

Direkt när vi började men Ruby kände jag att det var likt python. Man definierar funktioner med `def`, man behöver inte initiera variabler och det är mycket mer mänskligt språk som används i programmet

Uppgift 1

I denna uppgift valde jag att använda en *upto* funktion i *n_times*, detta då jag vill testa nya funktioner inom Ruby för att lära mig något nytt. Med *upto* har man även en måsvingar till höger som säger vad som sker varje loop. Inom Ruby fanns då en sak som kallas *block* och *block.call*, och så jag tolkar det så tar den allt som ifrån block samt "gör" det varje iteration.

Class är något jag inte använt tidigare inom python men är väldigt bekant med ifrån C++.

Med när man skapar ett Repeat-object ska den initieras med en siffra, denna siffra används i funktionen *each* som tar siffran samt ett block och kallar på den tidigare funktionen *n_times*.

Uppgift 2

I denna uppgift körde jag tyvärr bara en vanlig for loop då jag fastnade med vilket annan typ av loop som gick att använda. Det min funktion gör är att ta det första elementet i en Array och sätter den till *current*. Ifall något annat element är längre (fler karaktärer) än *current* kommer den bli *current*. När loppen är klar så returneras *current*.

Själva algoritmen för denna uppgift är given och kan ändras genom att ändra på ">"

Uppgift 3

Denna uppgift var lite udda då jag och det jag satt med tolkade uppgiften olika. Jag tolkar uppgiften som att *n1* och *n2* är de två första siffrorna i fibonaccis sekvens och att siffran man skriver till funktionen ska ta fram den siffran som är på det steget i sekvensen. Alltså 1:1, 2:1, 3:2, 4:3, 5:5, 6:8 ... Jag tolkar det även som att siffran som står innan är irrelevant men jag kan tänka mig att den skulle kunna användas.

Uppgift 4

Denna uppgift ska man rotera en Array (flytta alla variabler/element i en Array/lista att flytta ett steg till vänster och ifall det är först i listan rör det sig till slutet. Då Ruby redan har en funktion (*.rotate*) var det inte mycket jag behövde göra. Jag satte parametern till *n=1* vilket ger "n" ett standard värde samt att jag använder "self" i funktionen som fungerar likt C++ "this".

Uppgift 5

Denna uppgift var det givet hur man tog ut text från en fil (vilket var ganska mycket då det var all html kod för en sida). Jag behövde därför endast skapa en funktion som tar in en variabel (string) och med hjälp av regex få fram alla html-taggar.

Uppgift 6

Så som jag tolkar denna uppgift ska jag ta in en string till en funktion och sedan med hjälp utav Regex, ta fram 3 bokstäver (som inte är Q, I eller V) samt 3 siffror. Ifall strängen innehåller det jag är ute efter, så skrivs endast Regnummret ut (3 bokstäver, 3 siffror). Om inget Regnummer hittas eller ifall det innehåller Q, I eller V så får användaren "false".

Jag hade problem med denna uppgift när jag skulle skriva Regex sekvensen då jag inte hittade ett bra sätt att ignorera (Q I V), i början skrev jag `^[QIV][A-Z]{3}` men problemet med denna är att den tar ut ALLT som inte är Q, I eller V, innan 3 bokstäver. Alltså tar den in alla symboler, siffror och även mellanslag. Det var mycket fram och tillbaka men tyvärr hittade jag inget fint sätt att lösa det på och skrev `[A-HJ-PR-UW-Z]` vilket undviker Q, I och V.

Uppgift 7

Jag valde en uppgift som räknar framkomsten av ett ord i en string. Min fråga är lite lik uppgift 5 men använder en annan funktion då man nu läser in en fil och inte en hemsida. Jag kände att detta var en vettig uppgift då detta är något man absolut bör kunna inom programmering och hittills alltid annars använt i andra språk. Dock var det lite svårare att använda texten när man väl tagit ut den. Hade förväntat mig att jag skulle kunna *loopa* igenom en string som i python men behövde *splita* strängen innan jag kunde *loopa* igenom den. För output av variabler med strängar behövde man även använda `"#{text}"`. Detta var lite oväntat men även något som är bra att lära sig nu i början.

Jag märkte även när jag gjorde testar att min funktion inte har mycket fel sökning, utan ifall den inte hittar en match så ger den bara *"no match"* även ifall det är en lista, integer eller float.