

s2

January 25, 2022

1 2B

1.1 Simpsons-regla

forrit sem nálgar heildi með Simpsons-reglunni

```
[ ]: from math import sin,exp

def f(x): return sin(x)/x      # *
def g(x): return exp(x)        # **

def simpsons(f,a,b,n):
    deltax = (b-a)/n
    flatarmal = f(a)+f(b)
    xi = a+deltax
    for i in range(n-1):        # nota i+1 inn í formúlunni því er ekki að
    ↪notast 0
        xi = a+((i+1)*deltax)
        if (i+1) % 2 == 0:
            flatarmal+=(2*f(xi))
        else:
            flatarmal+=(4*f(xi))

    return flatarmal*(deltax/3)

print(simpsons(f,1,2,4))
print(simpsons(g,0,1,4))
```

0.6593312109452117

1.718318841921747

2 3A

2.1 hitastigum breytt

Forrit sem býr til töflu með gildum fyrir gráður bæði í celcius og fahrenheit á bilinu [-30°C - 50°C] með 5°C bili

```
[ ]: def hitamunur():
    tafla = '°C      °F\n'
    tafla += '-----\n'
    for i in range(16):
        c = -30+(5*i)
        f = ((9*c)//5)+32

        tafla += f'{c:>3} {f:>3}\n'
    print(tafla)
hitamunur()
```

°C	°F
-30	-22
-25	-13
-20	-4
-15	5
-10	14
-5	23
0	32
5	41
10	50
15	59
20	68
25	77
30	86
35	95
40	104
45	113

3 3B

3.1 lograr og veldi

forrit sem reiknar fjóra logra

```
[ ]: from math import log, log10, log2, exp

print(f'1. ln(1) = {log(1)}')
print(f'2. ln exp(3) = {log(exp(3))}')
print(f'3. log10(1000) = {log10(1000)}')
print(f'4. log2(8) = {log2(8)}')
```

1. $\ln(1) = 0.0$
2. $\ln \exp(3) = 3.0$
3. $\log_{10}(1000) = 3.0$
4. $\log_2(8) = 3.0$

4 3C

4.1 rætur, lograr, kvaðröt

finnur rót, ln og annað veldi talnanna 1-10, fyrst með for-lykkju svo með while-lykkja

```
[ ]: from math import sqrt, log
print("for-lykkja")
for i in range(10):
    tala = i+1
    print(f'{tala:2} rót: {sqrt(tala):3.2f}, ln: {log(tala):3.2f}, x²: {tala**2:
↪3}')

print("\nwhile-lykkja")
i = 1
while (i <= 10):
    print(f'{i:2} rót: {sqrt(i):3.2f}, ln: {log(i):3.2f}, x²: {i**2:3}')
    i+=1
```

for-lykkja

```
1 rót: 1.00, ln: 0.00, x²: 1
2 rót: 1.41, ln: 0.69, x²: 4
3 rót: 1.73, ln: 1.10, x²: 9
4 rót: 2.00, ln: 1.39, x²: 16
5 rót: 2.24, ln: 1.61, x²: 25
6 rót: 2.45, ln: 1.79, x²: 36
7 rót: 2.65, ln: 1.95, x²: 49
8 rót: 2.83, ln: 2.08, x²: 64
9 rót: 3.00, ln: 2.20, x²: 81
10 rót: 3.16, ln: 2.30, x²: 100
```

while-lykkja

```
1 rót: 1.00, ln: 0.00, x²: 1
2 rót: 1.41, ln: 0.69, x²: 4
3 rót: 1.73, ln: 1.10, x²: 9
4 rót: 2.00, ln: 1.39, x²: 16
5 rót: 2.24, ln: 1.61, x²: 25
6 rót: 2.45, ln: 1.79, x²: 36
7 rót: 2.65, ln: 1.95, x²: 49
8 rót: 2.83, ln: 2.08, x²: 64
9 rót: 3.00, ln: 2.20, x²: 81
```

10 rót: 3.16, ln: 2.30, x^2 : 100

5 3D

5.1 rúmmál kúlu

3 föll til þess að reikna ýmis gildi, rúmmál kúlu, radíus hrings og þyngd útfrá eðlismassa og rúmmáli

```
[ ]: from math import pi
def volSphere(radius):
    Volume = (4/3)*pi*(radius**3)
    return Volume

def radCircle(circumference):    #c fyrir circumference
    rad = circumference/(2*pi)
    return rad

def getMass(density,volume):
    density = density * 1000 # breyta úr g/cm³ yfir í kg/m³
    mass = density*volume
    return mass

# test prints
print(f'{volSphere(2):.2f}')
print(f'{radCircle(10):.2}')
print(getMass(19.3,0.008))

radEarth = radCircle(4000000000)
volEarth = volSphere(radEarth)
massEarth = getMass(5.5,volEarth)/1000
print(f"Massi jarðarinnar er um það bil {massEarth:.2} tonn")
```

33.51

1.6

154.4

Massi jarðarinnar er um það bil 5.9e+24 tonn