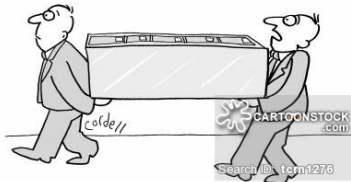


linux: rm, cp, mv (kuva:@funny-pictures.picphotos.net)

- ▶ `ls` Return listaa `työhakemiston` sisällön
- ▶ `ls *.dat` Return tulostaa kaikki `työhakemiston .dat` loppuiset tiedostot
- ▶ `*` merkki on "jokeri"
- ▶ `cp nimi1 nimi2` Return kopioi `työhakemiston` tiedoston `nimi1` `työhakemiston` tiedostoksi `nimi2`
- ▶ `mv nimi1 /home/username/hakemisto1/` Return siirtää `työhakemiston` tiedoston `nimi1` hakemistoon `/home/username/hakemisto1/` jos kyseinen hakemisto on jo olemassa
Tämän jälkeen `työhakemistossa` ei ole tiedostoa `nimi1`
- ▶ `rm nimi` Return tuhoaa `työhakemistosta` tiedoston `nimi`
- ▶ `rmdir nimi` Return tuhoaa `työhakemistosta` hakemiston `nimi`, jos hakemisto on tyhjä
- ▶ `touch nimi` Return luo `työhakemistoon` tyhjän tiedoston `nimi`
- ▶ `touch /home/username/hakemisto1/nimi` Return luo tyhjän tiedoston hakemistoon `/home/username/hakemisto1/`, jos kyseinen hakemisto on jo olemassa
- ▶ **Harjoitus 1:** Luodaan hakemistoja ja tiedostoja, jonka jälkeen tuhoataan ne



“Surely there’s an easier way of moving files?”

emacs

emacs: Peruskomentoja Pikaopas | www |

- **Kertaus:** Mikä `Ctrl` + `x`, `Ctrl` + `c` tai `Ctrl` + `x`, `Crtl` + `s` ero? Kts editorin yläpalkki: **File**
- **wget** komennolla **H2bkesken.tex** kotisivulta. Voidaan editoida miten tahansa, **koska ...?**
emacs H2bkesken.tex & (Tätä tiedostoa editoidaan **laskuharjoituksessa**)
- `Ctrl` + `s` ⇒ Kysyy **I-search**: Mitä etsit? ⇒ Vastataan **öt** `Return` ⇒ Toistamalla `Ctrl` + `s` löydät **öt** merkit **kursorista eteen päin** yksi kerrallaan ⇒ Voit lopettaa etsinnän muuttamatta tiedostoa painamalla mitä tahansa näppäimistä `↑`, `↓`, `←` tai `→`
- `Ctrl` + `r` etsii samalla tavalla tiedostosta merkkejä **kursorista taakse päin**
- Kuljeta kursori jonnekin keskelle tiedostoa **H2bkesken.tex** ja paina `Crtl` `Spacebar` ⇒ Siirry tiedostoa ylös tai alas näppäimillä `↑` tai `↓` ⇒ Osa tiedoston **H2bkesken.tex** riveistä tulee merkityksi ⇒ Paina `Ctrl` + `w` ⇒ Merkitty alue katosi ⇒ Siirrä kursori toiseen paikkaan tiedostoa ja paina `Ctrl` + `y` ⇒ Kadonnut alue palasi tähän kohtaan tiedostoa
Sama toiminto: Merkitään hiirellä ⇒ **Edit** valikosta **Copy** ⇒ Valitussa kohdassa **Paste**
Sama toiminto: Merkitään hiirellä. **Edit** valikosta **Cut** ⇒ Valitussa kohdassa `Ctrl` + `y`
Sama toiminto: Merkitään hiirellä. `←` ⇒ Valitussa kohdassa klikataan hiiren keskeltä
- **Johtopäätös:** Jokainen päättää itse, mitkä kaikki **emacs** komennot opettelee (ulkoa?)
- **Kertaus:** **emacs** ★ pieleen ⇒ Tallennus vahingossa ⇒ Poistu tallentamatta ⇒ **emacs** ★ ~ 

L^AT_EX: perusteita

L^AT_EX: perusteita (kuva:@play.google.com)

- ▶ **L^AT_EX**: ohittaa välilyönnit tai kommentit merkin % jälkeen
- ▶ Yksi tai monta tyhjää riviä ⇒ Uusi kappale
- ▶ Erikoismerkit \ # \$ % _ ^ ~ { } | varattuja. Ohjaavat ladontaa. Kaikki paitsi \ ja | voi kirjoittaa komennolla \verb|*|, missä * on erikoismerkki
Esimerkiksi \verb|*| tulostaa #
- ▶ Erikoismerkki \$ rajaa matemaattisen merkinnän
Esimerkiksi \$a^{b+c}_{d+e}\$ tulostaa a_{d+e}^{b+c} .
- ▶ Tekstin sisällä: esimerkiksi \% tulostaa %
- ▶ Jotkut komennot vaativat parametreja: Pakolliset { ja } väliin. Valinnaiset [ja] väliin
Esimerkiksi \documentclass[A4paper,10pt]{article} parametri article (Pakollinen: dokumenttiluokka) ja parametri A4paper (Valinnainen: paperikoko)
- ▶ Dokumenttiluokkia: article, report, book, proc, ... Lisäpaketteja: \usepackage[utf8]{inputenc}, ... Otsikkosivu: titlepage, notitlepage, Sivujenpuolisuus: twoside, oneside, ... Kolumnit: twocolumn, onecolumn, ...
- ▶ Voi määritellä omia komentoja. Esimerkiksi \newcommand{\omajuttu}{\AE\OE\ae} ⇒ Komento \omajuttu tulostaa **Æœ**

Back

Symbols, Acce

Symbols

& \&	- _	... \
\$ \\$	^ \^	\
% \%	~ \~	# \

Accents

ò \`o	ó \'o	ô \^o
ó \.o	ö \~o	q \c o
ç \c c	q \d o	q \b o
Œ \OE	æ \ae	Æ \AE
ø \o	Ø \O	ı \l
ı \j	ı \j	ı ?'

Delimiters

' ' " "	{ \{	[[((
' ' " "	} \}	]]))

L^AT_EX

L^AT_EX

- ▶ Sivutyylejä: `plain`, `headings` ja `empty`
- ▶ Esim: `\pagestyle{empty}` jättää sivunumerot pois
- ▶ Rivien päättäminen: `\\` tai `\newline`
- ▶ Sivujen päättäminen: `\newpage` tai `\clearpage`
- ▶ Tavutus: Englanti hyvin, Suomi:
`\usepackage[finnish]{babel}`
- ▶ Sanassa toistuvia tavutus ongelmia ⇒ Dokumentin alussa komento. Esim:
`\hyphenation{ta-vu-tus-on-gel-ma}`
- ▶ Merkillä `\-` yksi ongelma dokumentin sisällä. Esim: `yh\dys\sa\na`
- ▶ Tavutusesto: `\mbox{sana...sanoja}`
- ▶ Tavutus epäonnistuu ⇒ Sana ylittää sivun marginaalin ⇒ **L^AT_EX** varoittaa **Overfull hbox** ⇒ Lisätään `\\` merkki

L^AT_EX

- ▶ Kappalejako: `\section{...}`,
`\subsection{...}`, `\subsub...`
- ▶ Sisällysluettelo:
`\tableofcontents{...}`
- ▶ Tekstisyöte
`\title{Elämää Ankkalinnassa}`
`\author{Leenu \and Liinu \and Tiinu}`
`\date{09.12.2014}`
`\maketitle`
tuottaa tässä luentojen **L^AT_EX** ympäristössä

Elämää Ankkalinnassa

Leenu Liinu Tiinu

L^AT_EX: verbatim

L^AT_EX: verbatim

- Komento `\verb|$$$%|` tuottaa `$$$%`, koska L^AT_EX jättää käsittelemättä | merkkien välissä olevan tekstin eli komentoja tai erikoismerkkejä ei huomioida
- Tekstisyöte


```
\begin{verbatim}
Mitä tahansa & _ ^ \ %
Mitä tahansa & _ ^ \ %
Mitä tahansa & _ ^ \ %
\end{verbatim}
```

 tuottaa


```
Mitä tahansa & _ ^ \ %
Mitä tahansa & _ ^ \ %
Mitä tahansa & _ ^ \ %
```
- Tätä ympäristöä voi käyttää esimerkiksi ohjelmien listaamiseen
- Käytetty näiden luentojen laatimiseen

L^AT_EX: fontit, kirjasintyypit ja koot

► Fontti esimerkkejä

`\rmfamily What` tuottaa **What**

`\sffamily What` tuottaa **What**

`\ttfamily What` tuottaa **What**

► Kirjaisin esimerkkejä

`\textbf Lihavointi` ⇒ **Lihavointi**

`\texttt Kirj...kone` ⇒ **Kirj...kone**

`\textit Kursiivi` ⇒ *Kursiivi*

`\underline Abc` ⇒ Abc

► Kirjaisin koko

`\tiny pientä tekstiä` ⇒ pientä tekstiä

`\Huge XYZ` ⇒ **XYZ**

Muita näiden väliltä `\scriptsize`,
`\small`, `\normalsize`, ... `\large`,
`\Large` ...

Tietokannat: bibtex

Tietokannat: bibtex

- **L^AT_EX** tunnistaa edellisellä luennolla tietokannasta haetusta tiedostosta **lahteet.bib** lähteen 1. rivin merkeistä **2014A&A...561A.141C** (engl. **label**)

```
@ARTICLE{2014A&A...561A.141C,
  author = {{Cubarsi}, R.},
  title = "{Conditions of consistency for multicomponent axisymmetric stellar systems . Is an axisymmetr",
  journal = {aap},
  keywords = {galaxies: kinematics and dynamics, solar neighborhood, galaxies: statistics},
  year = 2014,
  month = jan,
  volume = 561,
  eid = {A141},
  pages = {A141},
  doi = {10.1051/0004-6361/201322436},
  adsurl = {http://adsabs.harvard.edu/abs/2014A%26A...561A.141C},
  adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}
```

- Kopioidaan **Hlbkesken.tex** kotisivulta ja editoidaan neljä “uutta riviä”

```
\documentclass{article}
\newcommand{\aap}{A&A}           % uusi rivi
\begin{document}
I cite \cite{2014A&A...561A.141C} % uusi rivi
\bibliographystyle{plain}        % uusi rivi
\bibliography{lahteet}           % uusi rivi
\end{document}
```

- Komennnot: **latex Hlbkesken** , **bibtex Hlbkesken** , **latex Hlbkesken** , **latex Hlbkesken** , **dvips Hlbkesken -o**

Tietokannat: bibtex

Tietokannat: bibtex (kuva:@itunes.apple.com)

- ▶ Katsotaan lopputulosta komennolla
`evince Hlbkesken.ps &`
- ▶ Onnistuttiin siteeraamaan Cubarsia
- ▶ Luotiin lähdeluettelo
- ▶ Uusia lähteitä voi kerätä tietokannoista tiedostoon `lahteet.bib` ja siteerata `label:`n avulla
- ▶ Kaikki `Bibtex entry`:t täytyy kopioida ensimmäisestä viimeiseen merkkiin, eikä saa lisätä tai poistaa mitään `Bibtex entry`:ien väliltä
- ▶ Hyviä esimerkkejä `\bibliography` komennon käytöstä [täältä](#) | [www](#) |
- ▶ Loputtomasti [vaihtoehtoja](#) | [www](#) |
- ▶ **Kokeile:** korvaa `[plain]` tekstillä `[alpha]` ja toista edellisen sivun viisi komentoa
- ▶ Esimerkiksi ApJ (The Astrophysical Journal) formaatissa: `\citet` = Lähde ilman sulkuja = esim. Einstein (1906), `\citep` = Lähde suluissa = esim. (Einstein, 1906)



Tietokannat: Bibtex

Tietokannat: Bibtex

- ▶ **L^AT_EX** ei tunnistanut komentoa `\aap` tiedostossa `lahteet.bib` ⇒ Komento määritettiin tiedoston `Hlbkesken.tex` alussa rivillä
`\newcommand{\aap}{A\&A},`
 missä `A\&A` on lehdestä `Astronomy & Astrophysics` käytetty lyhennys
- ▶ `\documentstyle{article}` ja/tai `\bibliographystyle{plain}` ei tunnista `\aap` komentoa
- ▶ Useimpien lehtien makrot olisivat tunnistaneet `\aap` komennon ja kirjoittaneet lehden `Astronomy & Astrophysics` lyhenteen lähdeviitteisiin siinä muodossa, jota kyseinen julkaisusarja käyttää
- ▶ Yksi listaus referoiduista julkaisusarjoista käytetyistä lyhenteistä löytyy [täältä](#) | [www](#) |

Tietokannat: Bibtex

- ▶ Komentoyhdistelmä
`latex tiedosto`
`bibtex tiedosto`
`latex tiedosto`
`latex tiedosto`
`dvips tiedosto -o`
 voi "mennä solmuun"
- ▶ Jos `alias \rm=' \rm -i'` on voimassa, ongelma ratkeaa komennolla
 ⇒ `rm tiedosto.*` ⇒ **Esim:** tuloste

```
rm: remove regular file `tiedosto.aux'? y
rm: remove regular file `tiedosto.bbl'? y
rm: remove regular file `tiedosto.blg'? y
rm: remove regular file `tiedosto.dvi'? y
rm: remove regular file `tiedosto.log'? y
rm: remove regular file `tiedosto.ps'? y
rm: remove regular file `tiedosto.tex'? n
rm: remove regular file `tiedosto.tex'? n
```

 Vastataan `n` vain kahteen
- ▶ Tulostuvat sivut voi rajata
Esim: `dvips -o -p3 -15 tiedosto`
 ⇒ `tiedosto.ps`:ään vain sivut 3–5

python ja octave

python

- ▶ Muuttuja: "Olio koneen muistissa"
- ▶ Kokeillaan **python** Return
(jätetään `>>>` kirjoittamatta)
- ▶ Yksinkertaisin tapa luoda muuttuja on sijoitus: **x=1** Return
Muuttujan **nimi=tunnus** on **x**
Muuttujan **arvo** on **1**
Tyyppi on **kokonaislukumuuttuja**
- ▶ Toinen tapa on luoda **tyypin**
avulla: **x = int()** Return
luo tyhjän **kokonaislukumuuttujan**
- ▶ **print(x)** Return tulostaa **arvon 0**
- ▶ Merkillä **;** riville monta komentoa
a=1;b=type(a);print(b) Return
antaa **arvon** ja **tyypin** `<class 'int'>`

octave

- ▶ Kokeillaan **octave** Return
(jätetään **octave:*** kirjoittamatta)
- ▶ **x=1** Return tulostaa **x = 1**
- ▶ **x=1;** Return ; estää tulostuksen
- ▶ **class(x)** tulostaa **ans = double**
- ▶ **x=[]** tulostaa **x = [] (0x0)**
- ▶ MatLab = Matrix Laboratory
a=1;b=class(a);disp(b); Return
tulostaa **double**
- ▶ Merkillä **;** voidaan kirjoittaa **python**:ssa ja **octave**: komentoja samalle riville. Se estää myös **octave**:n **ans** tulostuksen
- ▶ **Mielipide**: "Monta komentoa samalla rivillä on huonoa koodausta"
- ▶ **Vastaus**: "Luennoilla pitkäkin ohjelma saadaan näin esitettyä yhdellä sivulla"

python ja octave

python

- ▶ **python** jatkuu: jätetään myös `Return` kirjoittamatta
- ▶ `x=[1,2] ; print(x[0])` tulostaa 1 eli **python**:ssa indeksit alkavat nolasta
- ▶ `x=[1 2] ⇒ Virheilmoitus`
- ▶ `type(x)` tulostaa `<class 'list'>` eli kyseessä on `listamuuttujalue`
- ▶ `len(x)` tulostaa 2
- ▶ `x=[1,2] ; y=x+1 ⇒ Virheilmoitus`
- ▶ `import numpy ; x=numpy.arange(2)+1 print(x)` tulostaa `[1 2]`
Tälle muuttujalle yhteenlasku toimii
- ▶ `y=x+1 ; print(y)` tulostaa `[2 3]`
- ▶ `x.size` tulostaa 2
- ▶ **python**: Muuttujien indekseistä | www |

octave

- ▶ **octave** jatkuu ...
- ▶ `x=[1,2]` toimii
- ▶ `x=[1 2]` sama tulos
- ▶ `disp(x(0)) ⇒ Virheilmoitus`
- ▶ `disp(x(1))` tulostaa 1 eli **octave**:ssa indeksit alkavat yhdestä
- ▶ `class(x)` tulostaa `double`
- ▶ `size(x)` tulostaa 1 2 eli 1 rivi ja 2 saraketta
- ▶ `length(x)` tulostaa 2
- ▶ `x = [1 2] ; y=x+1` tulostaa 2 3
- ▶ **octave**: Muuttujien indekseistä | www |
- ▶ Käytännössä muuttujia, sekä niiden yhdistelmiä, on **python**:ssa ja **octave**:ssa lopputtomiin ⇒ Jokainen opettelee tarpeen mukaan

python ja octave

python

```
▶ a=1 ; b=1.0 ; c='ab'
   type(a) , type(b) , type(c)
```

tulostaa

```
<type 'int'> , <type 'float'> , <type 'str'>
```

- ▶ Muuttujatyypit:


```
a = kokonaisluku = int
b = reaaliluku = float
c = "stringi" = str
```
- ▶ `x=[1, [1, 2], "abc"]` toimii!
- ▶ `type(x[0])` tulostaa `<type 'int'>`
- ▶ `type(x[1])` tulostaa `<type 'list'>`
- ▶ `type(x[2])` tulostaa `<type 'str'>`
- ▶ **python**:ssa listat voivat sisältää lukuja, tekstiä, listoja, ...
- ▶ Alkeet ... [EnOIKoTa](#)

octave

- ▶ `a=1 ; b=1.0 ; c='ab'`

```
class(a) , class(b) , class(c)
tulostaa double, double, char
```
- ▶ Muuttujatyypit:


```
a = reaaliluku = double
b = reaaliluku = double
c = string = char
```
- ▶ `x=[1, [1, 2], "abc"]` ei toimi!
- ▶ **octave**:ssa voidaan **python**:n listaa vastaava rakenne luoda komennolla


```
x.a=1 ; x.b=[1, 2]; x.c='ab' ;
```
- ▶ `disp(x)` tulostaa

```
scalar structure containing fields:
  a =  1
  b =
      1   2
  c = ab
```

- ▶ Menee näpertelyksi ... [EnOIKoTa](#)