga in siffrorna "10" eller "1000" där istället. För tillfället låter du det ge en poäng i taget att klicka på katten, så låt ettan stå kvar.

Nu kan du testa att köra igång programmet igen, och klicka på katten. Förutom att katten jamar när du träffar honom med muspekaren (stackars katt!) så får du ett poäng för varje träff, vilket också visas direkt i den lilla rutan för din variabel "poäng" på scenen.

Här finns en viktig sak att förstå. Att det står "poäng" bredvid din poängsiffra på scenen är bara för att du döpte variabeln till "poäng" när du skapade den. Du hade lika gärna kunnat döpa den till "sillar" eller "jamningar" eller vad du vill. Då hade det stått det istället bredvid siffran på scenen.

#### Scenen kan också ha script

Du får ett poäng varje gång du träffar katten, men om du missar katten händer ju inget, så du kan klicka hej vilt för att få poäng utan att det gör något. Du kanske skulle ta och straffa spelaren om han eller hon missar katten och istället bara klickar bredvid?

I ditt första program är scenen rätt tråkig – bara en vit ruta. Men scenen kan ha egna scripts och det kommer du att testa nu. För att visa scenens script klickar du på rutan för "Scen" till vänster om spritelistan:



*Klicka här för att visa scenens script*

Nu blir scriptytan tom, för scenen har inga script än. För att visa kattens script igen klickar du bara på "Sprites" till höger, och på så sätt kan du växla mellan scenens script och de olika spritens script medan du skapar nya program.

I scenens scriptyta ska du nu lägga ett script som tar bort en poäng varje gång man klickar på scenen. För om du klickar på scenen har du ju missat katten. Börja med att välja den orange kategorin "Händelser" och dra in startblocket "när denna sprite klickas på". Namnet på detta block är något förvirrande nu, för det är ju ingen sprite du klickar på utan scenen, men blocket heter likadant ändå.

Nu har du början på ett script som kommer att köras varje gång man klickar på scenen. Under kategorin "Data" har du din gamle bekant, det orange blocket "ändra [poäng] med 1". Dra in det blocket och haka fast det under "när denna sprite klickas på". Som det ser ut nu skulle du alltså få ytterligare en poäng i din variabel om du klickar på scenen, men du vill ju dra av en poäng istället. Det är lätt gjort – klicka på siffran "1" i den vita bollen i blocket och skriv "-1" istället. Samma block används alltså för att både lägga till och dra bort poäng, och det är vad du skriver in i den vita bollen som avgör hur värdet på variabeln ändras.



*Dra av en poäng när scenen klickas på*

Kör igång programmet igen och testa att klicka på katten och att klicka på scenen. Som du ser så minskar poängen när du klickar på scenen, men du ökar poängen när du träffar katten.

#### **Initiering – hur ska du ha det när du börjar?**

Spelet är nästan klart, men om du klickat på katten och scenen några gånger nu så är vår poängsiffra antingen plus eller minus någon siffra. Du vill att poängen ska vara noll när spelet börjar, så att spelaren inte får börja spelet med en massa minuspoäng i bagaget, till exempel.

Att ställa in variabler och andra saker i ett program när det körs kallas att initiera programmet. Du vill kanske nollställa poängräknare, placera saker på rätt ställe eller göra andra inställningar så att alla spelare får samma förutsättningar varje gång ett nytt spel påbörjas. Du har ju redan ett script som körs när grön flagga klickas på, och det ligger kopplat till "Sprites", det vill säga katten. Klicka på katten i spritelistan för att se kattens script igen.

I ditt lilla spel räcker det ju om du ser till att variabeln "poäng" sätts till noll när spelet börjar. Under kategorin "Data" finns det orange blocket "Sätt [poäng] till 0". Dra in det till scriptet som har startblocket "när [grön flagga] klickas på" och se till att släppa det innan loopen men efter startblocket:



*Placera det nya blocket mellan startblocket och loopblocket*

Varför lägger du inte det här blocket inne i loopen? Jo, du vill ju bara att poängen ska nollställas när spelet först startar, alltså när du klickar på den gröna flaggan. Om du lägger nollställningsblocket inne i loopen kommer poängen att sättas till noll oavbrutet, och det spelar ingen roll hur mycket du försöker räkna upp eller ner poängen – den sätts genast till noll ändå.

Det här är ett sätt att göra det här på. Du hade också kunnat skapa ett tredje script med samma startblock som ditt första – "när [grön flagga] klickas på" och lägga nollställningsblocket ensamt där. Kom ihåg att en sprite kan ha många script, och det kan finnas flera script som har samma händelseblock som start.



*Ett alternativ till initiering*

När du senare gör program som har många och långa script kan det vara en god idé att hålla isär script som gör olika saker, men just nu spelar det inte så stor roll, så du kan välja vilken av de två lösningarna du vill.

Kör igång spelet nu genom att trycka på den gröna flaggan. Poängen sätts genast till noll och du kan börja spela genom att jaga katten. När du avslutar spelet och startar om det kommer poängenställningen alltid att börja från noll.



## Ljud – allt blir roligare med musik

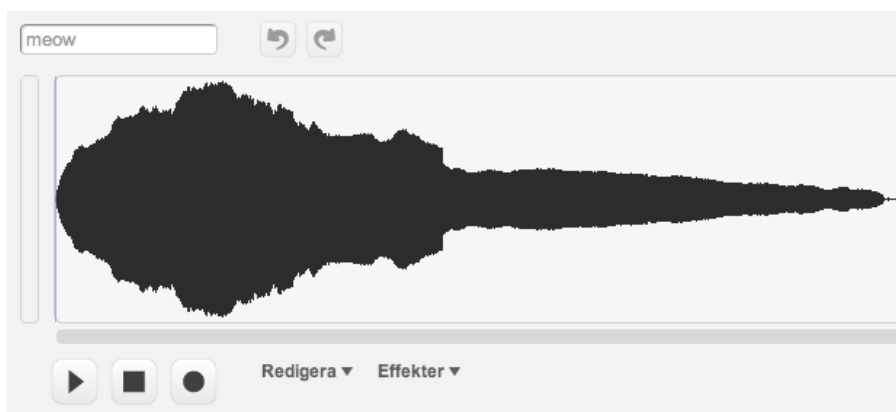
Du har redan använt ett ljudblock för att få katten att jama när du klickar på den, men det finns mycket mer du kan göra med ljudblocken i Scratch. Klicka på kategorin "Ljud" så ser du att här finns många block som har med instrument och taktslag att göra. Men du såg ju förut att det bara var "meow" som gick att välja i ljudblocket, så hur får du fler ljud i Scratch?



*Script, Costumes, Sounds*

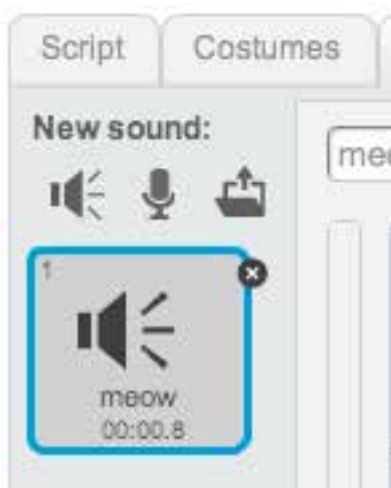
Jo, över blockkategorierna finns tre flikar. Först har du "Script", och det är den som är vald när du startar Scratch. Nästa kategori heter "Costumes" och den pratar gäs igenom om lite senare, för just nu är det "Sounds" som är intressant. Klicka på den.

Här ser du nu ljudet "meow" som kommer förloadat i Scratch. Du ser hela ljudet som en ljudvåg och du kan lyssna på det genom att trycka på playknappen.



*Ljudet "meow"*

När du vill ha fler ljud i Scratch använder du den lilla menyn som nu har ersatt blockkategorierna och där finns tre knappar att välja mellan:

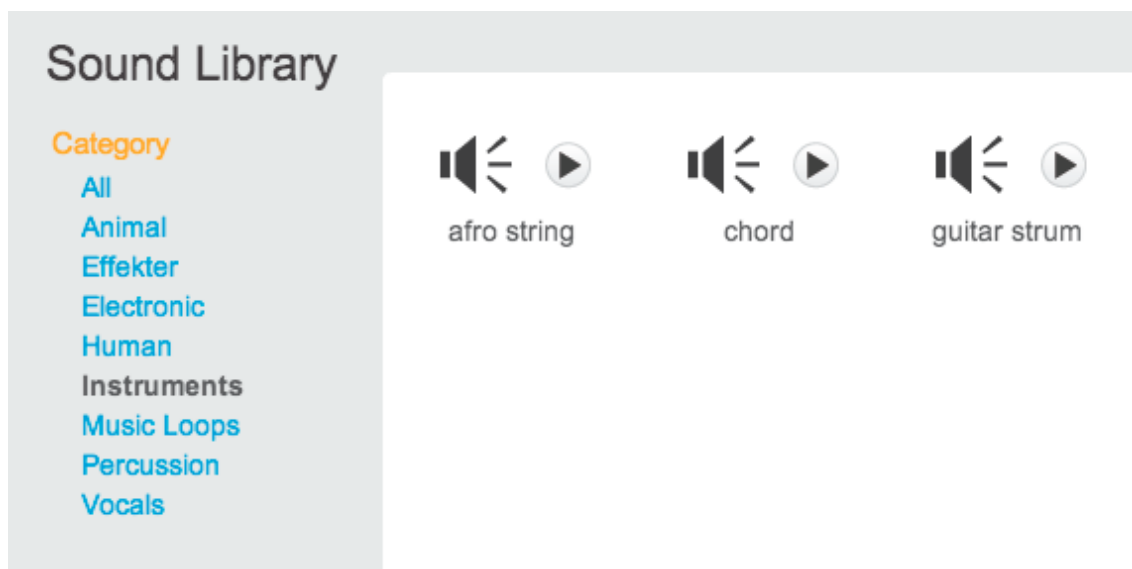


Ljudmenyn

Den första knappen låter dig ladda in ett ljud från Scratch inbyggda ljudbibliotek. Den andra knappen låter dig spela in ett eget ljud med datorns mikrofon. Och den tredje knappen låter dig ladda upp en ljudfil från din egen dator. För tillfället ska du försöka att hitta ett ljud från Scratch medföljande ljudbibliotek, så tryck på den första knappen, den som ser ut som en högtalare.

#### Ladda in ljud från det medföljande biblioteket

Du får upp en lista på massor av ljud, och till vänster har du möjlighet att filtrera ljud efter olika kategorier. Klicka på "Instrument" i listan till vänster.



Ljudbiblioteket, filtrerat på kategorin "Instruments"

Genom att trycka på den lilla playknappen bredvid varje ljud kan du provlyssna på alla ljuden. Du kan använda det ljud som heter "guitar strum" som grattisljud när spelet är slut, så dubbelklicka på det ljudet, eller klicka på det en gång för att markera och sen på knappen "OK" längst ner till höger.



*Ljudet "Guitar strum" laddat*

När ljudet laddats hamnar det i ljudlistan under "meow" som ju redan låg där innan. På det här sättet kan du ha massor av ljud inne i Scratch och använda dem vid olika tillfällen i dina program.

Klicka på fliken "Script" för att visa blockbiblioteket och scriptytan igen.

## Villkor – kör bara koden om något stämmer

Väldigt ofta vill du att ett block eller ett helt script bara ska köras om något annat händer, eller bara om en variabel har ett visst värde. Du kanske har gjort ett spel och vill att något ska hända varje gång du trycker på mellanslagstangenten, eller så vill du ge spelaren minuspoäng om en sprite träffar en annan. Det finns flera sätt att hålla koll på att saker körs bara under vissa förutsättningar, och det handlar om villkor.

Under kategorin "Kontroll" ligger alla block som hjälper dig att bara köra script när vissa förutbestämda saker inträffat. Det viktigaste och mest använda av dem heter "om < > då". Det du kan göra med det blocket är att först kolla efter något – till exempel värdet på en variabel – och endast om det stämmer med vad du vill, gå vidare och köra de block som ligger inuti "om < > då". Som du ser har "om < > då" en diamantformad ruta i mitten. Det finns nämligen ett antal block i Scratch som är just diamantformade, och det är bara de som passar att lägga till i "om < > då".



Blocket "om < > då"

Du kan fundera på några exempel på när det vore intressant att hålla koll på ett villkor. Exempelvis om du vill att ditt spel med katten ska ta slut när man fått 10 poäng. Då skulle du kunna använda blocket "om < > då" för att säga något i stil med "OM (värdet på variabeln poäng = 10) DÅ ... gör nånting!" Men hur kollar du då om värdet på din variabel är tio? Det här kräver ett antal steg. Titta först under den gröna kategorin "Operatörer". Här finns massor av block, men det du behöver för att jämföra två värden är just diamantformat och heter "[ ] = [ ]":



Operatören "[ ] = [ ]"

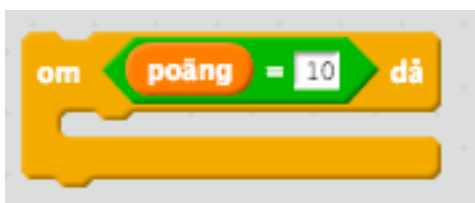
Som du ser kan du skriva in egna värden här, men det är ju ganska meningslöst att skriva in två fasta värden. Riktigt användbart är detta block för att du kan

släppa ett variabelblock i ena hålet. Titta under kategorin "Data" så ser du att vår variabel "poäng" har ett helt eget, ovalt orange block där. Det blocket passar i endera hål i det gröna operatorblocket, och det blocket i sin tur passar i det brandgula "om < > då":



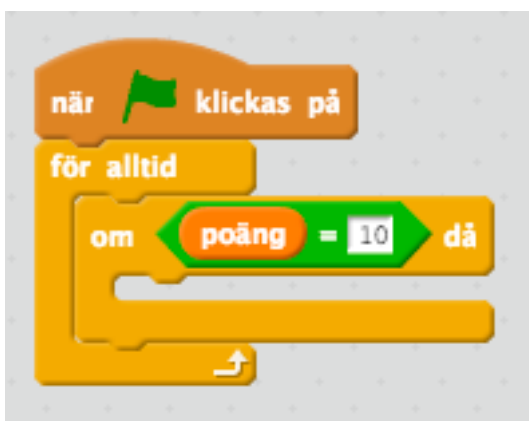
Variabelblock inuti operatorblock inuti kontrollblock...

Nu kan du skriva in siffran "10" i det högra hålet, och plötsligt har du en kombination av block som tillsammans bildar ett villkor. Om poäng = 10, då ska någonting hända.



Om poäng = 10 då

För att använda den här blockkombinationen måste du lägga den i ett script som körs. I ditt spel kan du lägga den i ett nytt script, och för att det ska kollas hela tiden behöver du lägga det i en loop som körs för alltid:

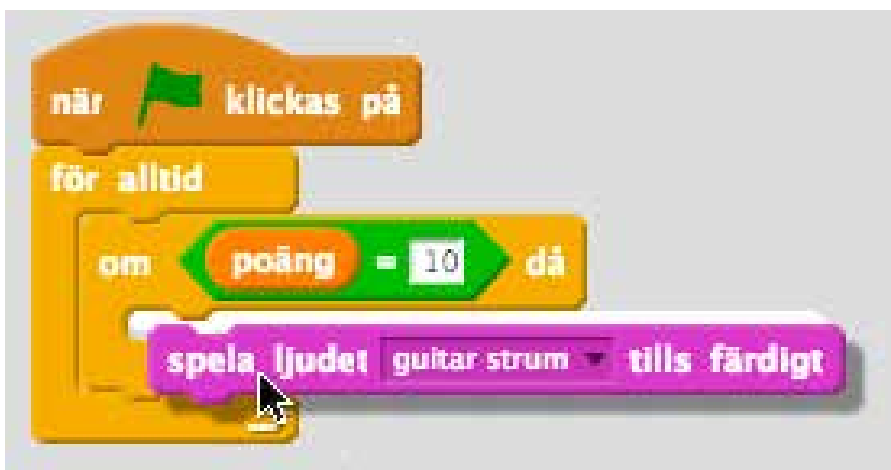


Script för att kolla om poäng = 10

Det här scriptet kommer att börja köras när den gröna flaggan klickas på, och loopen som körs "för alltid" ser till att Scratch hela tiden kollar om poäng är 10,

och då... Ja, vadå? Än så länge är det tomt inuti vårt villkorsblock. Du kan väl spela en liten fanfar för spelaren? En sån tur att du redan laddat in ljudet "guitar strum"!

Det finns ett annat block för att spela upp ljud som är användbart här. Det heter "spela ljudet [] tills färdigt", men eftersom Scratch gissar att du troligen vill använda det senaste ljudet som laddats in är ditt nya ljud redan förvalt, så det står "spela ljudet [guitar strum] tills färdigt" på det just nu. Det fina med det blocket är att det spelar upp ljudet och väntar snällt tills ljudet har spelat klart. Dra in det blocket och lägg det inuti ditt villkor:



Släpp ljudblocket  
inuti villkorsblocket

De block som ligger inuti vårt villkorsblock är ju de som kommer att köras bara när villkoret är uppfyllt. Om du skulle lägga det här ljudblocket utanför villkorsblocket skulle det köras "för alltid" och det skulle bli en hel gitarrkonsert under spelets gång istället för bara när spelaren vunnit.

Nu kan du testa att köra spelet igen. När du fått tio poäng börjar gitarren att spela. Men som du hör slutar den aldrig, utan spelar om och om igen. Det är nämligen precis vad du sagt åt Scratch att göra. Om poäng = 10 då ska ljudet spelas. Eftersom villkoret ligger inne i en loop som körs "för alltid" kommer villkoret att vara sant så länge poängen är tio.

Det kanske vore bra om ljudet spelades, och sen fick spelaren börja om spelet från början igen? Du kan lägga till en liten paus också, så spelaren hinner förstå vad som händer. Och katten kan väl säga "Grattis, du vann"? Du fortsätter att bygga inuti ditt villkorsblock.

Under kategorin "Kontroll" finns blocket "vänta (1) sekunder". Dra in det blocket och lägg det efter "spela ljudet [guitar strum] tills färdigt". Ändra så att Scratch väntar tre sekunder istället.

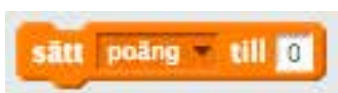


Vänta tre sekunder  
efter att ljudet spelats färdigt

Nu spelar Scratch alltså ett ljud och väntar tre sekunder när spelaren når tio poäng. Nu vill du att katten ska berätta för spelaren att hon eller han vunnit spelet, och för det ändamålet har Scratch fina pratbubblor inbyggda. Under kategorin "Utseende" finns det lila blocket "säg [] i () sekunder".

Ändra texten från "Hello!" till "Grattis, du vann!" genom att klicka i den vita rutan och markera texten. Och ändra antalet sekunder till fyra så spelaren hinner läsa.

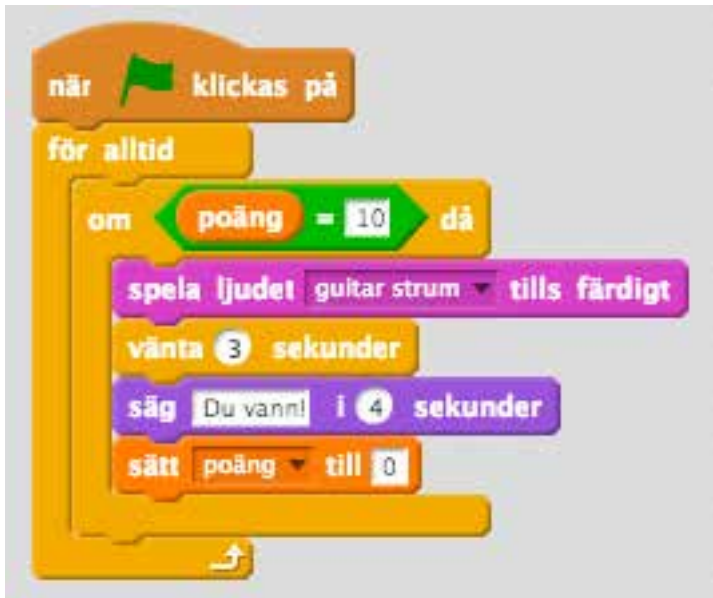
Scratch har spelat ljudet, väntat några sekunder och sen låtit katten säga att spelaren vunnit. Men för att ditt villkor ska sluta vara sant hela tiden och därmed bara fortsätta att köra blocken inuti om och om igen måste du se till att variabeln poäng inte längre är tio. Under kategorin "Data" finns det orange blocket "Sätt [] till []":



Blocket "sätt [] till []"

Just nu är det redan ifyllt på just det vis du vill, nämligen att variabeln "poäng" ska sättas till noll, så dra in det här blocket och lägg det sist inuti ditt villkorsblock. Det här blocket kan du annars använda för att sätta vilken variabel som helst till vilket värde som helst, så det är ett mycket användbart block.

Nu har du gjort klart scriptet som sköter spelets slut:

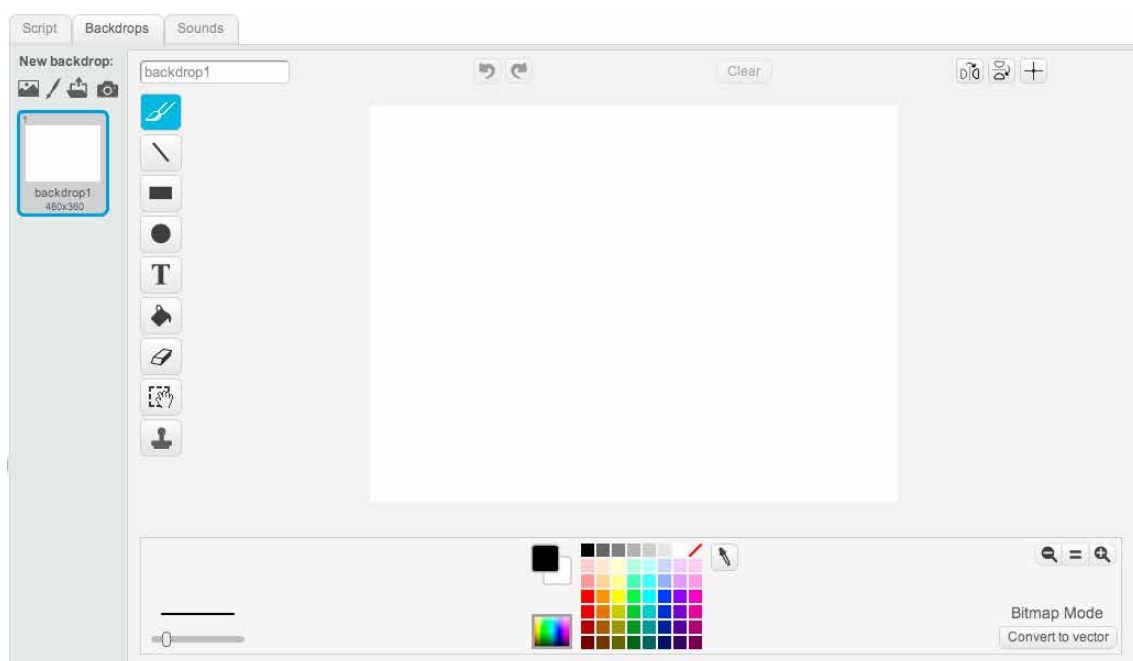


*Det färdiga villkorsscriptet*

Testa ditt spel nu, och se vad som händer när poängen når tio. Du får en liten trudelutt, katten säger att du vunnit och sen börjar det om igen för poängen sätts till noll.

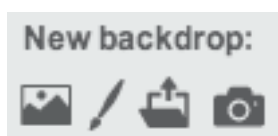


## Bakgrunder (klädslar) – scenens dekor



*Backdrops*

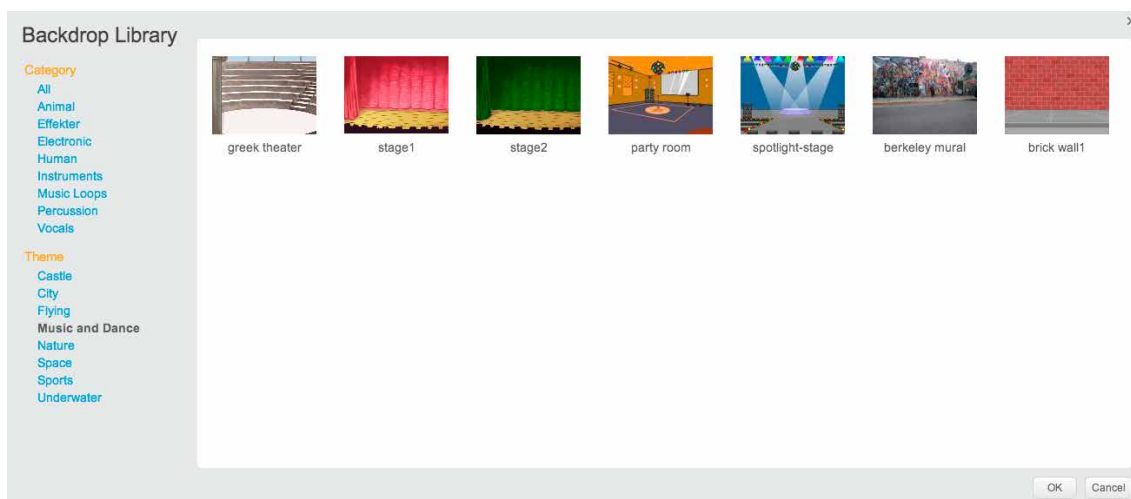
Det här är en rolig del av Scratch, för här inne gömmer sig ett helt litet ritprogram. Även om katten ser ut att springa mitt i ingenting springer han faktiskt på en bakgrundsbild, även om den är helt vit från början. Om du vill kan du använda penseln eller de andra målarverktygen och färgerna här för att rita din egen bakgrund.



*Menyn för ny bakgrund*

Överst till vänster i den här vyn hittar du valen för ny bakgrund. De är ganska lika valen som finns för ljud. Först är det något som ser ut som en tavla. Här får du välja bland de medföljande bilderna att ladda in. Sen kommer en liten pensel, och den knappen skapar en ny, tom och vit bakgrund att måla på. Den tredje knappen låter dig ladda upp en bild från din egen dator att använda som bakgrund. Den sista knappen är en liten kamera, och när du klickat på den kan du använda din dators webbkamera (om du har någon) för att ta ett foto att använda som bakgrund.

Just nu ska du testa de medföljande bakgrunderna, så klicka på den första knappen, den som ser ut som en liten tavla. Nu öppnar sig biblioteket med massor av bakgrunder att välja på, och till vänster kan du precis som i ljudbiblioteket filtrera på olika kategorier. Välj kategorin ”Musik och Dans”.



Bakgrunder filtrerat på ”Music and Dance”

Ladda in bakgrunden som heter ”brick wall1” genom att antingen dubbelklicka på den eller först klicka på den en gång för att markera den och sen klicka på ”OK” längst ner till höger.

Genast laddas bakgrunden in och läggs den på scenen. Nu står din sprite med katten ovanpå scenen med en bakgrundsbild, och kanske ligger katten inte riktigt rätt – för om den inte står på rätt ställe ser det ut som att katten flyger ovanför marken:



*Bakgrund laddad, men katten flyger!*

Det är lätt åtgärdat. Ta bara tag i katten med muspekaren och dra den till rätt höjd så det ser ut som att katten står på gatan istället. Om spelet är igång blir det svårt, så stoppa spelet först med stoppknappen.

Nu springer katten fram och tillbaka på gatan och dina script funkar precis som förut.

## Fortsättning följer

Du har nu gjort ett första, väldigt enkelt spel med Scratch. Under tiden har du lärt dig om sprites och script, loopar och villkor, operatorer och variabler. Det finns väldigt mycket du kan göra med bara olika varianter av de block du använt hittills. Du kan rita egna bakgrunder och sprites. Du kan spela in egna ljud och göra spel eller interaktiva filmer. Men Scratch kan mycket, mycket mer än så. Den här instruktionsboken kommer att utökas med många fler kapitel inom en snar framtid. Ha kul med Scratch, experimentera och lär dig mer på egen hand så länge!