

## S3

February 19, 2023

Skrifið fall `telja(s)` sem byrjar á nota `s.split()` til að búa til lista af einstökum orðum í `s` og telur síðan hve margir stafir eru í hverju orði. Fallið á að skila lista með þessari talningu. Þannig ætti `telja("Afi minn og amma mín")`

að skila listanum `[3,4,2,4,3]`. Prófið líka að telja stafi í nafninu ykkar (t.d. `telja("Kristján Jónason")`). Líklega borgar sig að búa fyrst til tóman lista og fara svo í lykkju og bæta nýrri talningu afast í listann í hverri umferð (t.d. með `append`).

```
[4]: def telja(s):  
    s = s.split(" ")  
    listi = []  
    for i in s:  
        listi.append(len(i))  
    return listi  
  
print(telja("Afi minn og amma mín"))  
print(telja("Magnús Daníel Budai Einarsson"))
```

```
[3, 4, 2, 4, 3]
```

```
[6, 6, 5, 9]
```

Búið til fall `hropa(s)` sem skrifar `s` út með stórum stöfum og tveimur upphrópunarmerkjum. Hrópið svo nafnið ykkar :)

```
[7]: def hropa(s):  
    return s.upper() + "!!"  
  
print(hropa("Magnus Daniel Einarsson"))
```

```
MAGNUS DANIEL EINARSSON!!
```

Til að athuga hvort stafur `c` sé hástafur má spyrja: `c == c.upper()`. Skrifðu fall `stórir(s)` sem skilar lista af rökildum með `i`-ta gildið satt ef `i`-ta orðið í `s` byrjar á stórum staf. Prófið með sjálfvöldu dæmi og sýnið niðurstöðuna.

```
[22]: def upper(s):  
    return [i[0].isupper() for i in s.split()]  
print(upper("Magnus Daniel Einarsson"))
```

[True, True, True]

1. Eftirfarandi formúlu má nota til að nálga afleiðu falls í punkti a:  $f'(a) \approx \frac{f(a+h)-f(a-h)}{2h}$  þar sem  $h$  er lítil tala. Skriðið fall `toldiff(f, a, h)` sem reiknar slíka nálgun og skilar henni. Prófið með  $f(x) = \sqrt{x}$ ,  $a=4$  og  $h = 10^{-4}$  (sem ætti að skila tölu sem er nálægt  $\frac{1}{2\sqrt{4}}$ ).
2. Skriðið nú töflu yfir nálgunina fyrir  $h = 10^{-k}$ ,  $k = 1, \dots, 10$ , og látið fylgja með dálk sem sýnir hve skekkjan er mikil. Merkið með \* línuna þar sem skekkjan er minnst.

```
[44]: from math import sqrt
# f(a+h)-f(a-h)/2h

def toldiff(f,a,h):
    return (f(a+h)-f(a-h))/(2 * h)

def f(x):
    return sqrt(x)

print(f"Svarið við lið a er {toldiff(f,4,10**-4)}\n")
k=1
k2=[]
while k<=10:
    k2.append(toldiff(f,4,10**-k))
    k+=1
minimum = min(k2, key=lambda x:abs(x-0.25))
print("k          útreikningur          mismunur")
print("-"*50)
for k,i in enumerate(k2):
    if i==minimum:
        stringI = str(i)
        stringI += "*"
        print(f"{k+1:<3}{stringI:^30}{i-0.25:>10}")
    else:
        print(f"{k+1:<3}{i:^30}{abs(0.25-i):>10}")
```

Svarið við lið a er 0.2500000000205116

k	útreikningur	mismunur
-----		
1	0.25001953659254283	1.9536592542834796e-05
2	0.2500001953130382	1.953130381870949e-07
3	0.25000000195318783	1.953187833692027e-09
4	0.2500000000205116	2.0511592424554692e-11
5	0.2500000000016378*	1.637801005927031e-12
6	0.2499999999239222	7.607781071783393e-11
7	0.24999999959085528	4.091447181053809e-10
8	0.249999987378402	1.2621597988982103e-08
9	0.25000002068509275	2.0685092749772593e-08

10            0.25000002068509275            2.0685092749772593e-08

Hugsum okkur að `ísl_ens` sé uppflettitafla sem geymir íslensk-enska orðabók. Hún gæti t.d. innihaldið þörin:

```
"reipi" → "rope"
"hús" → "house" og
"rauður" → "red".
```

Ef við vilum búa til ensk-íslenska orðabók gætum við snúið töflunni við og fengið þörin “rope” → “reipi”, “house” → “hús” o.s.frv.

1. Skrifðu fall `snúavið(U)` sem snýr uppflettistöflu við á þennan hátt og skilar viðsnúnu töflunni. Gerið ráð fyrir að taflan `U` hafi hvergi sama gildi fyrir tvo mismunandi lykla. Prófið með orðaskránni að framan og búið til `ens_ísl`.
2. Skrifðu nýja útgáfu af fallinu, `snúavið2(U)`, sem er ekki með slíkri einkvæmnitakmörkun. Það á að skila nýrri uppflettistöflu `V` sem er þannig að ef `g` er gildi svarandi til tveggja mismunandi lykla,  $U[x] = U[y] = g$  þá á gildi `V` fyrir lykilinn `g` að vera listi með `x` og `y`,  $V[g] = [x,y]$ . Bætið nú tveimur þörum við `ísl_ens`:

“tómarúm” → “vaccum” “ryksuga” → “vaccum”

og prófið `snúavið2`.

```
[22]: def snuavid(U):
       return {value: keyname for keyname, value in U.items()}

ordabok = {"reipi": "rope", "hús": "house", "rauður": "red"}

print(snuavid(ordabok))
```

```
{'rope': 'reipi', 'house': 'hús', 'red': 'rauður'}
```

```
[23]: def snuavid2(U):
       L={}
       for keyname, value in U.items():
           if value in L:
               L[value].append(keyname)
           else:
               L[value] = [keyname]
       return L

ordabok = {"reipi": "rope", "hús": "house", "rauður": "red", "tómarúm": "vaccum",
           ↪ "ryksuga": "vaccum"}

print(snuavid2(ordabok))
```

```
{'rope': ['reipi'], 'house': ['hús'], 'red': ['rauður'], 'vaccum': ['tómarúm',
'ryksuga']}
```