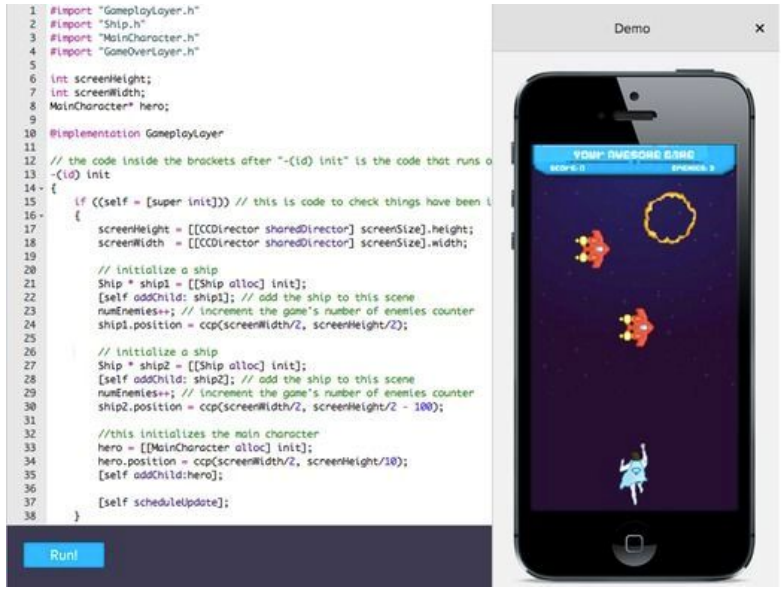
PROGRAMMERING ÅK 9 INTRODUKTION

VARFÖR PROGRAMMERING?

Med programmering kan man:

★ Skapa nästan vad som helst som är digitalt. Allt som är digitalt är uppbyggt av kod som människor har skrivit, finns i både mobiltelefon och mikrovågsugn

★ Lösa problem som vi människor har svårt att lösa själva



VAD ÄR PROGRAMMERING?

Programmering är att:

- ★ **Lösa problem**: fundera ut vad det är man vill göra med sitt program, dela upp det i flera små delar och hitta en lösning för varje del
- ★ **Planera kod**: beskriva med vanliga ord hur man tror att instruktionerna ska se ut för att lösa de uppgifter som behövs. Vilken ordning ska instruktionerna vara i? Hur ska instruktionerna för de olika uppgifterna hänga ihop?
- ★ **Skriva kod**: i det programmeringsspråk som man vill använda.
- ★ **Testa kod**: bra tips är att testa ofta och i så små delar som möjligt. Då är det lättare att vara säker på att koden fungerar som man tänkt sig.

TÄNKA DATALOGISKT

- ★ Datalogiskt = tänka på ett sätt som en dator förstår
- ★ Varför ska vi göra det?
  - En dator är **bra på att tänka snabbt** men den är **dålig på att tänka själv**

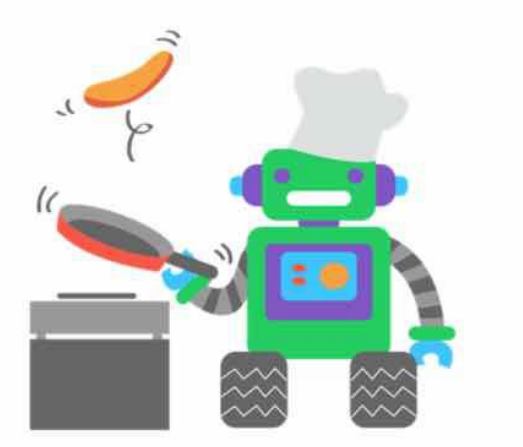
- ★ Därför måste **de instruktioner som vi ger datorn** vara:
  - **I rätt ordning**
  - **Exakta**
  - **Fullständiga**

EXEMPEL: ROBOT SOM SKA LAGA PANNKAKOR

Om vi ska lära en robot att laga pannkakor måste vi dela upp uppgiften "att laga pannkakor" i mindre deluppgifter

Varje uppgift måste vi definiera genom att tänka datalogiskt = så att roboten förstår

På samma sätt måste vi göra när vi ska programmera en dator



PANNKAKSRECEPT FÖR EN ROBOT

Ta fram en skål

(du måste berätta vad en skål är och var den finns)

Lägg följande ingredienser i skålen:

3 ägg

(du måste förklara att man först behöver knäcka äggen, sedan lägga ägget i skålen utan äggskal)

Mjöl, 2,5 dl

Mjölk, 6 dl

Salt, 0,5 tsk

Smör, 3 msk

(du måste förklara hur ingredienserna ser ut, var de finns och vad dl, tsk och msk betyder om inte roboten vet det redan)

Vispa alla ingredienser i skålen till pannkakssmet

(hur gör man när man vispar? Vad är en smet? Förklara!)

Ta fram en stekpanna

(förklara vad en stekpanna är och var den finns)

Sätt stekpannan på spisen och sätt på plattan

(vad är en spis, vilken platta? Samma som stekpannan står på? Hur sätter man på den?)

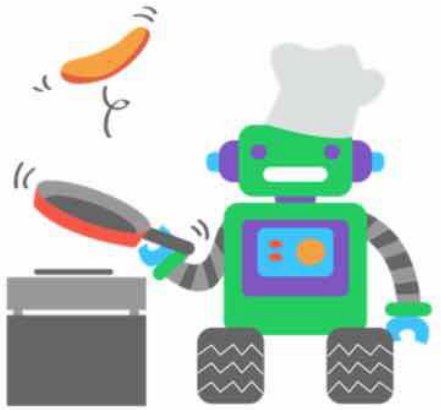
Vänta tills stekpannan blivit varm

När stekpannan är varm, häll i pannkakssmet

(du behöver berätta hur mycket smet du ska ha i)

Stek tills lagom gräddad

(vad är lagom gräddad? Det måste du berätta!)



“Rätt ordning”, “exakt” och “fullständig” för roboten

De instruktioner som vi ger datorn (roboten) måste vara:

1. **I rätt ordning**

   Datorn kan inte förstå om den behöver göra något i en annan ordning. Vad skulle hända om roboten knäckte ägg direkt i den varma stekpannan och hällde resten av ingredienserna i skålen? Du skulle bara få stekt ägg med pannkakssmet bredvid!

2. **Exakta**

   Datorn kan inte förstå om du stavar fel i din instruktion eller om du bara säger på ett ungefär hur något ska göras. Om du vill att en robot ska göra pannkakor behöver du till exempel ha exakta mått på hur mycket ägg, mjölk, mjöl och så vidare som behövs.

3. **Fullständiga**

   Datorn kan inte förstå om du har glömt ett tecken eller att skriva något som den behöver göra i instruktionen. Den kan inte tolka ofullständiga meningar på samma sätt som en människa kan. Alla moment är viktiga och måste beskrivas i detalj. Om du säger “knäck ägg i skålen” till pannkaks-roboten så vet den inte att äggskalen in ska med.

Arbeta i grupp och definiera upp er första kod

Ni ska bordsvis på samma sätt beskriva för en robot hur den ska göra **en** av följande aktiviteter (ni väljer gemensamt vilken):

- ★ Borsta tänderna
- ★ Sätta på sig kläder
- ★ Äta lunch

Försök att beskriva så noga som möjligt så att roboten förstår och i rätt ordning.

Skriv upp “receptet” till roboten i era skrivböcker och lämna in till mig efter lektionens slut.

ALGORITM

- En **instruktion som löser ett problem** eller en uppgift **kallas för en algoritm**
  - **En specifik instruktion** i en algoritm kallas ett **kommando**

- En algoritm måste kunna förstås av den som ska utföra algoritmen!
  - Datorer är förprogrammerade att förstå vissa saker (beroende på vilket språk de pratar). Det som de inte redan vet måste vi programmera dem att förstå.
  - Det finns nästan oändligt många algoritmer för att lösa samma problem.
  - Målet är att konstruera algoritmer som är *korrekta*, *enkla* och *effektiva* (så att vi människor kan läsa dem)

FINNS MÅNGA OLIKA PROGRAMMERINGSSPRÅK

- Olika sätt att prata med en dator beroende på vad man vill få den att göra. Exempelvis
  - C, C++ - "hårdvarunära" för att programmera kameror, mikrovåsugnar
  - Java script - internetapplikationer typ filmvisning
  - Matlab – matematiska beräkningar
  - Java - android
  - Swift - Apple

- Språket vi ska jobba med heter Swift och är utvecklat av Apple
- Swift Playground: Swift lekstuga. Vi ska lära oss Swift genom att programmera en robot att utföra olika uppgifter

KOMMANDO

Inom programmering är kommando ett direktiv till ett datorprogram att utföra en viss uppgift (kan vara sammansatta delar

Exempelvis: ta upp ett ägg ur äggförpackningen med din högra hand (vad måste roboten veta för att kunna göra det?)

“SEKVENS”, “ALTERNATIV” och “REPETITION”

Tre typer av algoritmer som är vanliga och användbara:

- ★ Sekvens
  - Gör först det här sedan det här och sist det här (knäck först ägget, ha sedan i det i smeten)
- ★ Alternativ
  - Om det här är uppnått så gör det här (om det är torsdag ska roboten steka pannkakor, annars koka gröt)
- ★ Repetition
  - Gör en sekvens om och om igen. Upprepa, medan, tills... (vispa ända tills det inte längre finns några klumpar kvar i smeten)

FUNKTION - SAMMANSÄTTNING AV KÄNDA KOMMANDON

En del av ett program som kan anropas (köras) för att utföra en viss uppgift

Kan anropas flera gånger på olika ställen i ett program

En funktion innehåller en uppsättning av (för datorn) kända kommandon

Innehållet i funktionen kallas kropp - den kod som körs när man anropar funktionen

T ex

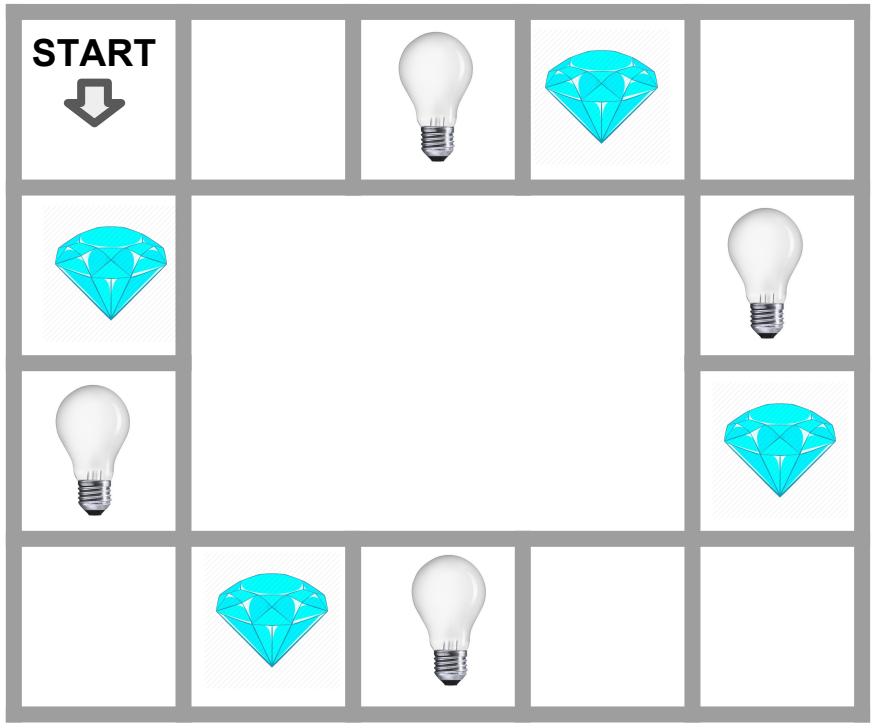
VARFÖR ANVÄNDER MAN FUNKTIONER?

Man använder funktioner för att:

- Inte behöva skriva samma sak flera gånger
- Göra program lätta att överblicka
- Inte behöva göra ändringar på flera ställen i ett program
- Låta flera programmerare arbeta med olika delar av koden samtidigt

Fundera på hur koden skulle kunna se ut för att:
Tända alla lampor och samla alla juveler med hjälp av en funktion

Använd kommandon som datorn känner till:
Gå Framåt = MoveForward
Sväng Vänster = TurnLeft
Samla Juvel = CollectGem
Slå på strömbrytaren = ToggleSwitch



Med funktion:

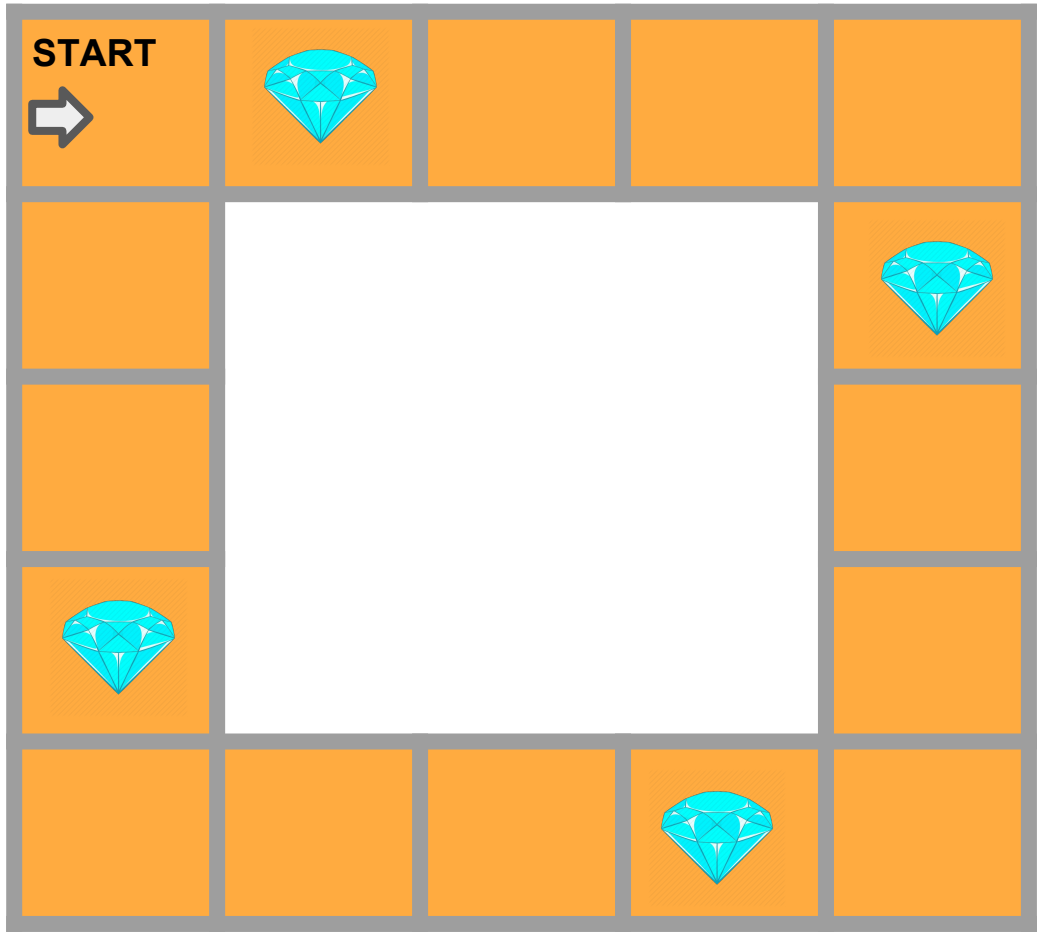
```
func moveAndCollect() {
    moveForward()
    collectGem()
    moveForward()
    toggleSwitch()
    moveForward()
}

moveAndCollect()
turnLeft()
moveAndCollect()
moveForward()
turnLeft()
moveAndCollect()
turnLeft()
moveAndCollect()
```

14 rader

Skriv koden för att samla in alla juveler

Använd kommandon som datorn känner till:
Gå Framåt = MoveForward
Sväng Vänster = TurnLeft
Samla Juvel = CollectGem



Med for loop - 8 rader kod

```
for i in 1 ... 4 {
    moveForward()
    collectGem()
    moveForward()
    moveForward()
    moveForward()
    turnRight()
}
```

REPETITION AV ETT MÖNSTER ETT ANTAL GÅNGER - FOR LOOP

For Loop

Upprepar en funktion ett bestämt antal gånger

Viktigt att hitta mönster - vad är upprepning?
T ex

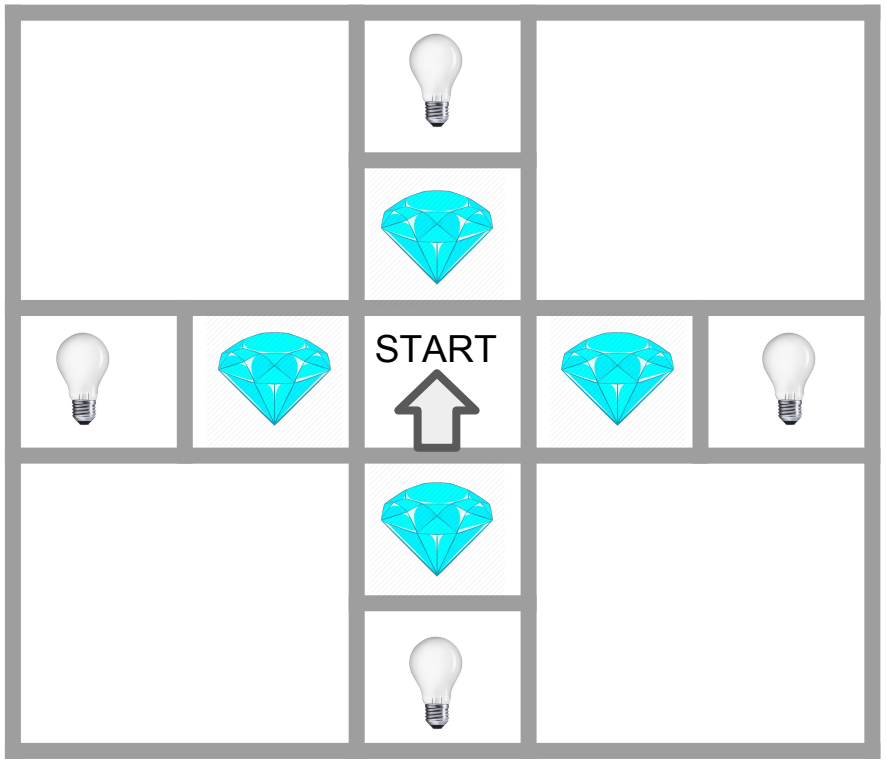
Om din pannkaksrobot ska steka pannkakor till många personer kan det vara bra att upprepa koden för att steka **en** pannkaka flera gånger istället för att skriva om samma rad kod flera gånger.

```
for i in 1...10 {
    laggISmorIStekpannan()
    hallSmetiStekpannan()
    stekPannkaka()
    serveraPannkaka()
    }
```

Nu gör roboten 10 pannkakor. Hur skulle det se ut om man skulle göra 20 pannkakor?

för dessa värden ska funktionen upprepas

Övning: Samla in alla juveler och tänd alla lampor med hjälp av en ***for loop***



<u>Kända kommandon</u>

Gå Framåt = MoveForward() = mf
Sväng Höger = TurnRight() = tr
Samla Juvel = CollectGem() = cg
Slå på strömbrytaren = ToggleSwitch() = ts

Med loop: 11 rader kod

```
for i in 1...4 {
MoveForward()
CollectGem()
MoveForward()
ToggleSwitch()
TurnRight()
TurnRight()
MoveForward()
MoveForward()
TurnRight()
        }
```

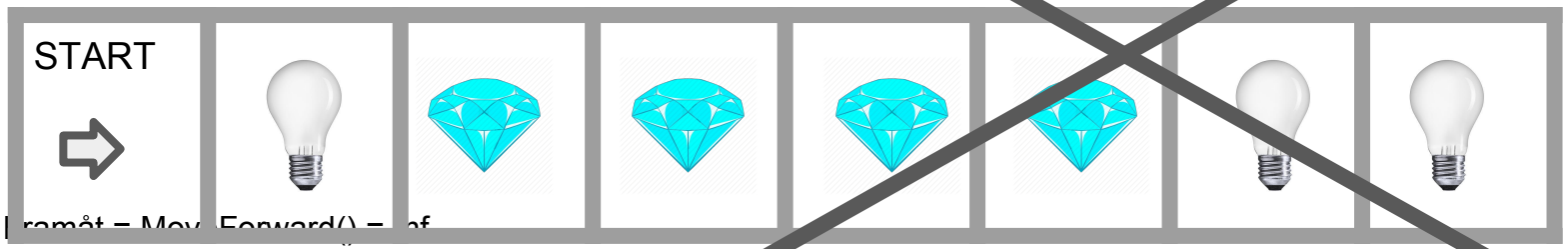
6 rader kod

```
for i in 1...10 {
        laggISmorIStekpannan()
        hallSmetiStekpannan()
        stekPannkaka()
        serveraPannkaka()
        }
```

Pannkaksrobot

https://www.youtube.com/watch?v=v9oeOYMRvuQ

Övning: Bakom varje frågetecken finns det antingen en lampa att tända eller en juvel att samla in. Hur skulle koden för att samla in alla juveler och tända alla lampor kunna se ut (utan att du vet var de olika finns?)



Gå framåt = MoveForward() = mf
Samla Juvel = CollectGem() = cg
Slå på strömbrytaren = ToggleSwitch() = ts

```
for i in 1...7

MoveForward()
If IsOnClosedSwitch  {
        toggleSwitch()
} else if IsOnGem {
        collectGem()
}
```

IF och ELSE IF

IF sats

Används för att **i förhand bestämma hur ett program ska reagera på olika värden** (t ex om det finns en juvel eller en lampa på en viss plats)

En **If sats** innehåller **ett villkor** och **ett kommando**

Villkoret **måste vara uppfyllt för att kommandot ska köras**

Ett villkor är **antingen uppfyllt eller inte uppfyllt** (antingen sant eller falskt)

Else if används på samma sätt men med ett annat villkor och kommando

Exempel IF-sats

Roboten ska laga pasta varje onsdag och laga soppa alla andra dagar:

```
if ÄrOnsdag {
    LagaPasta()
}
else if ÄrInteOnsdag
    LagaSoppa()
}
```

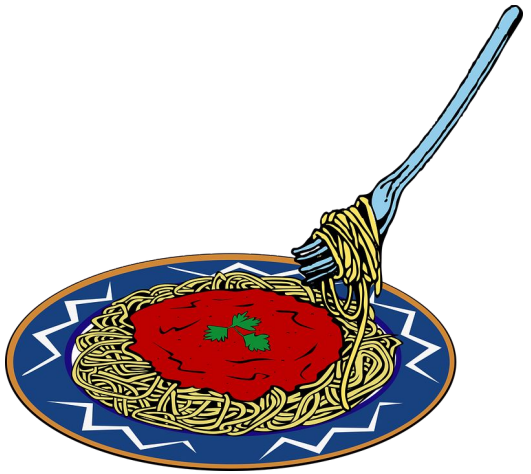
Villkor som ska vara uppfyllt

Kommando som körs

Villkor som ska vara uppfyllt

Kommando som körs

Onsdag

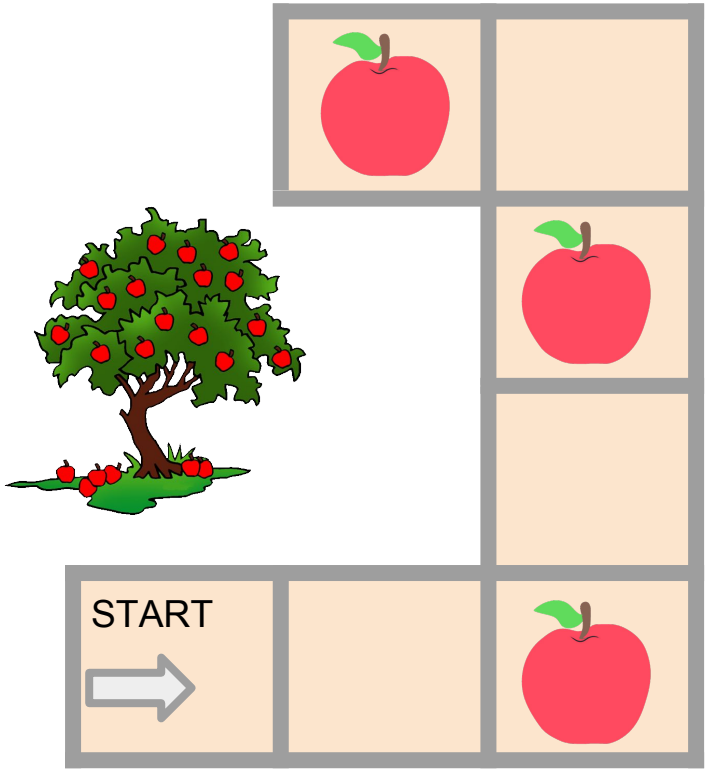


Måndag
Tisdag
Torsdag
Fredag
Lördag
Söndag

KOM IHÅG

- ★ En dator kan bara göra det vi har sagt åt den att göra. En dator lyder bara det vi människor har programmerat i kod.
- ★ Det finns alltid flera olika lösningar på samma problem.
- ★ Det är bra att först tänka ut vad det är du vill att datorn ska göra och försöka beskriva det med vanliga ord innan du skriver koden.

Övning: Skriv koden för att samla in alla äpplen med hjälp av en ***funktion*** för att svänga vänster



Kända kommandon:
GåFramåt() = gf
SvängHöger() = sh
SamlaÄpple() = sä

STRATEGI - SKAPA FUNKTION

Steg 1. **VAD** ska din **funktion göra**?

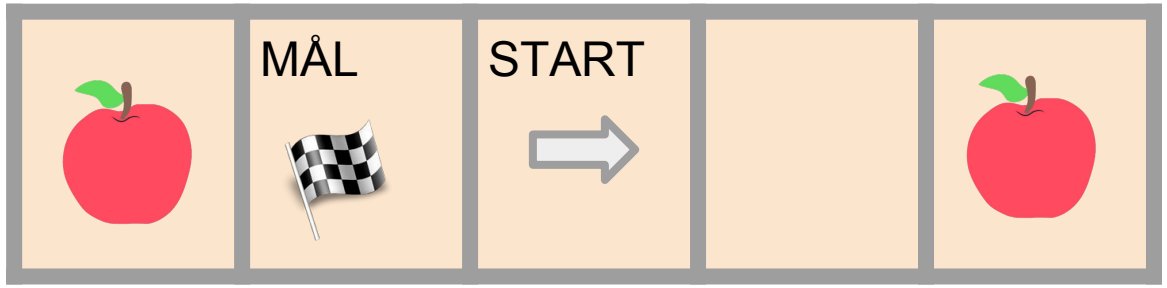
Steg 2. **VILKA kommandon behövs** för att göra det?

Steg 3. Skriv **"func"**, **namnge din funktion** och **lista de kommandon som ska ingå** i funktionen

Steg 4. **Anropa din funktion tillsammans med andra kommandon** för att lösa uppgiften

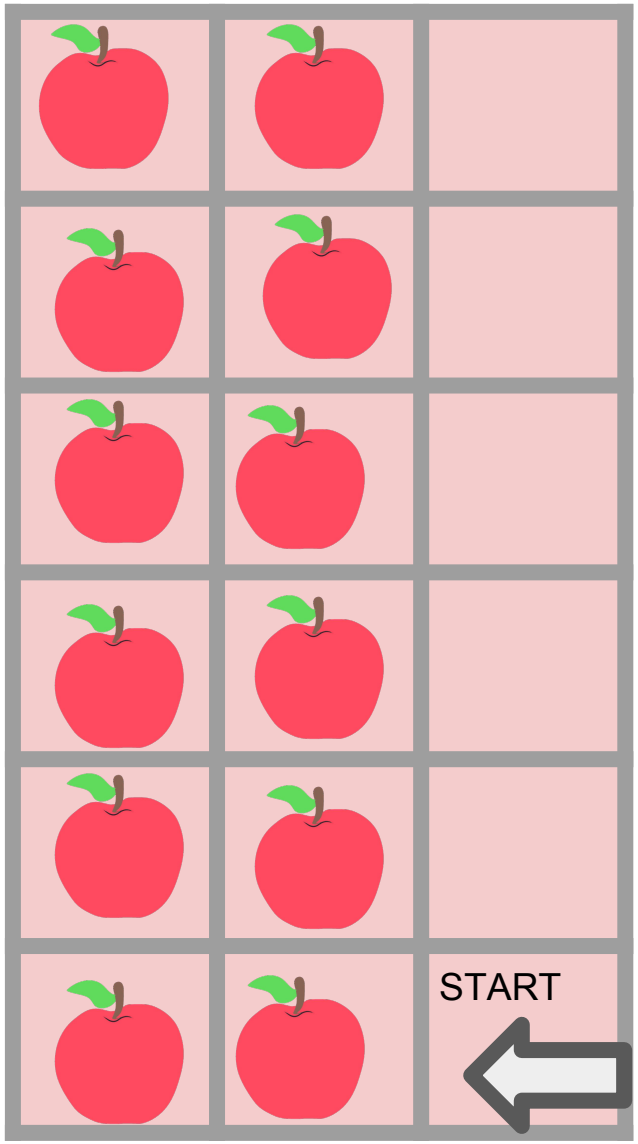
Steg 5. **VIKTIGASTE! Testa om det fungerar**. Om inte - var har det gått fel? **Vad ska du ändra på?**

Övning: Skriv koden för att samla in alla äpplen med hjälp av en ***funktion*** för att "vända om roboten"



Kända kommandon:
GåFramåt() = gf
SvängHöger() =sh
SamlaÄpple() = sä

Skriv koden för att samla in alla äpplen med hjälp av en for loop



Kända kommandon:
GåFramåt() = gf
SvängHöger() = sh
SvängVänster() = sv
SamlaÄpple() = sä

STRATEGI - FOR LOOP

Steg 1. **VAD** är det för **mönster** som **upprepas**?

Steg 2. **HUR många gånger upprepas mönstret**?

Steg 3. **GÖR** en **loop runt de kommandon som upprepas** använd **for i in 1 ... X** (där X är antal gånger)

Steg 4. **UNDERSÖK** om det behövs **andra kommandon runt loopen**. Skriv dem isåfall där de behövs (före, efter eller i loopen)

Steg 5. **VIKTIGASTE!** Testa om det fungerar. Om inte - var har det gått fel? **Vad ska du ändra på?**

Bedömningsmatris provet

	E	C	A
Resonemang	Förklarar din lösningsstrategi så att man ungefär förstår hur du har tänkt	Förklarar din lösningsstrategi så att man förstår hur du har tänkt	Förklarar din lösningsstrategi så att man tydligt förstår hur du ha tänkt
Tekniska begrepp	Använder vissa korrekta tekniska begrepp	Använder många av de korrekta tekniska begreppen	Använder alla korrekta tekniska begrepp: anropa, kommando, funktion och loop
Funktioner	Skapar och namnger funktion med kommandon. Anropar funktionen tillsammans med övriga kommandon och klarar med större justeringar uppgiften	Skapar och namnger funktion med kommandon. Anropar funktionen tillsammans med övriga kommandon som behövs och klarar med mindre justeringar uppgiften	Skapar och namnger funktion med kommandon. Anropar funktionen tillsammans med övriga kommandon som behövs och klarar uppgiften
Loop	Hittar upprepande mönster och skriver loopar som med större justeringar klarar uppgiften	Hittar upprepande mönster och skriver loopar som med mindre justeringar klarar uppgiften	Hittar upprepande mönster och skriver korrekta loopar som klarar uppgiften

Ord att kunna på provet

Anropa

Kommando

Funktion func NAMN {

}

For loop for i in 1 … X {

}