Willkommen zum Etherpad!

Mit Hilfe von Etherpad können Sie Texte zeitgleich in Gruppen bearbeiten; dabei erscheinen sämtliche Eingaben sofort bei allen Gruppenmitgliedern.
Auf diese Art und Weise lässt sich der umständliche manuelle Abgleich von Dokumenten vermeiden.

Vorhandene Pads können in diverse gängige Formate exportiert werden und somit nahtlos in OpenOffice, LibreOffice oder Microsoft Office weiter verarbeitet werden.

Wichtiger Hinweis: Bitte beachten Sie, dass Ihre Pads von jeder Person, die die URL zu Ihrem Pad kennt, bearbeitet und gelöscht werden können. Etherpads eignen sich somit nicht als langfristige Speichermethode. Inaktive Pads werden automatisch nach 180 Tagen unwiederbringlich gelöscht.
Weitere Informationen finden Sie auf der Startseite dieses Dienstes unter http://etherpad.gwdg.de

---

Welcome to Etherpad!

This pad text is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents!

Please notice that inactive pads will be deleted.

Hello world! Never used Etherpad before... you?

Wanna import something and give a shorter name to it?
> import numpy as np
then
> np.zeros(5)
instead of
> numpy.zeros(5)

reset_selective variable_name -> delete a specific variable
del(variable_name) ^^

Nice cheatsheet for numpy
https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf

Can I use function from anohter package on numpy dataset?
if the function supports numpy arrays. many functions require 'iterables', numpy arrays are iterables, but this use of them may be very slow.

matplotlib doc:
http://matplotlib.org/contents.html

scatterplot-test:
```
image = matplotlib.pyplot.scatter(x = data[0,:], y = data[1,:])
matplotlib.pyplot.show()
```

Red sticky note:
- image dataset for examples
- speed? I meant connection speed... the speed of the course is fine for me actually ^^
- sometimes trouble to catch the syntax of the language
- please don't explain something while I am still writing

Green sticky notes:
- good intro to start from scratch, +5
- notebooks on the cluster makes sure that everyone is on the same page
- preliminary data exploration
- first plotting, +1
- etherpad
- style of presenting, +5
- good speed, +2
- sticky notes
- matplotlib
- numpy
- learned a few tricks
- python is quite appealing
- interactive exercises
- questions possible
- cloud computing rocks

==========
Hashtag for twitter: #DLBC17
Standing for DeepLearningBootCamp 2017
==========

Lost in all the python language features? Here you can find it all as a single page summary: https://learnxinyminutes.com/docs/python3/

To coiterate over indices and values you could also use enumerate (see https://stackoverflow.com/questions/522563/accessing-the-index-in-python-for-loops)

https://stackoverflow.com/questions/931092/reverse-a-string-in-python good explanation

How to properly create a multi-dimensional list: https://stackoverflow.com/questions/6667201/how-to-define-two-dimensional-array-in-python
Why so complicated? Because the multiplication of a given list causes trouble... try this:

```
lst = [[1,2,3,4,5]]*5
print(lst)
lst[0][0] = 1000
print(lst)
```

Result is: [[1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5], [1000, 2, 3, 4, 5]]
Sucks, eh?

So what if you do this instead:

```
lst = [[1,2,3,4,5] for muh in range(5)]
print(lst)
```

```
    lst[0][0] = 1000
    print(lst)
```
Better, eh?

Now, since we want to start loving 'enumerate', look at this:
```
    lst = [ [row*15+col for col in range(15)] for row in range(2) ]
    for rownum, list in enumerate(lst):
        for colnum, elem in enumerate(list):
            lst[rownum][colnum] = elem%(1+((1+rownum*colnum)%7))   %ignore what it does,
since it is just some random crap
    print(lst)
```
Ahhhh.... sweeet!!!

also for pandas there is a nice cheat sheet
https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PandasPythonForDataScience.pdf

data.loc['Italy':'Poland', 'gdpPercap_1962':'gdpPercap_1967'].idxmax() to get the index of the maximum in a data frame

Feedback of the Python introduction
---
red sticky notes:
- a bit fast (typing while listening while understanding was hard to do)
- pandas not so clear
- food for breaks
- cheat sheet
- pretty dense material for the time given
- what about classes/constructors/methods

green sticky notes

- - everything before pandas was good
- - comparison to R dataframes was good
- - cool ways to work with .csv
- - nice to have everything together in one notebook
- - learned about indexing arrays
- - good speed, good to understand
- - quite intuitive after 5hours
- - good overview
- - learnt a lot, +2
- - second best course that I have attended after 11 years

There are also different types of regularization methods depending on the network typ your using like "Dropout" (from G. Hintons lab on Neural Networks) - just google it :-)