

# TDP007: Konstruktion av datorspråk

## Utvecklarblogg Seminarieuppgift 1

Författare

Jim Teräväinen, [jimte145@student.liu.se](mailto:jimte145@student.liu.se)  
Viktor Rösler, [vikro653@student.liu.se](mailto:vikro653@student.liu.se)

# 1 Reflektioner kring hur det är att lära sig ett nytt språk

## 1.1 Hur har ni gått tillväga för att lära er Ruby?

Igenom att ta del av föreläsningsmaterialet, testa olika funktioner och läsa dokumentationen för Ruby så har vi börjat förstå de klasser och metoder som finns i standard Ruby. Det största hjälpmedlet var att snabbt hitta och lära sig läsa dokumentationen på *[ruby-doc.org](http://ruby-doc.org)*. Därtill så har kursboken varit en bra resurs för att förstå koncept djupare då dokumentationen är väldigt rakt på sak utan längre förklaringar.

## 1.2 Vilka fel och misstag har ni gjort under tiden?

Gruppen läste delvis dokumentation för fel av version av Ruby vilket förvirrade oss. Dessutom så har Gems en egen dokumentation som vi letade i innan vi förstod skillnaden mellan *Ruby Core* och *Ruby Standard Library*.

## 1.3 Vad finns det för nya konstruktioner i Ruby som ni inte sett förut?

Möjligheten att fånga en grupp ur resultatet av en regex matchning med `$` tecknet, utan att implicit hänvisa till vilken regex matchning man plockar gruppen ur.

## 1.4 Vad finns det för konstruktioner som ni känner igen men som ser lite annorlunda ut?

Vi tycker att grundläggande saker som terminalutskrifter, loopar, if-satser, klasser, funktioner och retur värden i mångt och mycket är igenkännbara. Vi känner att vår tidigare kunskap inom programmering accelererar inläringen av Ruby när koncepten är liknande.

## 1.5 Finns det något som ni irriterar er på eller tycker om i Ruby?

Med tanke på den lilla mängden arbete vi utfört så har det varken bildats positiva eller negativa känslor kring någon aspekt av Ruby. Kort sagt är allting mycket spännande om inget annat.

# 2 Reflektioner kring de olika tekniker som ni stöter på

Hur användbart verkar detta vara?

## 2.1 Hur lätt/svårt är det att sätta sig in i?

Det har varit lätt att förstå och lätt att använda hittills. Syntaxen påminner tillräckligt mycket det vi hållit på med tidigare, och interpretatorn är inte så sträng med vad man kan och inte kan göra.

## 2.2 Har ni hittat alternativa källor för att ta reda på nya saker?

Vi hade inte kategoriserat de källor som vi använt som alternativa då de antingen är kursmaterial eller den officiella dokumentationen för Ruby. Vi har däremot aktivt undvikt att läsa inofficiellt material under denna uppgift för att just lära oss att hitta i dokumentationen.

## 3 Dokumentation av hur ni har tänkt när ni arbetat fram era lösningar

### 3.1 Vad arbetade ni med (i grova drag) vid varje labbpass?

Vi gjorde uppgifterna första passet, och skrev testerna + utvecklarbloggen under nästa pass. Under första passet skrevs även ett par enklare tester för att säkerställa någorlunda korrekt funktion.

### 3.2 Hur har ni tolkat uppgiften?

Av de uppgifter vi gjorde tyckte vi att de flesta inte lämnade mycket utrymme för tolkning. Undantagen till det följer nedan.

#### 3.2.1 Avsnitt 3, uppgift 6

Vi gjorde tolkningen att en person bör ha ett namn men eftersom datan ska kunna utelämnas sätts för- och efternamn till *name* respektive *surname* som standard.

#### 3.2.2 Avsnitt 3, uppgift 7

Vi tolkade uppgiften som att `n.fib` metoden ska retunera det `n:e` talet i Fibonacci-serien. Därmed är talen indexerade från 1 och inte 0.

#### 3.2.3 Avsnitt 4, uppgift 12

Vi tolkade det som att `i`, `q`, `v`, `å`, `ä`, `ö` ska vara förbjudna bokstäver. Vi baserade detta på en artikel från SVT Dalarna från 2018.

### 3.3 Vad var svårt eller lätt med uppgiften?

Uppgifterna var lätta för att Ruby var lätt att sätta sig in i, och för att uppgifterna berörde saker vi hållit på med tidigare.