

Willkommen zur Präsentation: "Schleifen mit Julia"

Ein simpler For-Loop

Man hat einen Vektor, welchen man mit den Zahlen von 1-100 befüllen möchte. Man könnte dies nun manuell für jede einzelne Zahl machen. Das bedeutet aber einen grossen Aufwand und glücklicherweise lässt sich diese Aufgabe mit Schleifen sehr elegant und schnell erledigen.

```
In [2]: for i=1:5
          println(i)
        end

1
2
3
4
5
```

Dieses Konzept kann man nun für weitere Anwendungen brauchen

Die Zinseszinsformel

- $S = P(1 + \frac{r}{n})^{(n*t)}$

Die Parameter dieser diskreten Formel sind:

- S = Der Wert nach t Perioden
- P = Anfangsinvestment (Principal)
- r = Zinssatz
- n = Zinszahlungen pro Jahr
- t = Laufzeit (in Jahren)

Kommen wir nun zur Implementierung dieser Formel

```
In [7]: function compoundInterestCalculation(principal, rate, interestPayPerYear, time)
          #The variables
          compoundedAmountArray = Array{Float64, time}()

          #The for loop is used for the time iteration (such that each year is one step)
          # At the same time, the compounded value is written into an array
          for i = 1:(time)

              compoundAmount = principal*(1+rate/interestPayPerYear)^(i*interestPayPerYear)

              compoundedAmountArray[i] = compoundAmount

          end
          return compoundedAmountArray
        end
```

```
Out[7]: compoundInterestCalculation (generic function with 1 method)
```

Für die Funktion werden nun folgende Werte verwendet:

- Anfangssaldo = 20'000 CHF
- Zinssatz = 6.0%
- Zinszahlungen pro Jahr = 4
- Dauer in Jahre = 40

```
In [8]: using Plots

#Parameter
startAmount = 20000

interestRate1 = 0.12
interestRate2 = 0.06
interestRate3 = 0.01

interestRate4 = -0.0075
paymentPerYear = 4
years = 20
#Parameter section end

#Positive interest rates of: 12%, 6% and 1%
values3 = compoundInterestCalculation(startAmount, interestRate1, paymentPerYear, years)
values2 = compoundInterestCalculation(startAmount, interestRate2, paymentPerYear, years)
values1 = compoundInterestCalculation(startAmount, interestRate3, paymentPerYear, years)

#Negative interest rates of -0.75%
values4 = compoundInterestCalculation(startAmount, interestRate4, paymentPerYear, years)

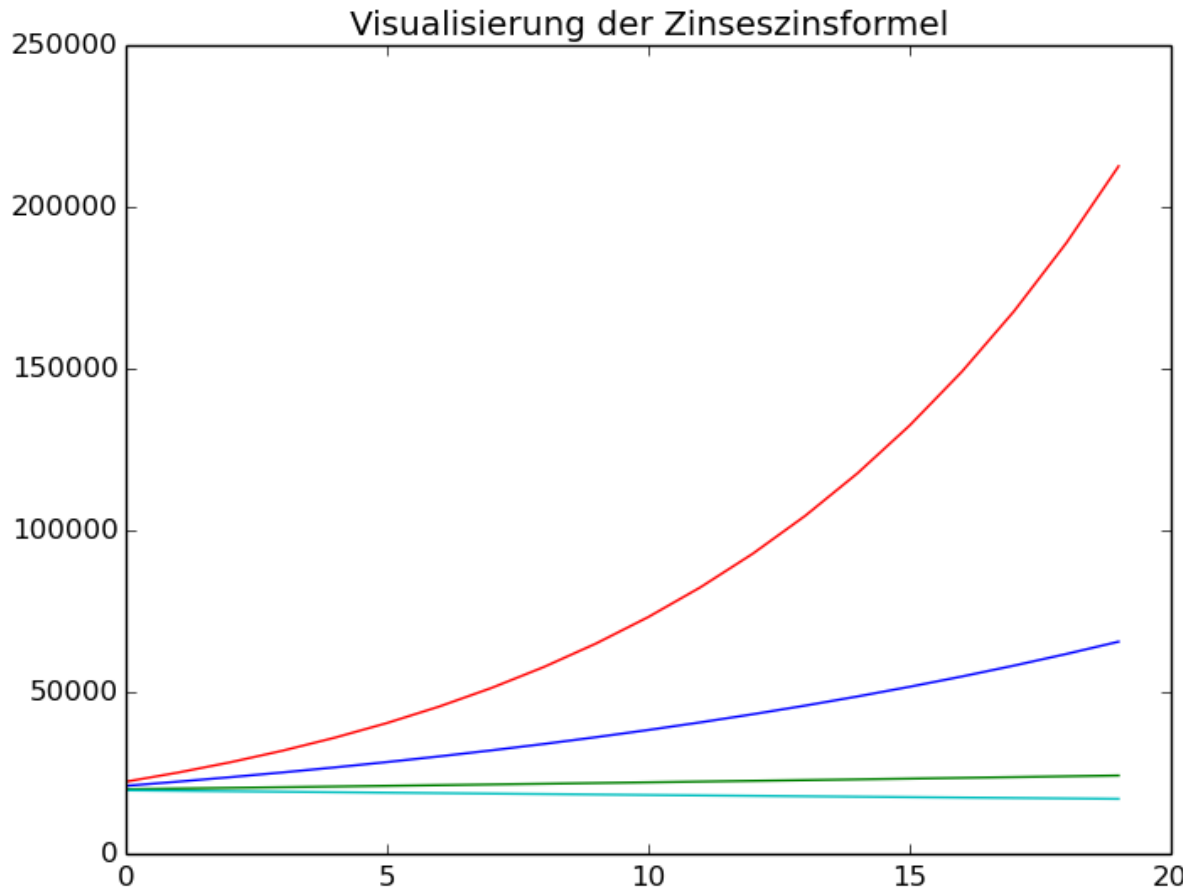
#Plot the values
plot(values)
plot(values2)
plot(values3)
plot(values4)
title("Visualisierung der Zinseszinsformel ")

print("Betrag mit einem anständigen Zins: ")
println(values3[years])

print("Der Minimalbetrag mit Negativzins: ")
println(values4[years])

print("Die Differenz: ")
difference = values3[years] - values4[years]
println(difference)
```

Betrag mit einem anständigen Zins: 212817.81112860437
 Der Minimalbetrag mit Negativzins: 17211.735927756177
 Die Differenz: 195606.0752008482



Bonusmaterial

Ein Vektor mit den Zahlen von 1-100, welcher mit einer Schleife befüllt wird

```
In [6]: Vektor = Array{Float64,100}
Vektor[1] = 1

for i = 1:100
    Vektor[i] = i
end

print("Der Inhalt des Vektor: ")
print(Vektor)
```

```
Der Inhalt des Vektor: [1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,16.0,17.0,18.0,19.0,20.0,21.0,22.0,23.0,24.0,25.0,26.0,27.0,28.0,29.0,30.0,31.0,32.0,33.0,34.0,35.0,36.0,37.0,38.0,39.0,40.0,41.0,42.0,43.0,44.0,45.0,46.0,47.0,48.0,49.0,50.0,51.0,52.0,53.0,54.0,55.0,56.0,57.0,58.0,59.0,60.0,61.0,62.0,63.0,64.0,65.0,66.0,67.0,68.0,69.0,70.0,71.0,72.0,73.0,74.0,75.0,76.0,77.0,78.0,79.0,80.0,81.0,82.0,83.0,84.0,85.0,86.0,87.0,88.0,89.0,90.0,91.0,92.0,93.0,94.0,95.0,96.0,97.0,98.0,99.0,100.0]
```

In []: