

Startövning micro:bit

Tärningen



Att räkna med slumpen genom t.ex. tärningsslag och föra statistik över antal slag kan vara tidskrävande. Speciellt om man ska göra 100 st eller 1000 st kast.

Genom att använda programmering till att simulera tärningskast, skapa listor som automatiskt lagrar värden kan du förstärka och förenkla det matematiska momentet samt få in en programmeringsuppgift i matematikundervisningen. I denna handledning gör vi första steget: simulerar tärningskast med hjälp av blockprogrammering och en Micro:bit.

MÅL MED ÖVNINGEN:

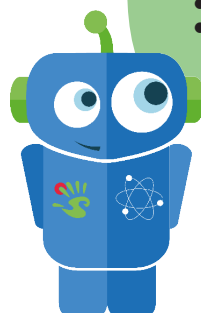
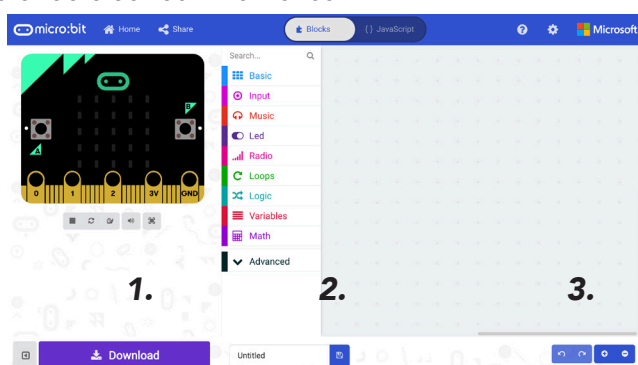
Att förstå hur du påverkar en micro:bit med olika kommandon som såsom if-sats och variabler. Att få micro:bit att ange slumpvisa tal när du skakar på den. Eleverna får bekanta sig med enkel blockprogrammering och dess terminologi.

HUR GÖR DU?

Programmera med micro:bit

För att börja skriva ditt program gå till:

www.microbit.org och välj **lets code**. På den vänstra sidan har du simulerings bilden (1). dvs en simulering av en micro:bit. Bredvid i mitten ligger ditt kodbibliotek (2) med alla block. Till höger finns kodytan (3) dit du drar de block du vill använda.



Det här behöver du:

- micro:bit
- Dator eller iPad
- USB-kabel
- Batteribox till micro:bit

En digital tärning med micro:bit

I denna handledning lär du dig programmera micro:bit till att fungera som en tärning. Övningen ger grundläggande kunskap i blockprogrammering samt användning av variabel, if/else satser och slumpgenerator.

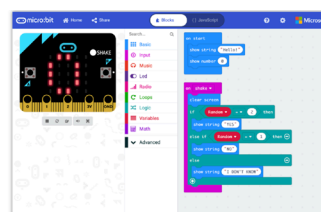
Microbit kan programmeras i olika miljöer

JavaScript Block Editor: Denna programmeringsmiljö utvecklas av Microsofts MakeCode är i grunden JavaScript. Det som är fuffigt med denna editor är att man kan skifta mellan blockprogrammering och textprogrammering. Man alltså börja programmet i block och avsluta i text. I editorerna har man en simulator av micro:bit på skärmen.

Python Editor: Textbaserad miljö med Python som språk. Renodlad textmiljö.

Kom igång med tärningen:

Välj: **JavaScript block editor** och öppna ett nytt projekt genom att trycka på **lets code**



JavaScript Blocks Editor

The JavaScript Blocks editor powered by MakeCode makes it easy to program your micro:bit with blocks and JavaScript.

We have recently updated the editor, and the previous version is still available for anyone that needs it. If you have any issues accessing the editor, check that it isn't blocked in your school.

[Let's Code](#)

[Reference](#)

[Lessons](#)

Tärningen i olika nivåer

Vi vill att micro:biten ska generera ett slumpstal mellan 1-6 och visa siffran på sin LEDdisplay. Med andra ord skapa en tärning. Handledningen kommer visa hur man kan skapa en tärning på tre olika sätt med blockprogrammering. Vi börjar med den enklaste formen av tärning.

VAD SÄGER LÄROPLANEN:

Lgr11, Centralt innehåll:

Matematik:

- Algebra i årskurs 4–6: Hur algoritmer kan skapas och användas vid programmering. Programmering i visuella/olika programmeringsmiljöer.

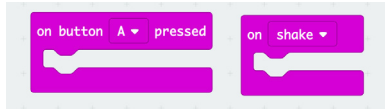
Teknik:

- Arbetssätt för utveckling av tekniska lösningar i årskurs 1–3: Att styra föremål med programmering.
- Arbetssätt för utveckling av tekniska lösningar i årskurs 4–6: Att styra egna konstruktioner eller andra föremål med programmering.

Visa slumptal när tärningen kastas

Träningen ska "kastas" genom att använda en *input*. En input är ett sätt att mata in data eller i vårt fall ett sätt att starta ett program.

För att starta ditt program behöver du först bestämma vilken input som ska styra din tärning genom att välja mellan dessa block:

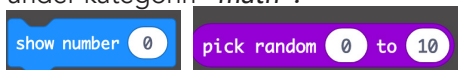


Knapptryckning eller skaka tärningen i handen? Välj den du tycker passar bäst, nu dags för slumpgeneratorn. I denna handledning visar vi hur input "shake" slumpar tärningen.

För att välja input, går vi till kategorin "Input" och välj *on shake blocket*.

När vi valt input "shake" ska vi bestämma vad som sker inuti "shake-blocket". I detta fall vill vi att micro:bit ska visa och slumpa en siffra mellan 1-6. För att få micro:biten att göra detta används blocket ett "show number" som du hittar under kategorin "Basic".

Nästa block slumpar fram en siffra, "pick random" hittar under kategorin "math".



Blocken i micro:biten är uppbyggda så att man ser vilka block som passar att sammanfoga.

Istället för att visa ett bestämt nummer vill vi att micro:biten ska visa ett slumpat nummer när den skakas. Vi behöver därför ändra intervallet 0 till 10 till 1 till 6. Sammanfoga samtliga block som bilden visar:



Testa att simulera innan du för över programmet till micro:biten. Varje gång du simulerar "Shake" visas ett nummer mellan 1-6.

Tänk på att inte ge en färdig steg för steg instruktion till eleverna när du ska lära ut hur man skapar en tärning utan ge byggstenarna och var tydlig med att förklara vad du vill att eleverna ska åstadkomma. Risken att eleverna endast följer instruktioner utan att förstå vad de gör.

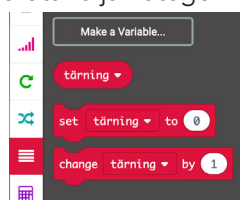
Använda en variabel till tärningen

I programmering använder man variabler för att hålla reda på t.ex. siffror eller text. Likt en behållare eller en mapp där man kan samla saker. I denna version av tärning med micro:bit vill vi att "tärningen" ska generera ett slumptal mellan 1-6 inuti en variabel. Vi använder "shake" som input för att "slå" tärningen.

Variabeln skapar du själv. Börja med att välja kategorin som heter "Variable".

Skapa en variabel och döp den förslagsvis till "tärning".

Vi behöver först bestämma vad variabeln ska innehålla. I vårt fall vill vi att den ska slumpa ett nummer likt en träning. För att göra detta behöver vi två block. Med det första blocket säger vi vad variabeln "tärning" ska innehålla. Innehållet kommer att ersätta 0:an.



Gå vidare till kategorin math och hämta slumpblocket "pick random 0 to 10".

Före dessa och sätt tärningen att slumpa mellan 1-6.



Nästa steg är att talen ska visas på micro:bitens LEDdisplay.

Du hittar blocket "Show number" i menyn Basic.

Därefter byter du ut 0:an mot variabeln *tärning*.

Vad som händer nu är att variabeln *tärning* slumpar fram en siffra mellan 1-6 och genom att sedan visa variabeln får vi fram siffran på micro:bitens display.

Så här ser den färdiga koden ut för en tärning med en variabel med siffrorna 1 till 6:



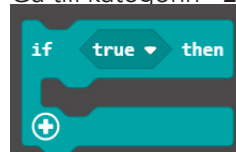
Tärning med variabel som visar tärningens prickar

Nu när vi lärt oss hur man kan skapa en variabel så ska vi använda den till att visa tärningens prickar istället för siffror. Inom programmering så kan man använda villkor/alternativ för att bestämma hur programmet ska bete sig.

Vi kommer använda oss av villkoren if/else.

Vi kan bygga vidare på den träning vi redan har med en variabel. Nu ska vi även lägga till ett logikblock.

Gå till kategorin "Logik". Välj blocket "if true then".

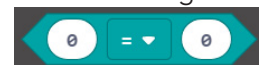


Vi vill att if-satsen ska vara baserad på variabeln "tärning". D.v.s om "tärning" == 1 visas en prick.

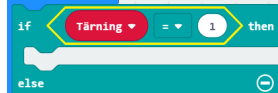
Märkte du att det var två stycken = tecken? Ekvivalens inom programmering skrivs med två stycken "= tecken" medan ett "=" tecken är en tilldelning, i blocken står dock endast ett likamedtecken men växlar man till JavaScript kommer det skrivas ut med ==.

if (Tärning == 1) {

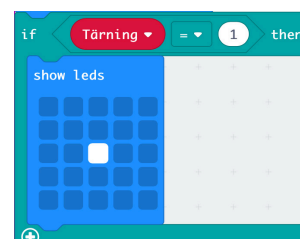
För att definiera villkoren så behöver vi något som bestämmer villkoren. Under samma kategori Logik hittar vi blocket "0 = 0"



Vi ersätter första 0:an med variabeln "Tärning" och sätter första villkoret. Sammanfoga "=blocket" med if-satsen:



Nästa steg är att rita tärningens sidor med prickar. Under kategorin "Basic" finns blocket "show led". Här bestämmer vi vilka av micro:bitens LED:s som ska lysa.



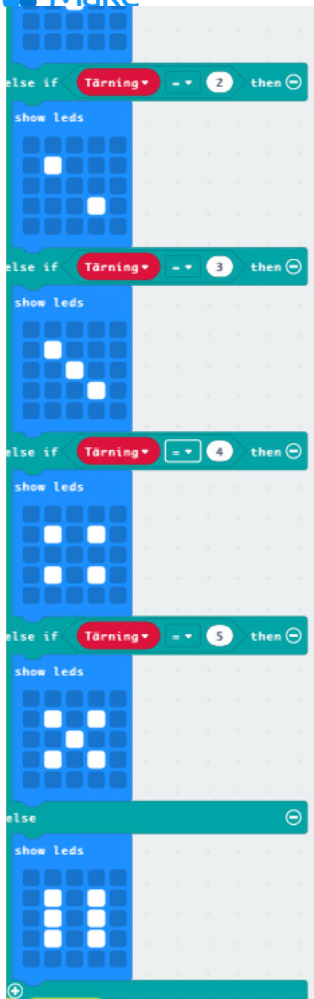
Fyll i en prick i mitten och sammanfoga blocket i "if" blocket.

Tryck på plustecknet i "if" blocket för att skapa villkor för siffrorna 2, 3, 4 och 5.

Siffran 6 behöver du inte

sätta något villkor för. Fundera på varför?

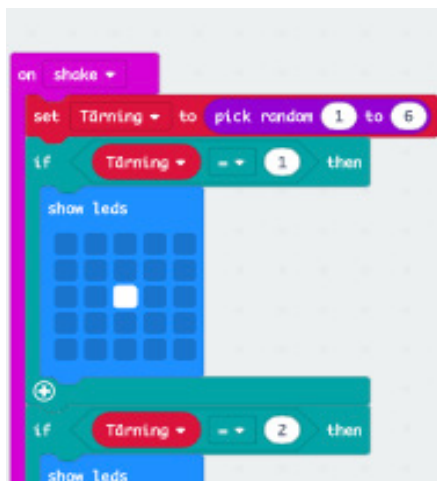
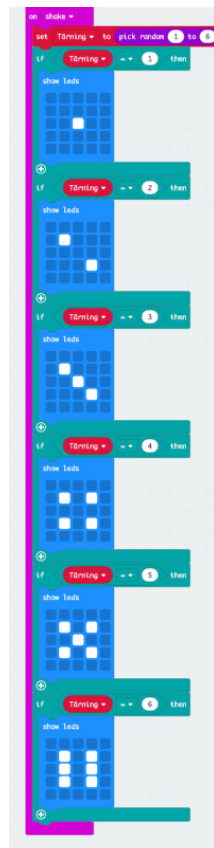
Svaret kommer längre ner. Använd samma block som ovan för de resterande av tärningens sidor.



Kom du på varför vi inte behöver skapa ett villkor till siffran 6?

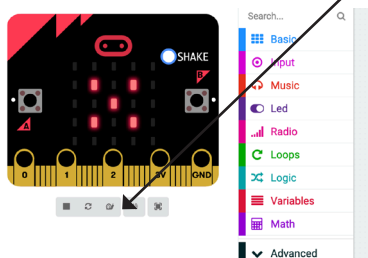
Vi sätter villkor för att om 1, 2, 3, 4 och 5 slumpas fram ska det visas motsvarande prickar på "tärningen". Men om ingen av dessa slumpas fram så är det ju 6:an som har slumpats fram och då ska sex prickar visas. Nu samfogar vi alla block. Ta bort "show number" blocket då vi vill att det är prickarna vi ritat som ska visas. Så här ser början och hela programmet ut: Villkoret "else" betyder alltså att om ingen av villkoren uppfylls används else-kommandot.

Nu samfogar vi alla block. Ta bort "show number" blocket då vi vill att det är prickarna vi ritat som ska visas. Så här ser början och hela programmet ut:



Testa nu i simulatorn:

Testa ditt program genom att trycka på play-knappen under micro:biten till vänster. Detta kan du göra flera gånger under din programmering.



För över program till micro:biten.

Bra jobbat! Till sist är det dags att föra över ditt program till din micro:bit.

För att göra detta börjar du med att koppla micro:bit till usb-kabeln och sedan kabeln in i datorn. Sedan trycker du på **download**. Då skapas en hex.fil som kan sparas på micro:bit-kortet. Denna fil är inte läsbar. Så du kan inte öppna den och se ditt färdiga program.



Filen läggs i foldern hämtade filer och heter micro-bit-Untitled.hex, flytta över filen till din micro:bit precis som du gör när du lagrar något på ett USB-minne.

Har du gjort ditt program på en lärplatta? Ta hjälp av denna tutorial: <https://youtu.be/Hwhws2e-cfl>

Tärningen är kastad!

