

# Julia-Testdatei

November 1, 2020

## 1 GPT-Übungen

### 1.1 Aufgabenblatt 1

#### 1.1.1 Nr. 1 - booleans

```
In [30]: a = false
         b = false
         c = true
```

```
Out[30]: true
```

```
In [29]: #a) python: (a and a) or (b and b)      wird in Julia zu:
         println( (a && a) && (b && b))
         # wird zu:
         println( a || b )
```

```
false
false
```

a && b ist falsch, sobald a falsch ist (b nicht mehr getestet)  
a || b ist richtig, sobald a richtig ist (b nicht mehr getestet)  
println() schreibt je neue Zeile, print() hat kein default-Ende

```
In [36]: #b) python: ( a and b) or ( a and c)
         (a && b) || (a && c)

         # c) not (a < b) or (a == b)
         !(a < b) || (a == b)
```

```
Out[36]: true
```

umgekehrter boolean durch !

```
In [43]: #d) (not (a < b) and not (a > b) )
         a = 1
         b = 1
         println( (!(a < b) && ! (a > b) ) )
         print(a == b)
```

```
true
true
```

```
In [50]: #e) (not (a and b) and (a or b)) or ((a and b) or not (a or b))
         a = true
         b = false
         println( (! (a && b) && (a || b)) || ((a && b) || ! (a || b)) )#immer wahr

true
```

### 1.1.2 Nr. 2 - Bedingung

```
In [63]: a = parse(Int64, readline())
         b = parse(Float64, readline())

         #
         if a % b == 0 || b % a == 0
             println("True")
         else
             println("False")
         end
```

```
stdin> 1
stdin> 5
True
```

readline() für Benutzereingabe, parse und Format nötig für Umwandlung in Zahl-Datentyp

### 1.1.3 Nr. 3 - Eingabe, Ausgabe

```
In [2]: zeit = parse(Float64, readline()) #in Sekunden
         print(zeit, " Sekunden sind ")
         anno = div(zeit, 3600*365*24)
         zeit = zeit % (3600*24*365)
         days = div( zeit, 3600*24)
         zeit = zeit % (3600*24)
         h = div( zeit, 3600)
         zeit = zeit% 3600
         min = div( zeit, 60)
         sec = zeit % 60

         print(anno, " Jahr(e), ", days, " Tag(e), ",
               h, " Stunde(n), ", min, " Minute(n) und ", sec, " Sekunde(n).")

stdin> 7464643
7.464643e6 Sekunden sind 0.0 Jahr(e), 86.0 Tag(e), 9.0 Stunde(n), 30.0 Minute(n) und 43.0 Sekun
```

div(zahl1, zahl) für Ganzzahldivision, % für Rest der Ganzzahldivision

## 1.2 Aufgabenblatt 2

### 1.2.1 Nr. 1 - sortieren, einlesen, schleife

```
In [17]: A = []
```

```
    for i in 1:3
        append!(A, [readline()])
    end

    #print(A, " ", sort(A))
```

```
stdin> wer
```

```
stdin> werasd
```

```
stdin> sdfwer
```

```
Any["wer", "werasd", "sdfwer"] Any["sdfwer", "wer", "werasd"]
```

```
In [19]: #Möglichkeit 1:
```

```
    if A[3] < A[1] && A[3] < A[2]
        println(A[3])
        if A[1] < A[2]
            println(A[1])
            println(A[2])
        else
            println(A[1])
            println(A[2])
        end
    end
    if A[1] < A[3] && A[1] < A[2]
        println(A[1])
        if A[0] < A[2]
            println(A[3])
            println(A[2])
        else
            println(A[3])
            println(A[2])
        end
    end
    if A[2] < A[3] && A[2] < A[2]
        println(A[2])
        if A[3] < A[1]
            println(A[3])
            println(A[1])
        else
            println(A[1])
            println(A[3])
        end
    end
end
```

```
sdfwer  
wer  
werasd
```

```
In [20]: #Möglichkeit 2:  
        for i in sort(A)  
            println(i)  
        end
```

```
sdfwer  
wer  
werasd
```

Arrayindizes startend bei 1, Array sortieren mit `sort(array)`;  
`append!([vorhandener Teil], [neu])`

### 1.2.2 Nr. 2 - Aufforderung, Eingabe, Ausgabe

```
In [23]: println("Dieses Programm berechnet die Summe der eingegebenen Zahlen. Eingabe 0 beei  
        ges = 0  
        zahl = 1  
        while zahl != 0  
            zahl = parse( Int64, readline())  
            ges = ges + zahl  
        end  
  
        println("Summe: ", ges)
```

Dieses Programm berechnet die Summe der eingegebenen Zahlen. Eingabe 0 beendet das Programm.

```
stdin> 1  
stdin> 2  
stdin> 3  
stdin> 0  
Summe: 6
```

while-Schleife analog zur for schleife, Ende mit end, ansonsten eingerückt

### 1.2.3 Nr. 3 - Zahlenreihe ohne if

```
In [24]: z = parse(Int64, readline())  
  
        for i in 1:10  
            println(i*z)  
        end
```

```
stdin> 3
3
6
9
12
15
18
21
24
27
30
```

range durch Anfangszahl:Endzahl (je inklusive)

### 1.3 Blatt 3

#### 1.3.1 Nr. 1 - Teiler

```
In [25]: z = parse(Int64, readline())
```

```
    for i in 1:(z-1)
        if z % i == 0
            println(i)
        end
    end
```

```
stdin> 12
1
2
3
4
6
```

#### 1.3.2 Nr. 2 - Array umkehren

```
In [46]: # Array anlegen:
```

```
    z = 3
    A = []
    for i in 1:10
        append!(A, [i*z])
    end
    print(A)
```

```
Any[3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
```

```
In [48]: for i in 1:div(length(A),2)
           t = A[i]
```