


linux

linux: käyttäjän oikeudet

- ▶ Käyttäjällä, **username**, on käyttöoikeus rajattuun **levytilaan**
- ▶ `du -h /home/username/` tulostaa **käytetyn** levytilan. Yhteenvedon antaa `du -h /home/jetsu/ - --summarize`
- ▶ `df` tulostaa sekä **vapaan** että **käytetyn** levytilan. Käytännöllinen **alias** on `alias Tila='df -h /home/username/'`
- ▶ Käyttäjillä rajatut **oikeudet** suorittaa komentoja ja käsitellä tiedostoja
- ▶ **linux**:ssa käyttäjä kuuluu ainakin yhteen ryhmään. **groups** kertoo mihin
- ▶ Käyttäjälle: **u**, Ryhmälle: **g** & Muille: **o** on erilaiset oikeudet seuraaviin:
Luku: **r**, Kirjoitus: **w** & Suoritus: **x**
- ▶ Listaa tiedostosi komennolla `ls -ls`

linux: käyttäjän oikeudet

- ▶ Esimerkiksi `-rwxr-x- -x` tarkoittaa:
 - käyttäjällä on **rwx** oikeudet
 - ryhmällä on **r-x** oikeudet
 - muilla on **--x** oikeudet
- ▶ **d**:llä alkava tulostus (Esim: `drwxr-x- -x`) tarkoittaa, että kyseessä on hakemisto
- ▶ Oikeuksia voi **muuttaa** komennolla `chmod [ugoa] (+-) [rwx] nimi`
- ▶ **(u)ser**, **(g)roup**, **(o)thers**, **(a)ll**
(**a** on kaikki **u**, **g** ja **o** yhdessä)
- ▶ **+** lisää ja **-** poistaa oikeuksia
- ▶ **(r)ead**, **(w)rite**, **e(x)ecute**
- ▶ **Esimerkki 1**: `chmod g+rwx nimi`
lisää ryhmällä **rwx**-oikeudet tiedostoon tai hakemistoon **nimi**
- ▶ **Esimerkki 2**: `chmod go-wx nimi`
poistaa ryhmältä ja muilta **wx**-oikeudet tiedostoon tai hakemistoon **nimi** 

L^AT_EX: kuvat

Figure 1: First caption text

Label `afig` is used to refer to Fig. 1. Command `\begin{figure}[t]` is used to put it on **top**.

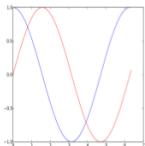


Figure 2: Second caption text

Curves in Fig. 2 are referred to with label `bfig`. Command `\begin{figure}[h]` puts figure **here** between text.

L^AT_EX: kuvat

► Tiedostot

Kuvamalli2.tex

SineCurves.eps

kotisivulta

pdf_latex Kuvamalli2

evince Kuvamalli2.pdf

```

\begin{figure}[t]
\begin{center}
\includegraphics[width=2.5cm,
height=2.5cm]{SineCurves.eps}
\end{center}
\caption{First caption text}
\label{afig}
\end{figure}

```

► **Opettele** tiedostoista

Kuvamalli1.tex ja

Kuvamalli2.tex

muut uudet komennot

Aliohjelmat

Funktio \approx Aliohjelma

- ▶ **Funktio** “syö” jotain sisään. Tämä “**input**” voi olla muuttujia, tiedostoja, jne..
- ▶ **Funktio** palauttaa jotain syömänsä perusteella. Tämä “**output**” voi olla muuttujia, jne..
- ▶ **Funktio** voi myös muokata “**input**” tietoa. Ei järkevää: kts **Aliohjelmamalli3.py**
- ▶ **Funktioiden** tarkoitus on jakaa ja jäsentää ohjelmaa, sekä vähentää koodin toistoa
- ▶ **Funktiot** on loogisinta määrittellä ohjelman alussa, mutta yleensä toimivat muuallakin
- ▶ “**Muuallakin**” = **Funktio** oltava määritetty ennen kuin sitä kutsutaan/käytetään
- ▶ Kopioidaan kotisivulta ohjelmat **AliOhjelmamalli1.py**, **AliOhjelmamalli2.py** ja **AliOhjelmamalli3.py**

- ▶ **python** aliohjelman rakenne
`def funktion(a,b,...):`
Sisennys: **Aliohjelman** komennot ...
Sisennys: `return A,B,...`
Input: Muuttujat `a,b, ...`
Output: Muuttujat `A,B, ...`



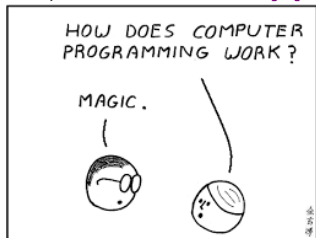
python

```
# Tama on ohjelmani
# AliOhjelmamalli2.py
# Aliohjelma alkaa ====
def puolita(a):
    puolet=a/2.0
    return a,puolet
# Aliohjelma loppuu ===
b=4.0
e=puolita(b) # Kokeilu 1
print('Input:_b_=', b)
print('Output:e_=', e)
print('Output:e[0]_=', e[0])
print('Output:e[1]_=', e[1])
print('type(e)_=', type(e))
```

▶ python3 AliOhjelmamalli2.py tulostaa

```
Input: b = 4.0
Output:e = (4.0, 2.0)
Output:e[0] = 4.0
Output:e[1] = 2.0
type(e) = <class 'tuple'>
```

- ▶ Tämä on **väärä** tai ainakin **harhaanjohtava aliohjelman rakenne**
 - Kaksi **output** **a,puolet** muuttujaa korvattu yhdellä muuttujalla **e** (Kokeilu 1)
 - Silti **ohjelma ei kaadu!**
 - **Output** muuttuja **e** sisältää **molemmat** muuttujat **a** ja **puolet**
 - **e** on **tuple** eli sitä ei voi muuttaa
 - Hyvää on ,että **Input** muuttujaa **b** ei muuteta, s.o. arvot samat: **b=e[0]=4.0**



python

python

```

# Tama on ohjelmani
# AliOhjelmamalli3.py
# Aliohjelma alkaa ====
def puolita(a):
    puolet=a/2.0
    a=a-9 # Kokeilu 2
    jotain=a/9. # Kokeilu 3
    return a,puolet
# Aliohjelma loppuu ===
b=4.0
c,d=puolita(b)
print('Input: b=', b)
print('Output: c=', c)
print('Output: d=', d)
print('jotain=', jotain)
▶ python3 AliOhjelmamalli3.py
Input: b = 4.0
Output: c = -5.0
Output: d = 2.0
Traceback (most recent call last):
  File "AliOhjelmamalli3.py", line 15, in <module>
    print('jotain =', jotain)
NameError: name 'jotain' is not defined

```

- ▶ Tämäkin on **väärä** tai ainakin **harhaanjohtava aliohjelman rakenne**
 - **Input** muuttujaa **a** muutetaan arvoon **a=a-9** (Kokeilu 2)
 - **Output** muuttuja **puolet** lasketaan **ennen** kuin muutetaan **a**:n arvo
 - ⇒ – **Input** muuttujan **b=4.0** arvo muuttuu **Output** arvoksi **c=-5.0**
 - Muuttujan **puolet** arvo on **a/2=2.0**, mutta se **ei ole** ole **puolet** arvosta **d=2.0**
 - Muuttuja **jotain** määritellään aliohjelman sisällä (Kokeilu 3)
 - **jotain** ei ole rivillä **return, ...**
 - ⇒ Pääohjelmassa ei muuttujaa **jotain**
- Kokeilu 3: Voit käyttää aliohjelmissa **mitä tahansa muuttujaa**. Se päättyy pääohjelmaan vain **Output** riviltä **return**

- **Summia.py** ohjelmalla voi **tarkistaa** pitävätkö alla olevat kaavat paikkansa millä tahansa n ja x arvolla

$$\sum_{k=1}^n k = n(n+1)/2$$

$$\equiv \mathbf{A} = \mathbf{B}$$

$$\sum_{k=1}^n k^2 = (n+1)(2n+1)/6$$

$$\equiv \mathbf{C} = \mathbf{D}$$

$$\sum_{k=1}^n \sin^2(kx) = \frac{n}{2} - \frac{\cos[(n+1)x] \sin(nx)}{2 \sin x}$$

$$\equiv \mathbf{E} = \mathbf{F} = \mathbf{ny1-ny2/n3}$$

n = int = skalaari

k = array = vektori

$\mathbf{np.sum(k)}$ = float = skalaari

x = float = skalaari

$\mathbf{np.sin(k*x)}$ = array = vektori

python Summia.py tulostaa

```
Anna n = 7
n= 7.0 A= 28.0 ,B= 28.0
n= 7.0 C= 140.0 ,D= 140.0
Anna x = 3.2
n= 7.0 x= 3.2 E= 0.459781864822 ,F= 0.459781864822
```

python: Summan `numpy.sum` käyttö

```
# Tama on ohjelmani Summia.py
# _____
import numpy as np
n=input('Anna_n_=_')
n=float(n) ;
k=1.+np.arange(n)
A=np.sum(k) # p3: 1. CE 333
B=n*(n+1.)/2.
print('n=',n,'A=',A,'B=',B)
C=np.sum(k**2) # p3: 2. CE 333
D=n*(n+1.)*(2*n+1.)/6.
print('n=',n,'C=',C,'D=',D)
x=input('Anna_x_=_')
x=float(x)
E=np.sum((np.sin(k*x))**2.) # p37: 1.
ny1=n/2.
ny2=np.cos((n+1.)*x)*np.sin(n*x)
ny3=2.*np.sin(x)
F=ny1-ny2/ny3
print('n=',n,'x=',x,'E=',E,'F=',F)
```


Aliohjelmat

- Kopioi kotisivulta `RayleighAliOhjelma1.py`

```
# Kommenttirivi: Tama on python ohjelmani RayleighAliOhjelma1.py
import numpy as np          # numpy impoitoitu
def rayleightest( t, f):    # Funktio alkaa
    pi=np.pi; theta=2.0*pi*f*t ; n=len(t)
    z=(1.0/n)*(sum(np.cos(theta))**2+sum(np.sin(theta))**2)
    return n,z

# -----
import os ; os.system('clear') # Tyhjennetaan naytto
t=10.*np.random.sample(10)    # Satunnaisajat 0 < t_i < 10 (n=10)
f=1.                          # Frekvenssi on f=1.
n,z=rayleightest(t,f)        # Aliohjelman soveltaminen
print('n=',n)               # n
print('z=_',z)               # z
```

- `np.random.sample(10)` luo kymmenen satunnaislukua `t` väliltä $[0, 1]$

- Kerrotaan `10*` \Rightarrow Luvut `t` väliltä $[0, 10]$

- Tulos on aina erilainen, koska aina arvotaan uudet aikapisteet `t`

```
n= 10
z= 0.949631396598
```

- Käytä tätä ohjelmaa `RayleighAliOhjelma1.py` mallina seuraavalla sivulla kuvatussa laskuharjoituksessa

Laskuharjoitusten aliaohjelmat, kuva: @www.pinterest.com

- **Laskuharjoitus:** Tee **ensin** aliohjelma `tau=aliohjelma1(t, f)` joka laskee

$$\tau = \frac{1}{4\pi f} \operatorname{atan} \left[\frac{\sum_{i=1}^n \sin(4\pi f t_i)}{\sum_{i=1}^n \cos(4\pi f t_i)} \right]$$

Input (aliohjelma1)

- Havaintoajat **t**, jotka ovat $t_1 = 1, t_2 = 2$ ja $t_3 = 3$ (**n** = 3)
- Frekvenssi on **f** = $f = 1.41$

Output (aliohjelma1)

- Tulos pitäisi olla **tau** = $\tau \approx 0.0496$



- Tee **sitten** aliohjelma `z1, z2=aliohjelma2(t, ydot, f, tau)` joka laskee

$$z_1(f) = \left\{ \sum_{i=1}^n y'_i \cos [2\pi f (t_i - \tau)] \right\}^2$$

$$z_2(f) = \left\{ \sum_{i=1}^n y'_i \sin [2\pi f (t_i - \tau)] \right\}^2$$

Input (aliohjelma2)

- **t** ajat $t_1 = 1, t_2 = 2$ ja $t_3 = 3$
- **ydot** havainnot $y'_1 = 4, y'_2 = 5$ ja $y'_3 = 6$
- **f** frekvenssi = $f = 1.41$
- **tau** muuttuja = $\tau \approx 0.0496$

Output(aliohjelma2)

- Tulos pitäisi olla **z1** = $z_1(f) \approx 1.148$
- z2** = $z_2(f) \approx 11.856$

- On siis laadittava aliohjelma rakenne, jossa on oikeassa järjestyksessä **def**, **sisennys**, **return**