**Matematik och Spelprogrammering**

* [p5Dojo](https://christernilsson.github.io/p5Dojo) p5dojo.com
* [p5](https://p5js.org/reference)
* [Coffeescript](http://coffeescript.org/)
* [Javascript](https://www.w3schools.com/js)
* [Underscore](http://underscorejs.org/)
* [p5Assert](https://christernilsson.github.io/p5Assert)
* [Nilsson](https://github.com/ChristerNilsson/Nilsson/blob/master/README.md)

**Så här arbetar du!**

* Du ska med kod återskapa den första bitmappen
* Resultatet av din kod hamnar i den andra bitmappen
* Skillnaden visas i den tredje bitmappen. Den ska bli helt svart när du har löst uppgiften.

**Chrome och Windows är ett krav.**

Ingen annan programvara behöver installeras.

**färger**

* 0,0,1 blå
* 0,1,0 grön
* 0,1,1 cyan
* 1,0,0 röd
* 1,0,1 magenta
* 1,1,0 gul
* 0 svart
* 0.5 grå
* 1 vit

**bakgrundsfärg**

* **bg** 1 vit
* **bg** 1,1,0 gul

**fyllningsfärg**

* **fc()** ingen
* **fc** 1 vit
* **fc** 1,1,0 gul
* **fc** 1,0,0 röd
* **fc** 1,0,0,0.5 röd, halvgenomskinlig

**streckfärg**

* **sc()** ingen
* **sc** 1 vit
* **sc** 1,1,0 gul
* **sc** 1,0,0 röd
* **sc** 1,0,0,0.5 röd, halvgenomskinlig

**strecktjocklek**

* **sw** pixlar

**ritkommandon**

* **point** x,y
* **line** x1,y1, x2,y2
* **ellipse** x,y, w,h
* **circle** x,y,r
* **rect** x,y, w,h
* **triangle** x1,y1, x2,y2, x3,y3
* **quad** x1,y1, x2,y2, x3,y3, x4,y4
* **arc** x,y, w,h, start,stopp, PIE

**lerp**

* linjär interpolation och extrapolation
* **lerp** 10,12,-1 == 8
* **lerp** 10,12,0 == 10
* **lerp** 10,12,0.5 == 11
* **lerp** 10,12,1 == 12
* **lerp** 10,12,2 == 14

**for loop**

* Glöm ej att indentera innehållet med ett tabsteg!
* **for** i in range 10 # [0,1,2,3,4,5,6,7,8,9]
* **for** i in range 5 # [0,1,2,3,4]
* **for** i in range 1,11 # [1,2,3,4,5,6,7,8,9,10]
* **for** i in range 0,10,2 # [0,2,4,6,8]
* **for** i in [0..10] by 2 # [0,2,4,6,8,10]
* **for** i in range 10,0,-2 # [10,8,6,4,2]
* **for** i in [1,1,2,3,5,8,13,21] # [1,1,2,3,5,8,13,21]

**if**

if i%3==0

fc 0

else if i%3==1

fc 0.5

else

fc 1

**koordinatsystemet**

* **translate** x,y flytta origo
* **rd** degrees rotera runt origo
* **scale** n skala upp eller ner

**modes**

* **rectMode** CORNER
  + CORNER (default)
  + CORNERS
  + CENTER
  + RADIUS
* **ellipseMode** CENTER
  + CORNER
  + CORNERS
  + CENTER (default)
  + RADIUS

**text**

* **textAlign** LEFT,BASELINE
  + LEFT (default)
  + CENTER
  + RIGHT
  + TOP
  + CENTER
  + BOTTOM
  + BASELINE (default)
* **textSize** n
* **text** "p5",x,y

**push & pop**

Sparar och återställer följande kommandon:

* rotate scale translate fc sc sw rectMode
* tint strokeCap strokeJoin imageMode ellipseMode colorMode
* textAlign textFont textMode textSize textLeading
* [information](https://www.processing.org/tutorials/transform2d)

**coffeescript**

* Orsak: Programmering ska vara så enkelt som möjligt
* Kodblock indenteras med tab (som Python) i stället för blockparenteser {}
* Tabstorlek alltid två mellanslag
* Python-kolon används ej
* Semikolon är frivilliga
* Parenteser behövs bara för att anropa funktioner som saknar parametrar.
* Funktioner skapas med ->
* f = (x) -> x\*x
* g = (a,b) -> a+b

**exempel 1: CoffeeScript**

bg 1, 0.5, 1

sw 2

sc 0.5

for i in range 10

fc i % 2

rd 5

rect 20 \* i + 5, 5, 10, 10

**exempel 1: normal Javascript**

background(255, 127, 255);

strokeWeight(2);

stroke(127);

for (var i = 0; i < 10; i++) {

fill((i % 2) \* 255);

rotate(radians(5));

rect(20 \* i + 5, 5, 10, 10);

}

**exempel 2: funktion i Coffeescript**

f = (a,b) -> a+b

**exempel 2: funktion i Javascript**

function f(a,b) {

return a+b;

}

**Interaktivitet**

Lektionerna i slutet innehåller interaktiva moment. Det innebär att man med kommandon påverkar ett objekts tillstånd. Utritningen av objektet är beroende av tillståndet.

**Tredje listboxen**

Denna innehåller kommandona. Dessa måste definieras i koden. Lämpligen gör man det i samma ordning som programkoden. Det innebär att man börjar med reset. Reset anropar super för att kunna rensa ut felstavade egenskaper. Efter reset kommer draw. Draw ritar upp bitmappen. Draw bör ej anropa super.

**Att tänka på**

Tabellen längst ner innehåller tillståndet.

* Första kolumnen innehåller egenskapernas namn. Alltid gul.
* Andra kolumnen innehåller förebildens data. Alltid grön.
* Tredje kolumnen innehåller resultat av användarens kod. Röd som blir grön.
* Klicka på reset om det ser konstigt ut.
* Draw anropas automatiskt varje gång innehållet i editorn förändras.

**Indata**

Följande funktioner läser indata från textrutan under skillnadsbitmappen:

* @readText()
* @readInt()
* @readFloat()

**mera information**

* [download](https://p5js.org/)
* [manual p5.js](https://p5js.org/reference)
* [coffeescript](http://coffeescript.org/)
* [engelsk e-bok i färg (om fem minuter) av Lauren McCarthy, SEK 55](https://play.google.com/store/books/details?id=iP3GCgAAQBAJ&rdid=book-iP3GCgAAQBAJ&rdot=1&source=gbs_atb&pcampaignid=books_booksearch_atb)
* [svartvit pappersbok (om fem dagar), 130 SEK](https://www.adlibris.com/se/bok/getting-started-with-p5js-making-interactive-graphics-in-javascript-and-processing-9781457186776)
* [funprogramming](http://funprogramming.org/)
* [p5.js video tutorial](https://www.youtube.com/user/shiffman/playlists?sort=dd&view=50&shelf_id=14)

**kontakt, synpunkter, felrapportering, förslag till exempelkod**

* janchrister.nilsson snabela gmail.com

**Installation**

* download [node.js](https://nodejs.org/en/download)
* Skriv in följande kommando i ett terminalfönster

npm install -g coffee-script

**Så här arbetar du**

* Använd valfri texteditor t ex Sublime eller Notepad.
* Kopiera [mallen](https://github.com/ChristerNilsson/000-Mall) för varje nytt projekt.
* Klicka på transpile.bat. Om det piper: läs transpileringsfel här.
* Den eller de javascriptfiler du använder måste anges i index.html
* Dubbelklicka på index.html. Då startar applikationen.
* Vid problem: Tryck på F12 i Chrome och läs eventuella felmeddelanden.
* Du kan skriva ut värden till F12-console med print.