COMPUTING SUBJECT:	Machine Learning
TYPE:	WORK ASSIGNMENT
IDENTIFICATION:	Jupyter notebook introduction
COPYRIGHT:	Michael Claudius
DEGREE OF DIFFICULTY:	Easy
TIME CONSUMPTION:	< 30 min.
EXTENT:	< 10 lines
OBJECTIVE:	Learning Jupyter
COMMANDS:	

IDENTIFICATION: Jupyter Introduction

The Mission

Your mission is to get used to the functionality and basics of Spyder and Jupyter.

Remark

Jupyter is an efficient tool when it comes to visualizing data and showing the functionality of code. Unlike Spyder, Jupyter does not supply you with a complete developer environment but are good for showing code fast and easy.

The problem

Jupyter has a different approach when it comes to creating projects and files, an approach that needs to be understood in order to start working productively with your python code. Your assignment is to create a feasible structure with new folders, build a new project and save it in your folder. *Remark*

By creating a structure, it will be easier for you to keep track of notebook-files and datasets.

Assignment 1 Building folders for Jupyter projects

Open Jupyter notebook on your pc. A console window will appear, telling you that Jupyter is running – it runs through your browser, so your pc will either ask you to choose a browser, or it will open Jupyter in your default browser (Chrome).

Your main page should look like this.

💭 Jupyter	Quit Logout
Files Running Clusters	
Select items to perform actions on them.	Upload New -
	Name Last Modified File size
D 3D Objects	2 måneder siden
Anaconda3	5 måneder siden
AndroidStudioProjects	9 måneder siden
	et år siden
Contacts	2 måneder siden
Desktop	en time siden
Documents	2 måneder siden
Downloads	19 dage siden
Evernote	10 måneder siden
Favorites	2 måneder siden
Links	2 måneder siden
Music	2 måneder siden
	et år siden
OneDrive - ErhvervsAkademi Sjælland	8 dage siden
PicStream	2 år siden
Pictures	2 måneder siden
Saved Games	2 måneder siden
Searches	2 måneder siden

You can choose you OneDrive path (it's a little tedious later on) or just create a new folder right away. We are following the last mentioned approach.

In the right corner of your screen. Click "New" -> Folder

	Upload New 🗸 🧲
	Notebook:
•	Python 3
-	Other:
	Text File
	Folder
1	Terminal

Your folder will be created as *untitled*, find it in the list, mark it (tick the box) and go to your upper left corner and choose "Rename". Name you folder "Machine Learning".

Files Running	Rename directory	×
Rename Move 💼	Enter a new directory name:	
	Machine Learning	×
DATAMAT	Cancel	Rename

Choose the new folder *Machine Learning* and inside create two new folders:

Solutions; this is for your own projects *Aurelien*: this is for the book examples and the Github examples

Make sure that your folder structure resembles:

💭 jupyter	Quit Logout
Files Running Clusters	
Select items to perform actions on them.	Upload New - C
0 V Machine Learning	Name Last Modified File size
۵	seconds ago
C Aurelien Book	a day ago
Solutions	16 hours ago

Assignment 2: Creating a new notebook

Now that you have learned how to create a new folder for projects, it is time to create a Python 3 notebook-file, to be used for testing some Python code.

Navigate to your solutions folder. Once inside create a new python 3 notebook file

perform actions on them.	Upload New -	3
I OneDrive - ErhvervsAkademi Sjælland / Machine Learning / Solutions	Notebook:	e.
	Python 3	
The notebook list is empty.	Other:	
	Folder	Γ
	Terminal	

Jupyter will **not** ask you to type in a name for the notebook. It will just create it as an untitled file. Go to "File / Rename" and change the name to *JupyterTest*.

A Jupyter file is built up upon cells – cells consist of the code you put in. You could put a lot of code in one cell, but that breaks the idea behind Jupyter.

JUPYTEr Untitled Last Checkpoint: 12 minutter siden (autosaved)	Logout
File Edit View Insert Cell Kernel Widgets Help	Trusted 🖋 Kernel O
E + ∞ 2 I	
In []:	

NOTE: Jupyter uses the "ipynb" extension (Iron Python NoteBook) and does not run within Spyder by default. Further you can **NOT** open a python-file (.py) in Jupyter and use it with the cell functionality.

Assignment 3: Working with code in Jupyter

When experimenting with code in Jupyter, you can output your code instantly. And see the results of your code.

In this assignment you are to work with some simple Python codes (one-liners and simple functions).

First, make a simple one-line code. Maybe like the below example:

print("Hello " + <your name>)

To run the code, simply hit the run-button (green triangle) and hopefully you will see something like:

```
In [1]: ▶ print("Hello Michael")
Hello Michael
```

So far so good!

Say you want to make a "smart" program that can greet a person when a name is entered, you would define a function that takes an argument, in this case a name, i.e. a string value. In cell [2]:

Define a function (C# method):

```
In [6]: M def say_hello(name):
    print("Hello " + name)
```

Then call the function:

In [7]: ▶ say_hello("Great Jens")
Hello Great Jens

Finally save the file in the *Machine Learning*\Solutions folder. Choose: Files -> Save As



Note: File extension must be .ipynb ! Note: If you later want to save the file just Choose: File -> Save with Checkpoint

If you like to rename the file later Choose: Files -> Rename

JUPYTET Hello Last Checkpoint: a minute ago (unsaved changes)								
File	Edit	View	Insert	Cell	Kernel	Widgets	Help	
New	Notebook n	•	↑ ↓	N Run	■ C	Code	V	
Make Save Rena	e a Copy e as ame	lun cint	rint("	Hello")				
Reve	ert to Chec	kpoint 🕨						
Print Dow	Preview	•						
Trus	ted Notebo	ok						
Clos	e and Halt							

Now the backbone operations are know and you proceed to next assignment !