

# Internetguide #31

# Barnhack



Kom igång med programmering!



Anders Thoresson

# I den här guiden lär du dig...

- ☑ Varför det är så viktigt att lära sig lite om programmering och it
- ☑ Vad barnhack innebär
- ☑ Vad du behöver kunna och inte behöver kunna
- ☑ Verktyg som är en bra introduktion
- ☑ Grundläggande termer som används vid programmering
- ☑ Hur du kan lära barnen grunderna i programmering genom fysiska lekar
- ☑ Bra sätt att börja programmera tillsammans med barnen
- ☑ Bra sätt att förbereda dig innan du sätter igång tillsammans med barnen
- ☑ Hur du själv kan leda barnhack med flera barn
- ☑ Att arrangera barnhack och engagera andra att delta
- ☑ Hur du sprider information om barnhack på webben

# Innehåll

<b>Förord</b>	<b>4</b>
<b>1. Inledning</b>	<b>5</b>
1.1 Att förstå är att kunna ha en åsikt	7
<b>2. Experimentera och upptäck</b>	<b>10</b>
2.1 Vad behöver ni lära er?	11
2.2 Våga hoppa med barnen	14
2.3 Planera vad ni ska göra	15
<b>3. Verktygen</b>	<b>16</b>
3.1 Andra verktyg	19
3.2 Programmeringsparlören	19
<b>4. Fysiska hack</b>	<b>21</b>
4.1 En introduktion till programmering	22
4.2 En vidareutveckling	23
<b>5. Så hackar vi - tillsammans med barnen</b>	<b>24</b>
<b>6. Så hackar vi - tips för föräldrar (eller mostrar)</b>	<b>27</b>
<b>7. Så hackar vi - kursledaren</b>	<b>31</b>
7.1 Ett Scratch-spel i fem steg	33
7.2 Övningar i Kojo	34
<b>8. Så hackar vi - arrangören</b>	<b>35</b>
8.1 Hitta en kursledare	36
8.2 Locka deltagare	37
8.3 En plats att vara på	37
8.4 Internetuppkopplade datorer	38
8.5 Välj verktyg	38
8.6 Projektor till kursledaren	38
8.7 Förfriskningar	38
8.8 Tillstånd att fotografera	38
8.9 Ett schema	39

<b>9. Så sprider du barnhackandet</b>	<b>40</b>
9.1 Det här är en webbplats	41
9.2 Vad behöver ni?	41
9.3 Kom igång på Wordpress.com	42
9.4 Ge barnhacket en egen mejladress	42
9.5 Lär er mer om webbpublicering	43
<b>10. Länklista</b>	<b>44</b>

# Förord

Idag använder de flesta unga i Sverige uppkopplade prylar och datorer. Hårdvaran, programmen och infrastrukturen som de kopplas upp till används flitigt, medan förståelsen för hur allt fungerar och hänger ihop är rätt liten. Den här guiden har sin bakgrund i att Måns Jonasson på IIS efter arbetstid höll kostnadsfria programmeringskurser för barn och deras föräldrar. Gensvaret var översvallande positivt och varje kurstillfälle blev fullbokat inom en timme efter att ha annonserats. Sedan dess har vi genomfört rikstäckande barnhacksturnéer och utbildad hundratals barn och lärare i programmering. Men av praktiska skäl är det inte möjligt för en relativt liten organisation i Stockholm att möta det stora behovet av lek och lärande med programmering i fokus som uppenbarligen finns i samhället. I stället har vi samlat erfarenheter och klokskap från olika håll och sammanställt dem i den här skriften, bland annat Tekniska museet. Syftet är att du och ditt barn (eller kursledare och andras ungar) ska kunna genomföra egna kurstillfällen där ni bor. Eller laborera hemma tillsammans. Det är därför tilltalet i texten ibland är niande: Innehållet vänder sig till dig som är vuxen och barnet gemensamt. Internetguiden som du läser just nu är en första del i en större satsning på att förklara it och internet för barn och unga. Vi anser att det är en viktig uppgift inför framtiden att väcka intresset hos juniorerna för hur datorer, programmering och nätet egentligen fungerar. Det är ytterst en demokratifråga, då det behövs kunskaper om it för att kunna vara med och påverka utvecklingen i både stort och smått. Om det ska bli några fler så kallade it-under i landet krävs sannolikt att generationerna på tillväxt inte endast kan använda innehåll på internet och appar i apparater. Med engagemang och hjälp av vuxna kan de yngre bli aktiva producenter i stället för att bara vara passiva konsumenter. Vare sig du eller barnen behöver vara mattesnillen eller ha teknik som största intresse för att komma igång. Och så är det väldigt kul att hacka!

**Anders Thoresson, Måns Jonasson, Jessica Bäck  
och Kristina Alexanderson**

Stockholm, April 2016

# 1. Inledning



## 1. Inledning

Det spelar ingen roll hur smart en dator tycks vara. Det spelar ingen roll hur användbar en modern mobiltelefon är. Det spelar ingen roll hur märkvärdiga tjänster som dyker upp på internet.

Prylarna kan verka hur magiska som helst – men de lyder bara instruktioner. En dator, en mobiltelefon – hela internet – gör bara vad människor bestämt.

De som fascineras mest av möjligheterna blir programmerare eller ingenjörer och lär sig utnyttja och utveckla tekniken för nya häpnadsväckande saker. Men också för den som väljer en helt annan yrkesväg är det viktigt att förstå något av vad som händer innanför skalet på prylarna omkring oss.

Med mobiltelefoner och surfplattor blir it i allt större utsträckning en färdigförpackad konsumentprodukt. På många sätt är det något positivt, en utveckling som inneburit att många fler än de mest teknikintresserade kan ha glädje och nytta av utvecklingen. Samtidigt är den grundläggande tekniken dold bakom färdiga funktioner och finesser. Och med det går två saker förlorade.

De föräldrar som själva programmerar eller gjorde det som unga kanske sörjer att deras barn inte utnyttjar datorerna till deras fulla potential. Det råder förvisso ingen brist på kreativa program och tjänster som låter barnen skapa de mest häpnadsväckande saker, men fortfarande bara under de förutsättningar som den utvecklare som gjort programmet eller tjänsten ställt upp. Men att den enskilda individen riskerar att missa glädjen med att på egen hand och med datorns hjälp skapa är en del av problemet. En annan, och kanske viktigare del, är demokratiaspekten.

- Vi använder it så oerhört mycket idag, men de flesta kan väldigt lite om hur tekniken fungerar. Vi behöver ta kontroll över den, säger Tomas Ribba, pedagogiskt ansvarig på Tekniska museet i Stockholm.

Björn Regnell, professor i datavetenskap på Lunds Universitet resonerar på ett liknande sätt:

- Mjukvara blir allt mer samhällsviktig men också livsavgörande. Både rent konkret, som i livsuppehållande maskiner på sjukhusen, men också i vårt ekonomiska system, på aktiemarknaden, inom statlig förvaltning och i rättsstaten. Därför blir det en grundläggande allmänbildning, precis som att läsa och skriva, att förstå hur tekniken fungerar. En mjukvaruanalfabet kommer inte att kunna ta ställning i politiskt brännande frågor, säger han.

## 1.1 Att förstå är att kunna ha en åsikt

Varför ser sökresultatet i sökmotorn ut som det gör? Hur kan det sociala nätverket veta vilka som är dina vänner och föreslå att du lägger till dem? Är elektroniska riksdagsval en bra eller dålig idé? Vilka personer har möjlighet att läsa innehållet i ditt känsliga mejl? Vilka digitala fotavtryck lämnar du efter dig på nätet?

Hos sökmotorn finns regler (som skapats av företagets anställda) som rankar vilken webbplats som bäst matchar orden som matas in i sökfältet. Vem det är som gör sökningen är ofta en av faktorerna som vägs in i det beslutet, men också andra saker, som var i världen hen befinner sig och vilka webbplatser hen besökt tidigare. På sociala nätverk används andra regler (också skapade av anställda) för att hitta sannolika kopplingar i vänskapsväven.

Så fortsätter det. För den som börjar fundera finns det gott om frågor att ställa om vår digitala vardag. Och bakom alla svar finns ett beslut fattat av en människa.

För så här ligger det till: Datorer är bra på att räkna snabbt och på att följa instruktioner. Däremot saknar de alla möjligheter att på egen hand hitta på saker. Allt som vi vill att en dator ska göra måste någon därför instruera den att göra. Det finns inga naturlagar som bestämmer vilken webbplats som ska hamna i topp när du använder sökmotorn, ingen magi som kan "veta" att du sannolikt kommer gilla en artist du aldrig tidigare hört.

I diskussioner kring barn och programmering finns återkommande invändningar. En av de vanligaste ifrågasätter behovet av att förstå vad som händer under skalet för att kunna använda tekniken.

### Viktigt! Vem är du som läser?

Den här boken, med tillhörande fördjupningsmaterial, vänder sig till dig som vuxen med barn i din närhet, dina egna eller andras. Du kanske är en förälder som vill väcka dina barns intresse för programmering, men kan lika gärna vara en moster, fritidsledare eller lärare.

Förhoppningsvis innehåller den här guidesamlingen tillräckligt mycket information och inspiration för att lägga grunden för många barnhack runt om i Sverige.





## 1. Inledning

”Jag behöver inte känna till hur en förbränningsmotor fungerar för att kunna köra en bil”, är ett resonemang som ofta hörs. Det stämmer på individnivå, även om den som kan mer har bättre förutsättningar att göra ett lyckat bilköp.

På samhällsnivå är det däremot bra ju fler som har en övergripande förståelse för de konsekvenser bilismen för med sig, exempelvis inom miljö och samhällsplanering. Kunskap gör det möjligt att ha en underbyggd åsikt när politiker presenterar förslag som ska styra bilåkandet åt det ena eller andra hållet, bildligt talat.

Informationstekniken har en enorm samhällspåverkan. Tekniken kan filtrera den information vi tar del av eller fatta snabba, automatiserade beslut. Ett sätt att skaffa sig en förståelse för hur it fungerar är att själv lära sig programmera. Eller åtminstone försöka lära sig programmera. Att få en praktisk förståelse för att det är människor som styr allt.

Thore Husfeldt, professor i datavetenskap vid Lunds Universitet, menar att det inte är en rimlig ambition att alla ska bli skickliga programmerare. Inte heller är det önskvärt, samhället behöver också många andra kompetenser. Däremot är det fullt realistiskt att alla ska vara dåliga programmerare, enligt honom. Och det är dessutom eftersträvansvärt:

- Att själv ha fått prova på och misslyckas ger många av de insikter som jag tror är väldigt värdefulla att ha som medborgare. Vi behöver inte fler systemtekniker, vi behöver fler jurister, sociologer och ekonomer som förstår datavetenskap, säger Thore Husfeldt och fortsätter:

- Samhället och våra demokratiska beslut och principer manifesteras i allt större utsträckning i form av it-system. Därför är det viktigt att många förstår vad it är.

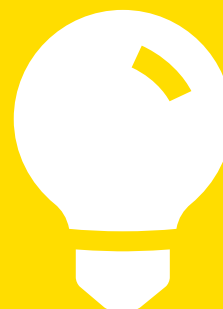
Med kunskap om hur datorer och internet fungerar blir du en bättre konsument och en mer kritisk medborgare. Politiker och tjänstemän som inte kan och förstår hur tekniken fungerar hamnar i händerna på teknikerna. Forskare som hanterar sitt material på ett vårdslöst sätt kan råka avslöja källor som ska vara hemliga. Personer som inte reflekterar över hur sökmotorer, sociala medier och andra webbplatser gör individuella anpassningar av sitt innehåll riskerar bland annat att missa saker som faktiskt är relevanta för dem. Sättet som Googles algoritmer väljer ut vad som ska ligga överst bland sökträffarna och sättet som Twitters algoritmer väljer ut de mest omtalade ämnena påverkar vad vi diskuterar med varandra. Att förhålla sig källkritiskt till internet handlar inte bara om informationen vi läser utan också om varför sökmotorn föreslår en viss sida.

Medvetenhet om att det är mänskliga beslut som styr all teknik ökar möjligheterna att förhålla sig kritiskt till den. Det är inte naturlagar som avgör vad som toppar sökresultatet, det är företagets

## Tips! Vad är ett ”barnhack”?

Ordet ”hack” har flera betydelser på engelska. Det kan bland annat betyda att en hackare, en programmerare, (som inte behöver ha brottsligt uppsåt, till skillnad från en cracker) löser ett problem. Det används också som beskrivning på en träff där många personer träffas och programmerar tillsammans.

Vi har därför valt att kalla de programmeringskurser för barn som IIS anordnar för Barnhack. Samma benämning används också av andra som arrangerar liknande träffar runt om i landet.



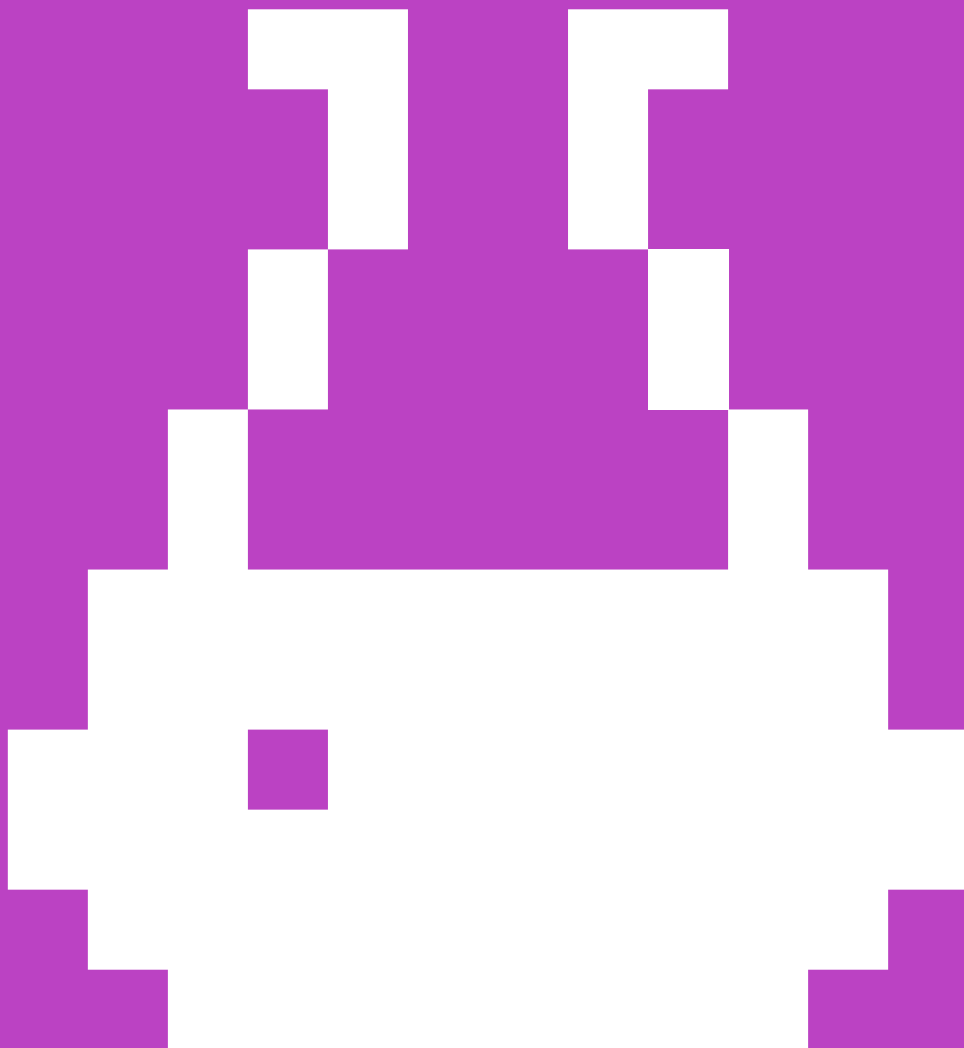
anställda, även om it-jättarna ibland föredrar att ducka bakom tekniken för att slippa ta ansvar. Hos Google finns exempelvis funktionen autocomplete som visar vilka de populäraste sökningarna är när du börjar mata in dina sökord. Ett antal personer runt om i världen har upptäckt att deras namn kompletteras med mindre smickrande omdömen. I ett fall försvarade Google sig med ett resonemang om algoritmer och användarnas bidrag:

*”We believe that Google should not be held liable for terms that appear in autocomplete as these are predicted by computer algorithms based on searches from previous users, not by Google itself.”*

Ett försvar som är en lek med ord. Ja, det är användarnas sökningar som används som utgångspunkt för funktionen. Men nej, det innebär inte att Google saknar ansvar. Det är företagets anställda som utvecklat reglerna för hur dessa ord ska användas.

Som användare av datorer, smarta telefoner och internet kan du nöja dig med att ha fått detta faktum berättat för dig. Eller så kan du börja experimentera och själv upptäcka vilka möjligheter du har att bestämma vad en dator ska göra.

## 2. Experimentera och upptäck



En dator är en maskin som bara gör vad den blivit tillsagd. Varken mer eller mindre. Av detta faktum följer två utmaningar.

För det första måste du lära dig prata ett språk som datorn förstår. De språk vi dagligen använder när vi pratar, skriver och läser fungerar nämligen inte. Datorer förstår varken svenska, engelska eller något annat mänskligt språk. Datorer använder bara ettor och nollor när de arbetar, men att som människa lära sig att behärska det som kallas för maskinkod är för svårt. Lösningen är i stället att mötas på halva vägen, med det som kallas för programmeringsspråk.

Ett programmeringsspråk är ett strukturerat sätt att skriva instruktioner som automatiskt kan översättas till de ettor och nollor som datorn använder. I programmeringsspråken används ord som vi människor förstår, men som i ett programmeringsspråk får en absolut betydelse, helt utan möjlighet till tolkning. För att lära dig att styra datorn behöver du därför lära dig ett programmeringsspråk.

Den andra utmaningen ligger i det absoluta och i att en dator bara gör det en människa sagt åt den att göra. Du har säkert svurit över att datorn inte gör som du vill. Men egentligen är det inte datorn och programmen i den som du ska vara arg på, utan den programmerare som gjort programmet som orsakar din frustration. Att kunna ett programmeringsspråk är därför den ena delen, den andra är bestämma hur du ska använda det i ditt program så att datorn utför precis det du vill att den ska göra.

I ett datorspel innebär det till exempel att fundera över vilka figurer som ska finnas med i spelet och vad som ska hända när de möts. När tennisbollen studsar mot spelarens racket ska den ändra riktning, men om bollen studsar två gånger i marken ska motståndaren få poäng. Detta kan datorn inte förstå utan måste få "berättat" för sig i form av programkod.

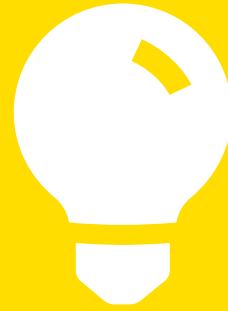
### **2.1 Vad behöver ni lära er?**

Hur mycket programmeringskunskap behöver ni? Svaret beror så klart på vad syftet med kunskapen är. Den som vill kunna jobba som programmerare måste givetvis lära sig massor. Men när du börjar utforska datorernas värld tillsammans med dina egna eller andras barn är det sannolikt inte målet. I stället är ambitionen att ha roligt i nuet och förhoppningsvis lära sig något som kan vara till nytta och glädje i framtiden.

Med ribban lagd på den nivån som Thore Husfeldt föreslår – att mycket är vunnet även när man är en dålig programmerare – blir utmaningen också mer rimlig för en förälder som själv inte kan programmera men ändå vill utforska datorernas värld tillsammans med sina barn.

### Tips! Ett recept för datorn

Det finns vissa likheter mellan ett kakrecept och ett datorprogram. I båda fallen handlar det om att ge instruktioner på ett begripligt sätt:



1. Sätt ugnen på 175 grader.
2. Vispa ägg och socker poröst.
3. Blanda vetemjöl och bakpulver.

Och så vidare – till slut blir det en sockerkaka, om receptet följs till punkt och pricka.

Till skillnad från en van bagare kan en dator däremot inte på egen hand upptäcka misstag i receptet/koden. Eftersom en dator bara gör exakt vad den blivit tillsagd skulle den försöka grädda kakan i 75 grader om ettan trillat bort i kokbokens manus. Dessutom är det viktigt att skriva instruktionerna i precis den ordning som datorn ska utföra dem. Datorn förstår inte att det blir svårt att vispa socker och ägg poröst om alla ingredienser redan är nedhållda i bunken.

När man programmerar skriver man en instruktion, ett recept, som datorn sedan följer. I datorernas värld kallas dessa recept för källkod.

Fördelen är att många av de koncept som används även i de enklaste, barnanpassade programmeringsverktygen är gemensamma med dem som yrkesverksamma programmerare använder sig av. (Se Programmeringsparlören på sidan 18)

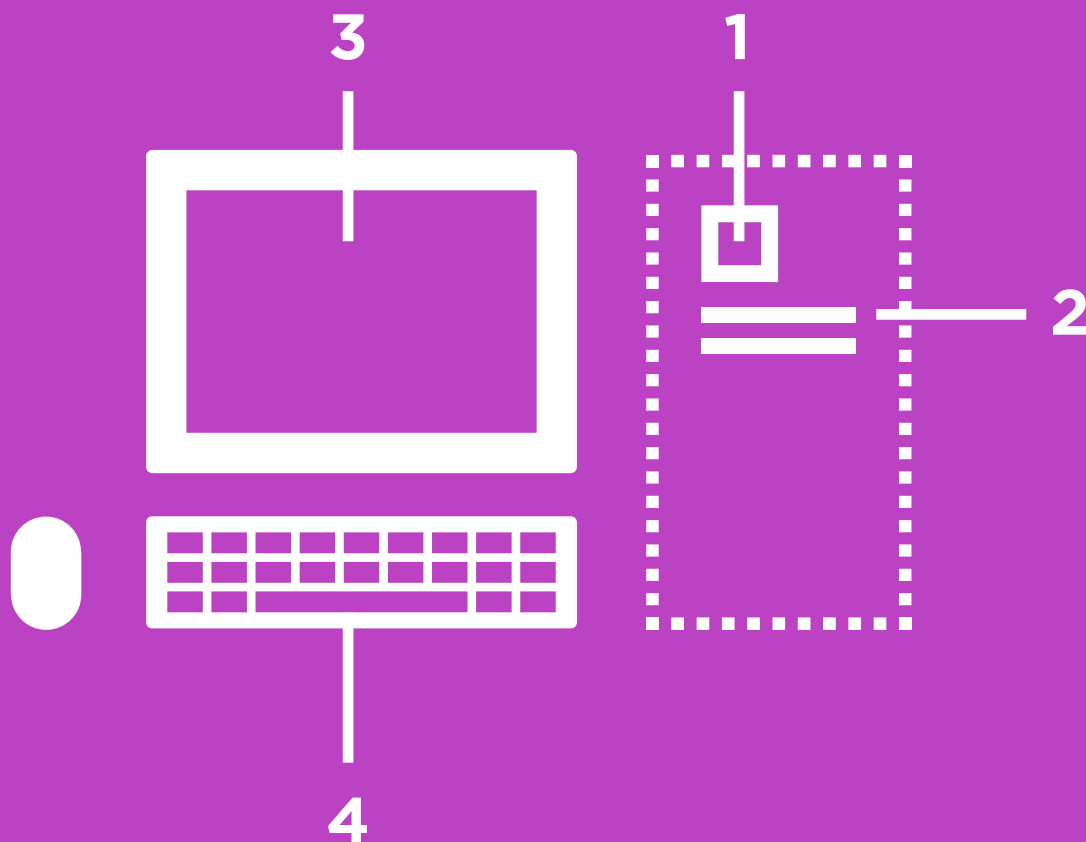
Genom att ge er i kast med programmering kommer du och barnen få helt nya kunskaper. För vissa läggs kanske grunden för en framtida programmeringsutbildning, medan andra får med sig en självupplevd insikt om att det är vi människor som styr tekniken. Programmering kräver även logik, planering och struktur, vilket är förmågor som är användbara också i andra sammanhang.

## Viktigt! Det här är en dator

Väldigt förenklat kan man säga att en dator – antingen i form av en dator som vi traditionellt tänker på dem, i form av en mobiltelefon eller något annat som är "datoriserat" – består av fyra delar:

1. En processor. Processorn fungerar som datorns hjärna. Det är processorn som utför instruktionerna en människa ger den.
2. Ett minne. Minnet används dels för att lagra instruktioner i form av datorprogram, men också data (information) som exempelvis textdokument, bilder och musik.
3. Delar för att presentera data, exempelvis en skärm eller en högtalare.
4. Delar för att samla in data. För en traditionell dator kan det exempelvis handla om en mus och ett tangentbord, medan en modern mobiltelefon har en skärm som det går att peka på.

Processorn bearbetar bara ettor och nollor i form av något som kallas för binära tal. Därför omvandlas också all information i en dator till ettor och nollor, exempelvis när du rör datorns mus.



### Tips!

Färdigheter barnen tränar medan de lär sig programmera:

- Språk, för att skriva dialog eller instruktioner
- Matematik och logiskt tänkande i själva programmeringen
- Estetiska förmågor, när ljud och bild ska skapas

Dessutom kan de nyvunna kunskaperna om hur ett datorprogram fungerar ligga till grund för en diskussion om teknik och demokrati.



## 2.2 Våga hoppa med barnen

Att du inte själv kan programmera behöver inte vara något problem. På internet finns gott om resurser som lär ut grunderna och IIS publicerar löpande utbildningsmaterial. Du hittar allt samlat på [barnhack.se](http://barnhack.se).

Det finns alltså goda förutsättningar för att hoppa tillsammans med barnen. Men var beredd på att deras förväntningar kan vara högt ställda ("Jag vill göra ett eget Super Mario!") och tålmodet för långsamma framsteg lågt ("När ska gubben börja röra sig egentligen?"). En bra idé är därför att ta de första stapplande stegen på egen hand och ha en tydlig idé om vad som ska bli resultatet när du första gången sätter dig vid datorn tillsammans med barnen. Det går förhållandevis lätt att göra ett enkelt tennisspel med exempelvis Scratch. Utseendemässigt kommer det inte vara i närheten av de spel som barnen spelar på datorer och telefoner utan snarare spel från 1970- och 1980-talet där tennisspelarna representeras av smala rektanglar i var sina ändor av spelplanen. Men känslan av att ha gjort spelet själv brukar trots det vara häftig. Att varken Scratch eller Kojoser ut som spelen hemma i vardagsrummet eller i telefonen hjälper också till att lägga förväntningarna på en rimlig nivå.

## 2.3 Planera vad ni ska göra

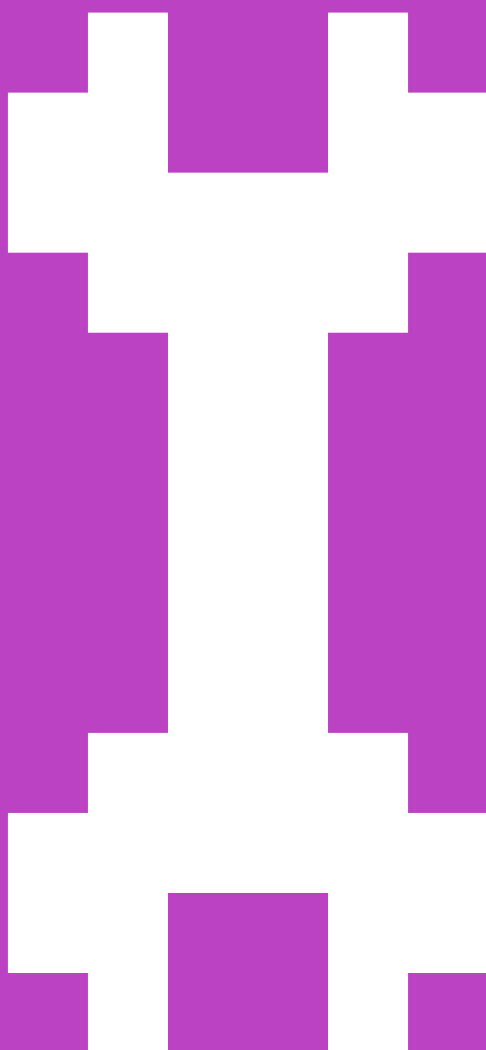
När ni tar de allra första stegen går det utmärkt att prova sig fram lite planlöst, att försöka få figuren på skärmen att gå till höger och vänster när piltangenterna trycks ner eller att lyckas rita upp en röd kvadrat på skärmen. Men så snart ni känner er redo för att ta nästa kliv, till exempel genom att göra ett enkelt spel, är det bra att först svara på några frågor:

- Vad går spelet ut på? Bestäm vad som är målet och hur spelaren ska nå dit.
- Vilka figurer och objekt ska vara med? Kanske en figur att styra, några fiender att fånga och ett hinder att undvika.
- Vad ska hända när de olika delarna möts? Just interaktionen mellan de olika delarna är en väldigt viktig del i spelet. Reglerna som ni sätter upp här avgör vad som kommer att hända och när den som spelar får poäng, vinner eller förlorar ett liv.

Med svaren på de här frågorna klara är det lättare att skrida till verket och bygga spelet i etapper. Först se till att figuren som spelaren ska styra rör sig som ni tänker, sedan lägga till andra objekt och slutligen programmera reglerna som ger poäng.



# 3. Verktygen



Antalet programmeringsspråk är stort. En del är nya och på uppgång. Andra har använts under många år och är på väg att bli omoderna. Däremellan finns de alternativ som är populära just nu. Diskussionerna om vilket språk barn och ungdomar bör lära sig blir lätt infekterade. Men då är utgångspunkten ofta att kunskaperna på sikt ska leda till en yrkeskarriär som programmerare och att det därför är viktigt att barnen lär sig ett programmeringsspråk som har framtiden för sig.

Om utgångspunkten i stället är att skaffa kunskaper för att bli en skicklig konsument och medveten medborgare är det viktiga inte vilket språk du och barnen använder utan att ni över huvud taget får testa att programmera. I stället kan valet göras utifrån andra premisser:

- En låg tröskel som gör det enkelt att komma igång.
- Möjlighet att nå snabba resultat, så att barnet tycker att det är roligt och vill lära sig mer.
- Gärna ett verktyg som bekantar er med vanliga programmeringsmetoder som upprepning och val. (Se Programmeringsparlören senare i detta kapitel.) Då spelar det mindre roll vilket programmeringsspråk som används. De grundläggande principerna är gemensamma och har man lärt sig dem i ett språk kan de tillämpas i andra.
- För de yngsta barnen som ännu inte lärt sig engelska är det självklart bra att välja ett språk som finns översatt till svenska. På det sättet river ni den mänskliga språkbarriären och kvar finns bara utmaningen att lära sig hur en dator fungerar. Eller helt enkelt börja med "programmeringslek" utan datorer: Till exempel kan en vuxen agera robot. Barnen får en uppgift som ska lösas med hjälp av roboten, som får sina instruktioner via papperslappar där barnen ritat eller skriver.

### **Scratch**

Ett grafiskt programmeringsverktyg som utvecklades vid det amerikanska universitetet MIT. I stället för att skriva kod för hand byggs programmen med hjälp av pusselbitsliknande block som alla har olika funktioner.

Scratch har flera fördelar för barn som vill lära sig att programmera. Både själva programmeringsmiljön och en del av hjälpmaterialet finns översatt till svenska. Det finns dessutom en uppsjö av resurser på engelska, både i form av skrivna instruktioner och videosnuttar på bland annat Youtube. (Om ni söker på "scratch programming tutorial" i stället för bara "scratch tutorial" slipper ni de flesta instruktionsfilmer för blivande discjockeys.)

### 3. Verktögen

Att Scratch går att använda i webbläsaren gör det också väldigt lämpligt som verktyg i ett barnhack där inte alla deltagare använder sin egen dator. Genom att skapa ett Scratch-konto kan de sedan fortsätta sina projekt vid vilken dator som helst.

På Scratchs webbplats går det att klicka sig runt bland spel som andra användare gjort. Och när man hittat något som verkar kul finns möjlighet att titta hur det är gjort men också att kopiera det och prova att ändra och förbättra.

#### **Kojo/Scala**

Kojo är en barnvänlig utvecklingsmiljö för programmeringsspråket Scala. Precis som Scratch är Kojo gratis och finns översatt till svenska. Men till skillnad från Scratch är Kojo riktig programmering, där man skriver riktig kod. För nybörjare innebär det bland annat att en sköldpadda som styrs med kommandon som *\*fram\**, *\*vänster\**, *\*cirkel\** och *\*färg\** kan få att rita olika mönster på skärmen.

Den svenska översättningen är gjord av Björn Regnell, professor i datavetenskap vid Lunds Universitet. Han förklarar sin syn på verktyget och varför han översätter Kojo till svenska så här:

– Jag vill att det ska vara enkelt att komma igång. Men jag vill inte att det ska finnas ett tak, jag vill att det ska vara på riktigt från början. Jag vill att man ska ha tillgång till alla verktyg när man är mogen för dem, utan att behöva byta till ett annat språk. Hela Formel 1-bilens kraft finns under huven.

På sin webbplats har han mer material och länkar till fler källor. Se länklista på sidan 45.

#### **Minecraft**

Minecraft är i första hand ett datorspel. Men det skulle också kunna kallas för ett digitalt lekland. Det finns inga förutbestämda uppdrag. Istället gör spelarna det de har lust med.

För den som vill finns möjlighet att skapa logiska kretsar, vilket är datorkunskap på riktigt grundläggande nivå. En sökning på "minecraft circuits" ger en uppsjö av förslag på hur. Kretsarna kan till exempel användas för att bygga ett system av strömbrytare som endast öppnar en dörr när alla strömbrytarna står i rätt position.

Minecraft går också att bygga ut. Med så kallade moddar, från engelskans modifications, kan spelet få nya funktioner.

Computercraft är en Minecraft-mod som gör det möjligt att bygga både programmerbara datorer och programmerbara robotar inne i spelet. Kul och med robotarna dessutom användbart. Genom att ge robotarna, som kallas för turtles, instruktioner i programspråket Lua, kan de bli användbara medhjälpare som till exempel kan ta hand om jordbruket i spelet. Rätt programmerad kan roboten både så och skörda säd.

Andra Minecraft-tillägg som kan vara värda att utforska är Feed the Beast, Buildcraft och Tekkit. Alla ger på olika sätt Minecraft nya funktioner. Feed the Beast är en Minecraft-värld där spelaren ska lösa olika uppdrag, och till hjälp finns en hel del teknisk utrustning, medan Buildcraft och Tekkit lägger till mängder av olika teknikrelaterade nyheter.

### 3.1 Andra verktyg

Börjar du leta runt på nätet kommer du att hitta massor av andra verktyg tänkta att lära barn att programmera. Kids Ruby är en barnanpassning av programmeringsspråket Ruby. Programmet man använder finns översatt till svenska, men till skillnad från Scratch och Kojo är kommandona fortfarande på engelska.

Till Apples prylar finns också en del programmeringsappar. Hopscotch gör det möjligt att utveckla spel i ett verktyg som påminner om Scratch, med block som kombineras på olika sätt. Move The Turtle är en annan iPad-app. Uppgiften är att lösa uppdrag genom att styra en sköldpadda med programmeringskommandon. En annan variant på nästan samma tema är Kodable. Mer avancerad spelprogrammering är möjlig med Codea.

### 3.2 Programmeringsparlören

Oavsett vilket programmeringsspråk ni väljer finns det vissa grundläggande byggstenar som återkommer. Och har ni en gång lärt er att använda dem på ett sätt är det förhållandevis enkelt att återanvända samma grundprinciper i andra språk:

**En loop** talar om för datorn att göra om samma sak minst två gånger, att den ska repetera ett eller flera kommandon, instruktioner. Att repetera kommandon är ett smart sätt att inte behöva skriva mer kod än nödvändigt. I stället för att skriva fram, vänster, fram, vänster, fram, vänster, fram, vänster för att rita en kvadrat kan du skriva upprepa 4 gånger: fram, vänster. (Nu finns det inget programspråk som skulle förstå exakt de instruktionerna, men det är så principen för loopar ser ut.)

**Variabler** är som lådor där ni kan stoppa in saker. Lådans innehåll är ett värde. I ett spel är antalet liv figuren har kvar och poängställningen två exempel på variabler. I liv-lådan minskar värdet när figuren krockar med en fiende och när värdet är noll är spelet slut.

För att datorn ska förstå hur ett värde kan användas har det en viss typ. Heltal kan till exempel användas i matematiska beräkningar, medan strängar är ord som "Hej".

### 3. Verktögen

Värden kan kombineras till sammansatta uttryck. Heltalen 2 och 3 kan summeras som  $2+3$ , medan strängarna "dator" och "kunskap" kan kombineras till ett ord, "dator" + "kunskap" blir datorkunskap.

**Funktioner** gör det möjligt att samla ihop ett antal instruktioner och utföra dem alla på en gång. Funktionen kvadrat kan tala om för datorn att rita ett rakt streck och sedan svänga vänster – och repetera, loopa, sekvensen fyra gånger. Det gör det möjligt att återvända kod som man själv eller andra tidigare har skrivit. Kvadrat-funktionen skulle i princip kunna se ut så här:

- kvadrat (sidlängd)
- upprepa fyra gånger
- fram (sidlängd)
- vänster

När datorn använder funktionen kvadrat ersätts sidlängd med ett värde. Är det 1000 blir kvadraten stor, är det 10 betydligt mindre. Sidlängd är vad man kallar en parameter. En parameter är ett värde som funktionen använder.

Funktioner slutar med ett resultat, i det här exemplet en kvadrat.

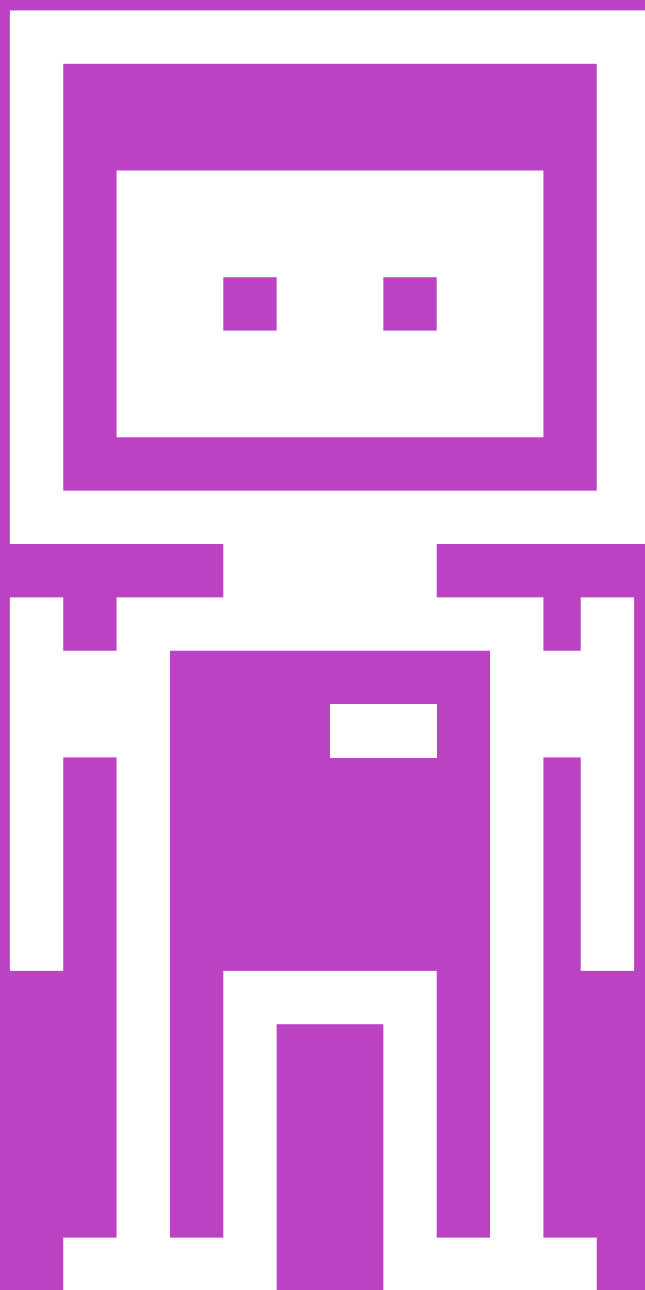
**Slumptal** är också viktiga i programmering, inte minst när man gör spel. Slumptal är datorns motsvarighet till ett tärningskast.

Väldigt ofta behöver man kontrollera om något har hänt, till exempel om spelaren tryckt ner en knapp för att styra sin spelare eller om bollen träffat en vägg. Dessa kontroller utgör alternativ i programkoden, ofta i form av så kallade **if-satser**. Uttryckt på svenska är följande fyra exempel på if-satser:

- Om vänster piltangent är nedtryckt, flytta till vänster.
- Om höger piltangent är nedtryckt, flytta till höger.
- Om mellanslag är nedtryckt, hoppa.
- Om bollen passerar mållinjen, öka variabeln Mål med ett.

Väldigt viktigt i programmering är också i vilken ordning datorn får dina kommandon. Ska du styra en figur ger vrid dig åt vänster, fram, fram, fram och fram, fram, vrid dig åt vänster, fram två helt olika resultat.

# 4. Fysiska hack



Hackandet måste inte ske instängt i en dator. Det går också att hacka i den fysiska världen runt datorn med hjälp av övningar och lekar i rummet. Antingen som en introduktion till programmeringens principer, som ett steg innan ni börjar med Scratch, Kojo eller något av de andra verktygen, eller som påbyggnad när ni tagit de första stegen med ett programmeringsspråk.

### 4.1 En introduktion till programmering

Det finns flera möjligheter att "leka" programmering och lära sig en del av grundprinciperna i Programmeringsparlören på sidan 18 och liknande. Ge sedan barnen i uppdrag att lösa en uppgift med hjälp av en robot (leksaker eller egentillverkad) och de instruktioner som finns tillgängliga. Ett förslag är att ta sig från ena änden av ett rum till den andra utan att krocka med något av hindren som finns utplacerade. Det går också att introducera koncept som variabler med hjälp av liv som kan förloras eller poäng som ska plockas upp.

För den som hellre spelar spel på köksbordet finns RoboRally. I spelet tävlar upp till åtta deltagare om att först lyckas ta sin robot till spelplanens alla checkpunkter. I övrigt är upplägget likt förslaget här ovan. Spelet är inte enkelt att hitta i handeln men går att köpa på nätet.

Ett annat alternativ som ger en fysisk introduktion till programmering är Littlebits. Littlebits är en serie magnetiska byggklossar med olika funktioner. Bitar som är blå eller orange är de minst intressanta. Blått är strömförsörjning, orange är förlängningssladdar. I stället är det de rosa och gröna bitarna som väcker mest intresse.

De rosa är sensorer och strömbrytare av olika slag. Här finns allt från vanliga tryckknappar till sådana som reagerar på ljus och ljud. De gröna är motorer, fläktar och olika ljuskällor – bitarna som ger konstruktionen dess slutliga funktion.

Tack vare att kopplingen mellan de olika delarna består av magneter är det inga problem för barn att själva koppla ihop och testa vad som händer. Och med lite ledande frågor som hjälp på vägen går det snart att bygga saker som gör det de vill. Den "fysiska programmering" som är möjlig med Littlebits är därmed ett sätt att upptäcka att tekniken runt omkring oss är konstruerad av människor och att vi har möjlighet att kontrollera den på olika sätt.

Nackdelen med Littlebits är att det, precis som med Lego, krävs många delar för att kunna bygga roliga saker och att kostnaden därför snart blir ganska stor. Samtidigt är möjligheterna begränsade eftersom det inte går att finjustera hur olika bitar ska fungera.

## 4.2 En vidareutveckling

När ni lärt er grunderna i programmering finns det stora möjligheter att gå vidare i den fysiska världen. Har ni gjort spel i Scratch kan ett första alternativ vara att bygga en handkontroll med hjälp av Makeymakey. Makeymakey är ett enkelt tangentbord med usb-anslutning där vad som helst kan användas som tangenter så länge det leder ström. Namnet är helt enkelt en sammanslagning av de engelska orden "make" och "key", gör tangent.

Mer spännande blir det om ni börjar utforska möjligheterna med Arduino. Arduino är en så kallad mikrokontroller, ett styrkort som kan användas för att samla in information från sensorer och för att styra olika typer av reglage, som motorer, skärmar och ljusdioder, som alla är kopplade till kortet. Likheter med Littlebits är på sätt och vis många, men möjligheterna oändligt mycket större eftersom det går att kontrollera vad som ska hända på helt andra sätt och utbudet av både sensorer och reglage är mycket större.

Arduinon programmeras med hjälp av ett språk som är baserat på C och C++, två programspråk som används vid professionell utveckling. Det innebär att utvecklingen sker på engelska och all kod skrivs för hand. Tröskeln är alltså relativt hög, eftersom det både krävs investering i hårdvara och vissa programmeringskunskaper. För den som vill komma igång finns dock färdiga startpaket som innehåller både Arduino, ett urval sensorer och reglage samt instruktioner för att bygga olika saker med dem.

För den som är beredd att lägga lite mer pengar finns också Lego Mindstorms, populärt kallat robotlego. Ett startpaket kostar några tusenlappar. Det innehåller bland annat en mikrokontroller, sensorer och motorer.



# 5. Så hackar vi - tillsammans med barnen



Utgångspunkten för den här Internetguiden med tillhörande utbildningsmaterial är att programmering är ett sätt att låta barn skaffa sig viktig kunskap om hur datorer och internet fungerar. Många barn är redan duktiga användare av prylar och tjänster. Men att kunna använda en dator är inte samma sak som att förstå hur den fungerar eller att det är människor som fattat alla beslut.

– Det går inte lära sig kemi genom att äta medicin, inte heller lära sig datavetenskap genom att använda färdiga program och tjänster. Att äta medicin och använda en dator är att använda en tillämpning, att lära sig kemi och datavetenskap är att förstå hur, säger Thore Husfeldt.

Barn som är med i IIS skolsatsning Webbstjärnan fick under våren 2013 svara på två frågor: Vad är internet? och Hur fungerar internet?

Som du själv kan läsa i sammanställningen IIS frågar barn om internet – Vad är och hur fungerar internet? är spännvidden i svaren på den första frågan stor, men kretsar ofta kring hur de använder nätet. Barnen lyfter fram informationssökning, digitalt umgänge med kompisar, spel, bloggar och videotjänster som exempel på vad internet är.

Svaren på den andra frågan är många gånger betydligt mer trevande:

– Jag vet faktiskt inte riktigt, det är så mycket som man inte vet och mycket som händer ”bakom scenerna”, svarar en trettonåring. Andra nämner behovet av elektricitet och att datorer är inblandade. Kablar och signaler är också viktiga tror några av barnen. Och alla de svaren ligger nära sanningen.

Någon tror att internet är kommunens ansvar, andra att hela nätet sköts av Google. Men det finns också de som lutar åt övernaturliga väsen: ”Internet fungerar med teknologi, som magi. Jag tror nästan inte det är någon som riktigt vet hur det fungerar”, svarar en elvaåring. ”Jag tror att internet fungerar med magi som när till exempel en trollkarl befinner sig någonstans och sedan är på någon helt annanstans”, gissar en annan. En tolvåring svarar att han aldrig tänkt på frågan tidigare, men gissar att det kanske finns robotar som fixar allt.

Vi tror att det går att avmystifiera tekniken genom att lära barnen att programmera. Får de själva vara med och ge en dator instruktioner och se hur datorn lyder dem blir det också tydligt att det är ett formligt material på samma sätt som de material och den utrustning som finns i skolans slöjd- och kemisal. Dessutom ger det en insikt om hur olika delar hänger ihop, en systemförståelse. Att vissa saker händer i och finns i barnets egen dator medan annat finns och händer på en server någonstans på nätet.

Det här är kunskap som blir allt viktigare att ha. Det handlar om förmågan att navigera i ett digitaliserat samhälle och att förstå

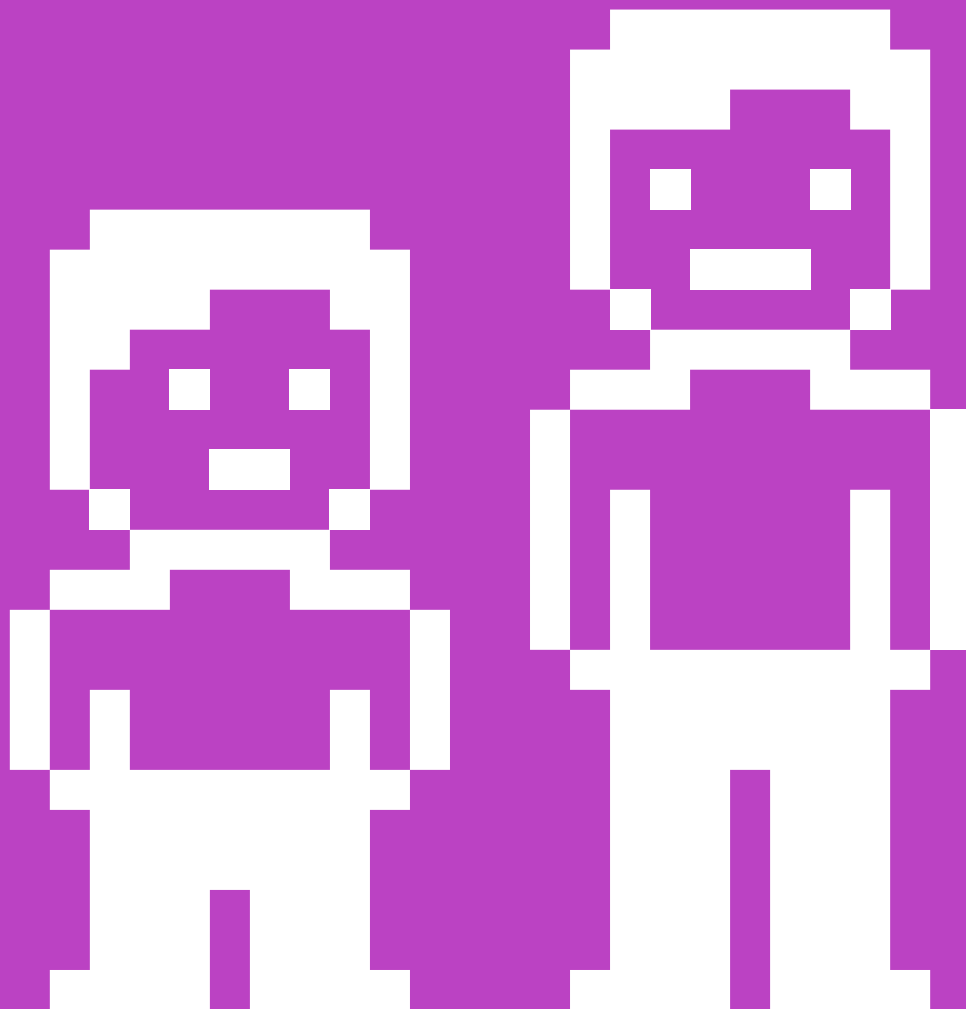
## 5. Så hackar vi – tillsammans med barnen

att de val och beslut som fattas av datorer i själva verket är val och beslut som görs av andra människor. Datorer är inte smarta – människor som programmerar dem är smarta.

Men att lära barn att programmera hjälper dem inte bara att bättre förstå datorer och internet. Lärare som jobbar med programmering i lägre årskurser, på låg- och mellanstadiet, berättar om hur programmeringslektionerna också tränar helt andra färdigheter än de som är direkt kopplade till tekniken. Problemlösning och språk är två exempel. Problemlösning när en uppgift ska klaras av eller när en egen spelidé ska omsättas i praktiken, språk när instruktioner ska skrivas så att andra barn kan spela spelet eller lära sig att själva göra ett likadant.

Att som barn få lära sig att göra egna spel är också att ha kul i nuet. Att ta sina tankar från idé till färdigt spel kan ge en enorm kick. Att visa upp sitt spel för andra ger också stor glädje, liksom möjligheten att lära andra och att samarbeta för att tillsammans bli duktigare.

# 6. Så hackar vi – tips för föräldrar (eller mostrar)



## 6. Så hackar vi – tips för föräldrar (eller mostrar)

Förhoppningsvis är du vid det här laget övertygad om varför det är en bra idé att barnen i din närhet får prova på att programmera. Men kanske känner du dig lite uppgiven – om du inte kan programmera själv kan det kännas som en övermäktig uppgift att lära en tioåring göra det.

Men det behöver inte alls vara omöjligt. Se det i stället som en gemensam utmaning för er att tillsammans utforska datorernas värld och lära er en del om hur de fungerar. De verktyg som vi tipsar om i den här guiden, med tillhörande kursmaterial, är lätta att komma igång med och ger snabba resultat.

– Det handlar om att ta sig tid, säger Tomas Ribba, pedagogiskt ansvarig på Tekniska museet i Stockholm, och fortsätter:

– Barnet och du har samma förutsättningar, ni börjar från noll. Det kan vara kul att utforska tillsammans, och som vuxen har du förmågan att läsa och förstå instruktioner.

Men om du själv inte kan programmera är det bra om du själv först skaffar dig lite grundkunskaper på egen hand. Det tar inte mer än några timmar och gör det sedan enklare att fånga barnets nyfikenhet. Om både du och barnen börjar från noll när ni tillsammans sätter er vid datorn är risken stor att intresset snabbt försvinner när alla resultat uteblir i väntan på att du ska läsa instruktioner.

Bestäm också i förväg vad du och barnet ska göra de första gångerna ni programmerar tillsammans. Planlöst utforskande är kul ibland, men kräver att ni har en grund att stå på. Ett tydligt mål är bättre vid datorn på samma sätt som med kriterior och ett vitt papper: ”Nu ska jag rita ett slott!”

Fundera därför i förväg på problem som ni ska lösa tillsammans. Vilka problem som innebär rimliga utmaningar beror både på vilket verktyg du bestämt att ni ska använda och hur mycket ni kan. Viktigt är också att du själv, åtminstone i början, vet hur de ska lösas.

Det enkla Scratch-spel som vi föreslår att en kursledare kan använda på ett barnhack med flera deltagare (se sidan 31) är en utmärkt startpunkt också hemma vid köksbordet. Följ först instruktionerna på egen hand och lär dig hur det är uppbyggt. Fördelen är att det är ett ”riktigt” spel där det räknas poäng och barnet kan vinna, men också att det bakom kulisserna är enkelt och förhoppningsvis möjligt för er att förstå vad som händer och varför. Gör sedan om allt tillsammans med barnet – och försök därefter att utveckla spelet vidare. Lägg till en klocka som mäter hur lång tid det tar att nå tio poäng. Eller vänd på det, låt spelaren spela i en minut och se hur många poäng hen får.

Ett annat viktigt råd är annars att lägga ribban på en rimlig nivå, att sänka både dina egna och barnets förväntningar. Ni kommer inte att göra ett nytt Minecraft. Åtminstone inte än på några år. I stället är det betydligt enklare spel som gäller. Men våra erfarenheter är

att det smäller högt att få spela sina egna skapelser även när de är betydligt enklare än spelen barnen brukar spela.

Hoppa sedan tillsammans och ha inställningen att de utmaningar som ni stöter på har en lösning och att ni tillsammans ska hitta den. Eller lösningar, rättare sagt. När man programmerar går ett problem ofta att ta sig an på många olika sätt.

– När barnet efter ett tag springer före dig i kunnande – häng på: Låt tioåringen visa vad hon har gjort, låt henne förklara hur och provspela resultatet, säger Tomas Ribba.

## Tips! Missa inte kursmaterialet!

Till den här guiden finns det kursmaterial i form av utförliga steg-för-steg-instruktioner, *Kom igång med Scratch!* av Måns Jonasson.



Du kan också hjälpa barnet att starta en blogg där hen kan visa sina spel för kompisarna och få dem intresserade också. Några steg på vägen finns i kapitlet Så sprider du barnhackandet.

Om du upptäcker att du lärt dig tillräckligt mycket för att lära dina egna barn att programmera, överväg att arrangera ett eget barnhack. Antingen på egen hand tillsammans med några av barnens kompisar, eller i ett större sammanhang i samarbete med exempelvis fritids eller ett studieförbund.

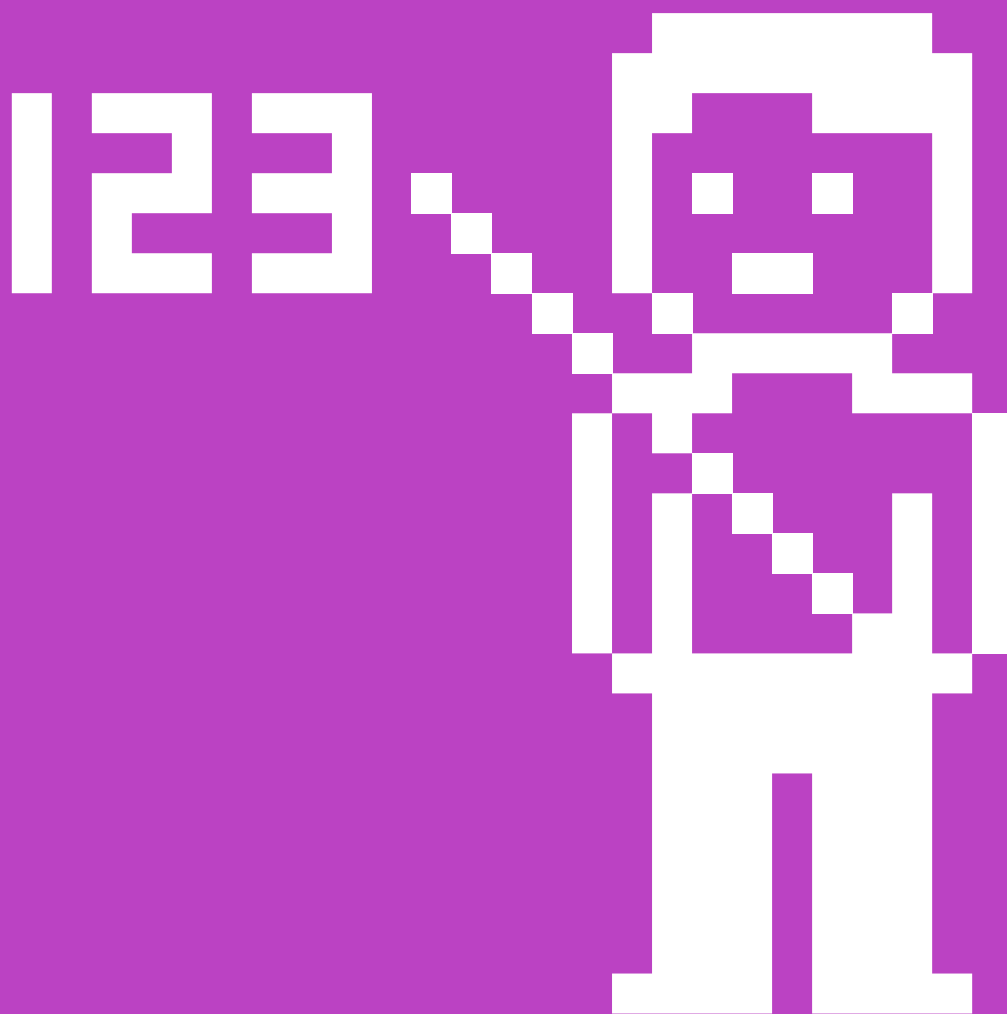
Avslutningsvis några ord om en fråga som ofta förekommer kring föräldraskap och datorer: Den om skärmtid. Hur mycket och hur ofta ska barnen få sitta vid datorn? Vi har inget svar på den frågan

## 6. Så hackar vi – tips för föräldrar (eller mostrar)

eftersom den sannolikt inte har något universellt svar. Varje barn har sina unika förutsättningar och de eventuella regler som behöver ställas upp måste vara anpassade till dem och inte till vad som gäller för kompisarna.

Men vi vill påminna dig som vuxen om att all tid vid en skärm inte är identisk. Själv använder du säkert datorn för att betala räkningar, för att söka efter matrecept och för att hålla kontakten med släkt och vänner. En dator, surfplatta eller mobiltelefon går att använda till en mängd olika saker där den enda gemensamma nämnaren är just tekniken. Som ansvarig vuxen kan det finnas anledning att inte bara fundera över hur mycket tid som tillbringas vid en skärm utan också till vad den tiden används. Kanske är det inte skärmtiden som ska begränsas, utan vissa specifika aktiviteter? Fundera på om det är rimligt att begränsa all datoranvändning om det i själva verket är ett specifikt spel som är den stora tids- och fokustjuven.

# 7. Så hackar vi - kursledaren





## 7. Så hackar vi – kursledaren

När du har tillräckliga kunskaper i programmering är det kanske dags att leda ett barnhack. Börja med att lära dig verktyget som du ska använda tillsammans med barnen. Även om du inte har tidigare erfarenheter av Scratch, Kojo eller något av de andra alternativen kommer det att gå snabbt för dig att lära dig grunderna. Men gör det innan du ställer dig framför barnen, så att du kan leverera snabba svar på de mest grundläggande frågorna.

Men som en del i barnhackandet är det också bra att sträva efter att skapa en miljö som tillåter experimenterande och att saker och ting inte blir som man tänkt från början.

– 8-12-åringar är helt orädda för misslyckandet – om miljön tillåter och uppmuntrar att man misslyckas, att man ser det som möjligheter att lära sig något nytt, säger Carl Heath, forskare på Interactive Institute Swedish ICT.

Därför är det inte nödvändigt att du har svar på alla frågor som kan dyka upp. Det går inte att förutse vilken väg deltagarnas projekt kommer att ta och därför inte heller att förutse vilka programmeringsutmaningar de kommer att stöta på. I stället får du och deltagarna tillsammans försöka lösa de frågeställningar som dyker upp under barnhackets gång.

Innan ni dyker ner i datorerna kan det dock vara en bra idé att köra lite ”analog programmering”. Innan alla deltagarna riktar sin uppmärksamhet mot sina skärmar, gå igenom de mest grundläggande principerna (se Programmeringsparlör på sidan 18), i teorin. Scratch lämpar sig särskilt väl för detta, då det är en programmeringsmiljö som är väldigt visuell. IIS har skapat pdf-dokument med några av de mest använda blocken i Scratch. Med dem som utgångspunkt går det att förklara bland annat repetitioner och alternativ.

För att hjälpa barnen igång är det bra att i förväg skapa några enkla övningar att starta med. Risker är annars att barnen antingen baxnar inför alla möjligheter och inte kan bestämma vad de ska göra, alternativt att de lägger ribban alldeles för högt.

Är det spelutveckling i Scratch som står på schemat är exempelvis ett lämpligt första steg att lära sig att styra figuren på skärmen med piltangenterna. Det är en väldigt avgränsad utmaning som har en tydlig och relativt enkel lösning och som dessutom ger kunskap som är till glädje i det fortsatta experimenterandet. Lös problemet tillsammans i gruppen, du som kursledare på projektorns storbild medan deltagarna härmar det du gör på sina egna datorer.

Därefter är det dags att ta nästa steg – vilket med fördel kan vara att barnen jobbar vidare på egen hand. Din roll blir då att hjälpa dem som kör fast. Men glöm inte att uppmuntra barnen att hjälpa varandra. De lär sig mycket av att lösa problem tillsammans. Dessutom gör det att kursledarens kunskap snabbare sprids i gruppen.

Beroende på hur länge barnhacket håller på kan det vara ett bra alternativ att lägga in ett mellansteg mellan den gemensamma utmaningen och det egna utforskandet: En stunds planering.

Låt barnen lämna datorerna en stund och fundera på vad de vill göra. Vad ska målet med deras spel vara, vilka figurer ska vara med, hur ska de styras, när får man poäng eller förlorar liv och när vinner eller förlorar man? Låt barnen sedan presentera idéerna för varandra och flytta slutligen om i lokalen. Placera de barn som har liknande spelidéer bredvid varandra. De ska fortfarande jobba på egen hand, men liknande spelidéer innebär också liknande problem som ska lösas. Att skapa barngrupper utifrån deras idéer underlättar därmed möjligheten för samarbete och kunskapsdelande.

Ett annat alternativ är att förbereda ett par övningar som barnen får försöka lösa på egen hand. Om ni använder Scratch, ge dem i uppdrag att göra ett enkelt spel.

När barnhacket går mot sitt slut, glöm inte att låta barnen visa varandra – och omvärlden – vad de gjort. Spel gjorda i Scratch är enkla att publicera på webbplatser, så varför inte skapa en spelsamling på webben? (Se kapitel 9, Så sprider du barnhackandet)

Om tid finns kan det också vara en bra idé att inte bara titta på slutresultaten utan också på koden bakom. Det kommer då att bli uppenbart för barnen att samma problem går att lösa på många olika sätt. Vissa är korta och effektiva, andra innehåller många rader kod vilket gör det svårt att hitta fel. Sporra barnen att försöka göra så enkla lösningar som möjligt.

## 7.1 Ett Scratch-spel i fem steg

Ett förslag till ett förberett uppdrag om ni använder Scratch på barnhacket.

### **Steg 1 – Lär katten gå**

Ge katten instruktioner att gå 10 steg framåt för varje klick på den gröna startflaggan.

### **Steg 2 – Katten går själv**

Efter ett klick på startflaggan ska katten gå själv, vända när den kommer till kanten och gå tillbaka åt andra hållet där den återigen vänder vid kanten.

### **Steg 3 – Poäng för kattklick**

Dags att lägga till spelmomentet. För varje gång spelaren lyckas klicka på katten med muspekaren ska hen få en poäng.

### **Steg 4 – Minus för bom**

Nu måste det bli lite svårare att samla poäng. Om spelaren missar katten med sitt musklick, dra bort ett från poängställningen.

### **Steg 5 – När vinner man?**

När spelaren fått tio poäng ska katten säga ”Du vann!” och spelet stanna. När det startas igen ska poängställningen nollställas.

Lösningar på alla fem utmaningarna finns på <http://scratch.mit.edu/studios/261532/>.

## 7.2 Övningar i Kojo

Använder ni er av Kojo är några förslag till enkla övningar:

Rita ett trappa.

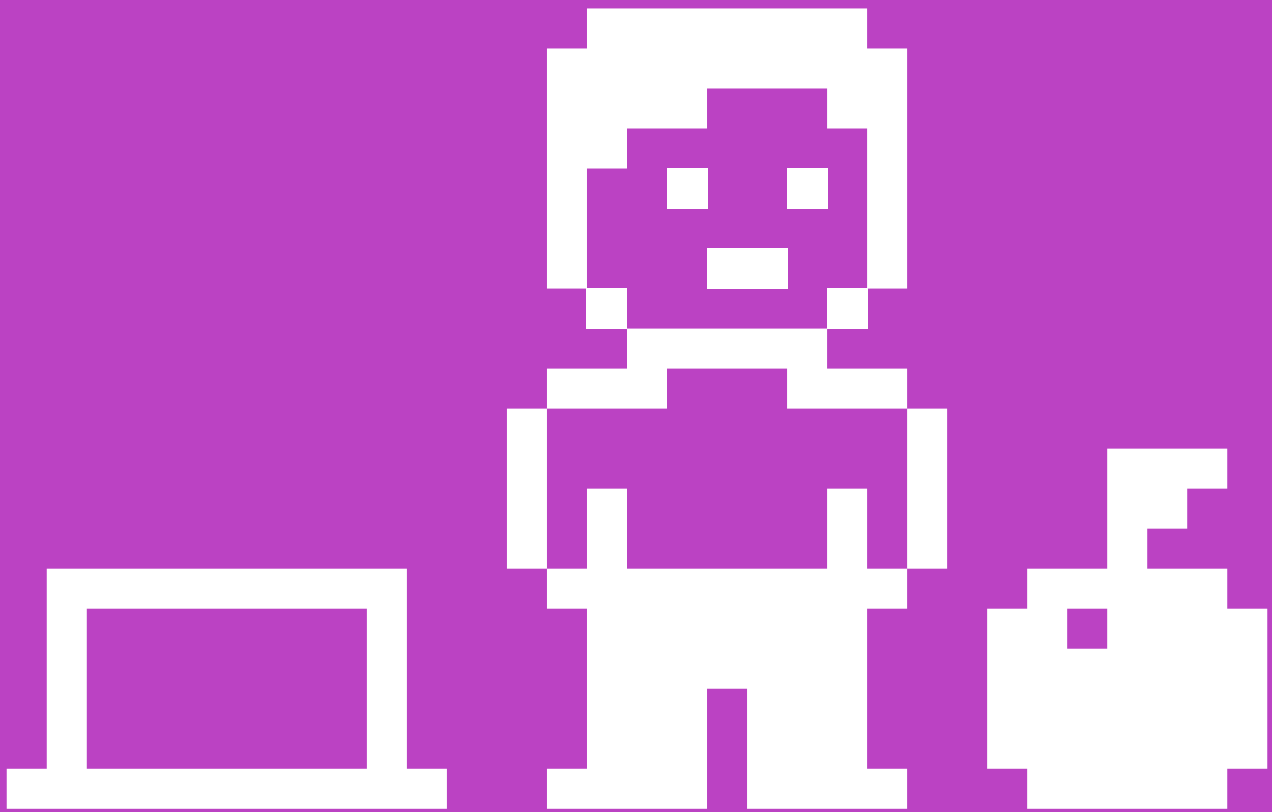
Rita en stege.

Rita ett djur.



Katten Scratch är programmets symbol, men är även ett exempel på sprites.

# 8. Så hackar vi - arrangören



Att arrangera ett barnhack för mer än ett par barn kräver förberedelser. Det är inga komplicerade saker, men en hel del att tänka på och komma ihåg.

### 8.1 Hitta en kursledare

Utan ledare – inget barnhack. Fundera på om du själv har lyckats skaffa dig tillräckliga kunskaper för att kunna lära barnen de enklaste grunderna och sedan gå vidare tillsammans med dem. Om du själv inte känner dig mogen att ta dig an uppgiften ännu, fråga runt bland dem du känner. Kanske finns det någon som kan eller kan tipsa om någon. Ett annat alternativ kan vara att höra dig för hos lokala it-företag.

Bra är också om kursledaren kan avlastas av ytterligare en eller två personer som kan hjälpa till med inloggning på det trådlösa nätverket och annat teknikstrul som dyker upp. Försök över huvud taget att ha en beredskap för teknikproblem. En alternativ internetuppkoppling kan till exempel vara bra att ha i bakfickan i form av en 4G-router, gärna någon dator extra också.

Om deltagarna har med sig egna datorer är ett alternativ att be dem installera Scratch. Scratch går nämligen att använda både från webbläsaren eller som ett program som installeras i datorn. Fördelen med webbversionen är att det går att sitta vid vilken dator som helst – barnets projekt finns tillgängligt så snart de loggat in.

### **Checklista!** För ett lyckat barnhack krävs följande ingredienser:

- En kursledare
- Gärna några medhjälpare
- Deltagare
- En lokal
- Internetuppkopplade datorer
- Programmeringsverktyg
- En projektor
- Förfriskningar
- Tillstånd att fotografera
- Ett schema



Men webbversionen kräver givetvis en fungerande internetuppkoppling. Om deltagarna i förväg har installerat programversionen av Scratch i sina datorer kan ni därför köra på även om det blir problem med internetuppkopplingen.

## 8.2 Locka deltagare

Men en ledare gör inget barnhack. Barnen är de viktigaste deltagarna! Prata med barnens kompisar och deras föräldrar, starta en webbplats (se kapitlet Så sprider du barnhacket) och komplettera med en grupp för föräldrar på Facebook. På nätet kan du vara mer utförlig om vad ett barnhack är, vad deltagarna kommer att få lära sig och vilka förkunskapskrav som behövs.

En nedre åldersgräns på 6–7 år kan vara bra eftersom barnen behöver kunna läsa hjälpligt. Beroende på lokalens storlek kan det också finnas en gräns för hur många barn som kan få vara med. En annan begränsande faktor är kursledarens möjlighet att kunna hjälpa till när barnen jobbar på egen hand: 14–15 barn kan vara ett lämpligt tak. Det är också bra om varje deltagare har med sig en vuxen som kan hjälpa till. Utan andra vuxna i lokalen är risken allt för stor att för mycket av kursledarens tid går åt till att ge individuell hjälp som i första hand inte handlar om programmering utan hur en webbadress eller ett kommando ska stavas eller var ett visst menyalternativ finns. En annan erfarenhet är att barnen blir väldigt ivriga och börjar experimentera på egen hand direkt. Egentligen är det en bra sak, men missar någon kursledarens genomgång kommer hen sedan att få lägga mycket tid på individuell hjälp. Därför kan det vara bra med en vuxen som bromsar barnen lite under genomgångarna.

Om för många barn anmäler sig, överväg att göra plats för de tjejer som vill vara med. Tyvärr är killarna fortfarande överrepresenterade i tekniksammanhang. Att göra plats för de tjejer som vill vara med på ett barnhack kan förhoppningsvis skapa ringar på vattnet, när de i sin tur kan få sina tjejkompisar intresserade.

Anmälningar går att ta in på många olika sätt. Enklast är givetvis via e-post, men att använda tjänster som Google Drive eller Wufoo för att skapa anmälningsformulär på webben är andra alternativ.

## 8.3 En plats att vara på

Om gruppen inte innehåller mer än några få barn går det nästan att hålla till var som helst. Det behöver inte vara mer komplicerat än ett köksbord i någons hem. Men om det ska bli ett större barnhack är det bra med en projektor och även en whiteboard.

Om du inte har tillgång till en bra lokal, fråga runt i bekantskapskretsen eller bland kollegor på jobbet. Många konferensrum och liknande lokaler står tomma på kvällar och helger.

## 8.4 Internetuppkopplade datorer

En del barn kan säkert ha med en bärbar dator hemifrån, men inte alla. Är det möjligt att använda en lokal där det finns datorer att låna är det självklart att föredra. Om barnhacket ska använda programmeringsverktyget Scratch krävs inte att några program installeras eftersom verktyget körs i webbläsaren.

Finns det inte internetuppkoppling i lokalen går det att lösa med en bärbar 4G-router.

## 8.5 Välj verktyg

Vilket programmeringsverktyg som ska användas beror till stor del på vilka verktyg kursledaren behärskar. Men eftersom ambitionen är att sänka tröskeln för dem som vill lära sig programmera är det bra att välja ett som finns översatt till svenska. Scratch är ett alternativ som ger snabbt resultat medan Kojo mer liknar ”riktig” programmering.

Meddela deltagarna vilka verktyg som ska användas och uppmana dem att i förväg testköra/installera det på sina datorer så att ingen tid måste ägnas åt detta vid kurstillfället. Som kursledare är det frustrerande att behöva felsöka varför Scratch inte fungerar i någon av deltagarnas webbläsare.

## 8.6 Projektor till kursledaren

Om det är ett större antal barn som deltar på barnhacket kan det vara omöjligt att trängas runt en skärm när kursledaren ska visa något. En projektor är lösningen på det problemet. Kontrollera i förväg att de adaptrar och kablar som krävs finns till hands.

## 8.7 Förfriskningar

Beroende på hur långt varje hacktillfälle är kan det behövas påfyllning av energidepåerna. Be antingen barnen packa en frukt i väskan eller ta själv med något gott att bjuda på.

## 8.8 Tillstånd att fotografera

Om du tänker fotografera under barnhacket, kanske för att publicera på webben, fråga om tillstånd. Det kan finnas barn som har skyddad identitet eller där föräldrarna av andra anledningar inte vill att bilder på barnen ska publiceras på nätet. Respektera det.

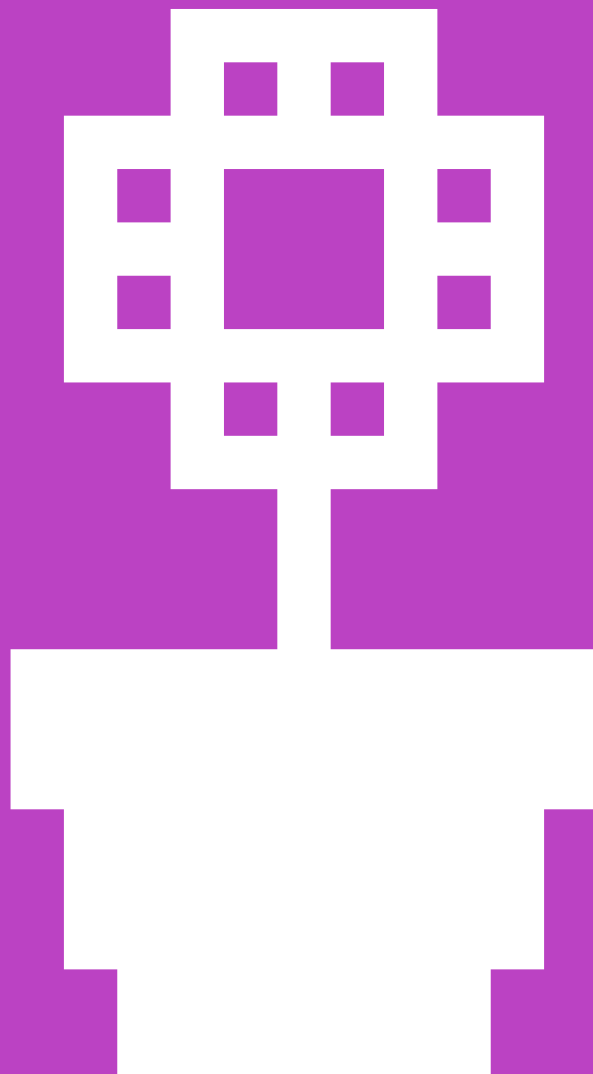
## 8.9 Ett schema

När ni planerar barnhacket behöver ni också bestämma hur länge ni ska hålla på och hur många gånger ni ska köra. IIS har arrangerat ett barnhack i form av en Scratch-introduktion vid ett flertal tillfällen. Kör ni en enstaka gång är erfarenheten att två timmar är lagom. Första timmen ägnas åt gemensamma uppgifter och efter en kort rast får deltagarna fortsätta på egen hand medan kursledaren kan gå runt bland datorerna och ge den hjälp som behövs.

Om ni bestämmer er för att köra barnhacket vid flera tillfällen är det ett utmärkt sätt att successivt introducera nya funktioner steg för steg. Förbered uppgifter till varje tillfälle! Första gången kan ni nöja er med att skapa en figur som går att styra, vid tillfälle två börja räkna poäng och så vidare.



# 9. Så sprider du barnhackandet



På internet kan du och dina barn sprida ert barnhackande på många olika sätt. En bra utgångspunkt är att skaffa en egen webbplats. Den kan ni använda för att visa upp vad ni gjort, för att lära och inspirera andra eller för att locka deltagare till ett större barnhack på orten.

## 9.1 Det här är en webbplats

Du brukar säkert använda en webbläsare för att surfa på webben. Men har du funderat på vad som händer bakom kulisserna? Ska du och dina barn bygga en egen webbplats behöver ni veta lite om det.

Webben består av ett gigantiskt antal webbservrar runt om i världen. De väntar på besökare i form av webbläsare. När du knappar in [www.iis.se](http://www.iis.se) i webbläsarens adressfält kontaktar den IIS webbservare och ställer en fråga. Översatt till mänskligt språk lyder den ungefär: "Kan du vara snäll och skicka mig förstasidan på din webbplats?"

Webbservern svarar med att skicka tillbaka webbsidan med allt innehåll i form av text, bild, video och annat. När webbläsaren tagit emot det visar den det på datorns eller telefonens skärm.

## 9.2 Vad behöver ni?

För att kunna skapa en webbplats behöver du och dina barn därför tre saker:

- En server där webbplatsen ska finnas.
- En adress till servern så att besökare kan hitta till den.
- En webbsida med text och bilder som servern kan skicka när någon vill surfa in på er webbplats.

Om detta känns förvillande är ett förslag att ni börjar med att skaffa en webbplats på [Wordpress.com](http://Wordpress.com). Wordpress är ett av världens mest använda webbpubliceringssystem. Det kan vara lika enkelt att använda som en ordbehandlare i datorn, men också bli hur avancerat som helst.

De som vill ha full kontroll på sin Wordpress-webbplats brukar installera verktyget på sin egen webbservare. Men på [Wordpress.com](http://Wordpress.com) finns allt ni behöver som färdig tjänst: Genom att registrera ett gratiskonto får ni tillgång till en server där er webbplats kan ligga, en adress och publiceringsverktyget Wordpress som gör det enkelt att skriva texter och visa bilder från ert hackande. Precis som på många andra gratistjänster på nätet finns också möjlighet att bli betalande kund. Vilka tillval som finns varierar över tid, men en möjlighet som funnits länge och sannolikt kommer att finnas under lång tid framöver är att kunna använda ett eget domännamn.

## 9. Så sprider du barnhackandet

Den som registrerar en webbplats på [Wordpress.com](http://Wordpress.com) får nöja sig med adresser som slutar just så, med [wordpress.com](http://wordpress.com). En tänkbar adress skulle därför kunna vara [barnhack.wordpress.com](http://barnhack.wordpress.com). Vill ni hellre ha en snyggare och enklare adress går det att registrera en egen domän, till exempel [barnhack.se](http://barnhack.se), och peka den till bloggen. Den som knappar in [barnhack.se](http://barnhack.se) skickas då automatiskt vidare till er blogg på [Wordpress.com](http://Wordpress.com).

En fördel med att välja [Wordpress.com](http://Wordpress.com) är att publiceringsverktyget finns översatt till svenska. En annan fördel jämfört med många andra publicerings- och bloggtjänster på webben är att ni i framtiden kan flytta er webbplats till en egen webbserver men ändå fortsätta använda Wordpress. Kanske blir barnen om ett tag nyfikna på webbprogrammering och webbutveckling, och då finns alla möjligheter att vidareutveckla bloggen ni redan startat utan att behöva flytta den från ett publiceringssystem till ett annat.

### 9.3 Kom igång på [Wordpress.com](http://Wordpress.com)

Att starta en webbplats på [Wordpress.com](http://Wordpress.com) går snabbt. Surfa till <http://sv.wordpress.com> för att få den svenska versionen. Klicka på Skapa webbplats. Välj mellan de olika ämneskategorierna och de teman som dyker upp. Webbplatsens utseende är beroende av det Wordpress kallar för teman, och de har alla olika layouter, typsnitt och liknande. Välj ett som du tycker ser bra ut. Blir du inte nöjd går det att byta längre fram.

I nästa steg kompletterar du med din e-postadress och väljer ett användarnamn och lösenord.

Därefter får du möjligheten att köpa ett eget domännamn till din nya webbplats. Välj Nej, tack om du inte vill göra det än.

I sista steget väljer du vilket "paket" du vill använda. Välj gratisvarianten så länge. Om det visar sig att du vill ha eller behöver någon av funktionerna som kostar pengar går det att ordna då.

Och till slut är det dags att skapa ditt första inlägg.

I kontrollpanelen kan du sedan administrera bloggen och publicera nya inlägg.

### 9.4 Ge barnhacket en egen mejladress

Som arrangör kan det vara enkelt att använda din befintliga e-postadress för att göra utskick och ta in anmälningar. Men det innebär också att barnhacket knyts väldigt mycket till dig som person. Överväg därför att ge barnhacket en egen e-postadress. Det innebär att du i framtiden kan lämna över ansvaret till någon annan eller att ni kan vara flera personer som hjälps åt att ta in anmälningar och svarar på frågor via e-post.

Registrerar ni ett eget domännamn är en bra adress info@, men det går givetvis bra att skapa en ny e-postadress på vilken e-posttjänst som helst.

## **Viktigt!** Publicera inför barnhacket

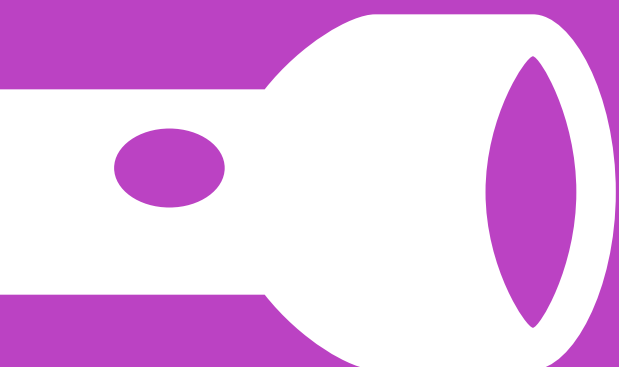
- Starta en egen webbplats, där ni presenterar vad ett barnhack är och visar vad deltagarna kommer att få lära sig.
- Se till att webbplatsen får besökare. Prata med barnens kompisar och deras föräldrar, sprid adressen i sociala medier.
- Publicera efter barnhacket
- Uppdatera webbplatsen med text, bilder och nya spel.
- Länka till webbplatsen i sociala medier.



## **9.5 Lär er mer om webbpublicering**

Vill ni lära er mer om webbpublicering finns flera Internetguider från IIS som är värda att titta närmare på men kan givetvis läsas av andra också. I Skapa en webbplats med Wordpress finns mer detaljer om ni vill bygga med Wordpress. Domännamn – allt du vill veta om din adress på nätet förklarar hur nätets adresssystem fungerar.

# 10. Länklista



## Svenska barnhack och liknande initiativ

Vi på IIS håller regelbundet kostnadsfria programmeringskurser för barn. Men det finns fler organisationer i landet som gör samma sak. På vår webbplats [barnhack.se](http://barnhack.se) har vi samlat initiativtagare över hela Sverige! Här hittar du även kostnadsfria videokurser och utbildningsmaterial.

### Verktyg

#### Scratch

<http://scratch.mit.edu>

#### Kojo/Scala

<http://lth.se/programmera>

#### Minecraft

<https://minecraft.net>

#### Kids Ruby

<http://www.kidsruby.com>

#### Littlebits

<http://www.littlebits.cc>

#### Arduino

<http://www.arduino.cc>

### Minecraft-tillägg

#### Computercraft

<http://www.computercraft.info>

#### Feed the Beast

<http://www.feed-the-beast.com>

#### Tekkit

<http://www.technicpack.net>

## **iPad-appar**

### **Hoptscotch**

<http://www.gethopscotch.com>

### **Kodable**

<http://www.surfscore.com>

### **Codea**

<http://twolivesleft.com/Codea/>

## **Lär mer**

### **Rymdspel i Ruby**

<http://rorbecker.com/spelskola>

### **Code Academy**

<http://www.codeacademy.com>

### **Kahn Academy**

<http://www.khanacademy.org/science/computer-science>

### **Learn Python The Hard Way**

<http://www.learnpythonthehardway.org>

### **Computer Science Unplugged**

<http://www.csunplugged.org>

### **Teacherhack**

<http://www.teacherhack.se>

## **Annan inspiration**

### **DIY**

<http://www.diy.org>

### **Makezine**

<http://www.makezine.com>

### **Instructables**

<http://www.instructables.com>

## Anders Thoresson

Anders Thoresson är journalist och föreläsare. Han har bevakat teknikutvecklingen sedan 1999. Först på tidningen Ny Teknik och sedan 2006 som frilans. Under åren 2011–2014 skrev han Teknikbloggen på [dn.se](http://dn.se). Han föreläser i samma ämnen, bland annat om digitalt källskydd för journalister och programmering i skolan för lärare och skolledare. Anders Thoresson har författat flera Internetguider för IIS, exempelvis om säkerhet, webbpublicering och omvärldsbevakning.



Foto: Sebastian LaMotte CC-BY ND



## Barnhack

### - kom igång med programmering!

Internetguide, nr 31. Version 2.0 2016

Anders Thoresson

Texten skyddas enligt lag om upphovsrätt och tillhandahålls med licensen Creative Commons Erkännande 2.5 Sverige.



Illustrationerna skyddas enligt lag om upphovsrätt och tillhandahålls med licensen Creative Commons Erkännande-Icke-Kommersiell-IngaBearbetningar 2.5 Sverige.



Läs mer om ovanstående villkor på <http://www.creativecommons.se/om-cc/licenserna/>

Vid bearbetning av verket ska IIS logotyper och IIS grafiska element avlägsnas från den bearbetade versionen. De skyddas enligt lag och omfattas inte av Creative Commons-licensen enligt ovan.

IIS klimatkompenserar för sina koldioxidutsläpp och stödjer klimatinitiativet ZeroMission.

Författare: Anders Thoresson

Redaktör: Hasse Nilsson

Projektledare: Jessica Bäck

Formgivning: AGoodId

Andra upplagan

ISBN: 978-91-7611-682-1

I denna guide bidrar Tekniska museet med erfarenheter och tips.

TEKNISKA  
MUSEET

**Vi driver internet framåt!** IIS arbetar aktivt för positiv tillväxt av internet i Sverige. Det gör vi bland annat via projekt som samtliga driver utvecklingen framåt och gynnar internetanvändandet för alla. Exempel på pågående projekt är:

**Bredbandskollen**

Sveriges enda oberoende konsumenttjänst för kontroll av bredbandsuppkoppling. Med den kan du på ett enkelt sätt testa din bredbandshastighet.

[www.bredbandskollen.se](http://www.bredbandskollen.se)

**Internetdagarna**

Varje höst anordnar vi Internetdagarna som är Sveriges ledande evenemang inom sitt område. Vad som för tio år sedan var ett forum för tekniker har med åren utvecklats till att omfatta samhällsfrågor och utvecklingen av innehållet på internet. [www.internetdagarna.se](http://www.internetdagarna.se)

**Internetfonden**

Hos Internetfonden kan du ansöka om finansiering för fristående projekt som främjar internetutvecklingen i Sverige. Varje år genomförs två allmänna utlysningar, en i januari och en i augusti. [www.internetfonden.se](http://www.internetfonden.se)

**Internetguider**

IIS publicerar kostnadsfria guider inom en rad internetrelaterade ämnesområden, som webb, pdf eller i tryckt format och ibland med extra-material.

**Internetstatistik**

Vi tar fram den årliga, stora rapporten "Svenskarna och internet" om svenskarnas användning av internet och dessemellan ett antal mindre studier.

**Webbstjärnan**

Webbstjärnan är en skoltävling som ger pedagoger och elever i den svenska grund- och gymnasieskolan möjlighet att publicera sitt skolarbete på webben. [www.webbstjarnan.se](http://www.webbstjarnan.se)

**Internetmuseum**

I december 2014 lanserade IIS Sveriges första digitala internetmuseum. Internetmuseums besökare får följa med på en resa genom den svenska internethistorien. [www.internetmuseum.se](http://www.internetmuseum.se)

**Federationer**

En identitetsfederation är en lösning på konto- och lösenordshandlingen till exempel inom skolans värld eller i vården. IIS är federationsoperatör för Skolfederation för skolan och Sambi för vård och omsorg. [www.iis.se/federation](http://www.iis.se/federation)

**Internets infrastruktur**

IIS verkar på olika sätt för att internets infrastruktur ska vara säker, stabil och skalbar för att på bästa sätt gynna användarna, bland annat genom att driva på införandet av IPv6. [www.iis.se](http://www.iis.se)

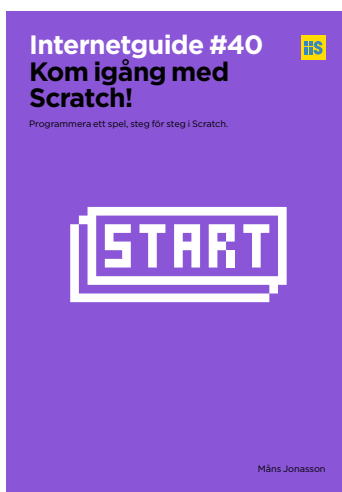
**Sajtkollen**

Sajtkollen är ett verktyg som enkelt låter dig testa prestandan på en webbsida. Resultatet sammanställs i en lättbegriplig rapport. [www.sajtkollen.se](http://www.sajtkollen.se)

**Läs mer på nätet redan idag!** På Internetguidernas webbplats hittar du mängder av kostnadsfria publikationer. Du kan läsa dem direkt på webben eller ladda ner pdf-versioner. Det finns guider för dig som vill lära dig mer om webbpublicering, omvärldsbevakning, it-säkerhet, nätets infrastruktur, källkritik, användaravtal, barn och unga på internet, digitalt källskydd och mycket mer.

---

## Nya Internetguider!



### **Kom igång med Scratch!**

Programmera ett spel, steg för steg i Scratch.

Av: Måns Jonasson

Det här kursmaterialet innehåller utförliga steg-för-steg-instruktioner. Det är till för dig som är en nyfiken nybörjare när det gäller programmering och Scratch. Du lär dig bland annat att bygga ett till spel. Blir du biten finns det mer material att läsa och se på [barnhack.se](http://barnhack.se)!



### **Ungas integritet på nätet**

Råd till dig som är vuxen

Av: Åsa Secher

Den här guiden belyser vad integritet på nätet betyder för unga idag. Det är ett viktigt ämne att ta sig an för att vi som vuxna ska kunna stötta och hjälpa barn och unga att känna var deras gränser går så att de inte råkar illa ut. Guiden riktar sig till vuxna i barns närhet och även om internetanvändandet börjar i tidig ålder, handlar innehållet framför allt om unga i åldrarna 10 till 16 år. Du får bland annat ta del av vad unga gör på nätet, vad grooming är, hur du som vuxen kan hjälpa unga att sätta gränser och vad Barnkonventionen säger om ungas rätt till integritet. Guiden är producerad i samarbete med Barnens Rätt i Samhället (BRIS).