The deadline for this exercise sheet is **Monday, 09.04.2018, 8:00.**

Some words before we start...

Welcome to the first exercise! Please submit your solutions to your group's folder on Stud.IP by the deadline each week. So in case you have not joined a group yet, please do so now. Though we strongly recommend that every group member solves all exercises on their own, please upload only one solution per group. We prefer if you put all your files into a .zip archive and upload this to your folder but uploading them separately is also fine.
So without further ado - let's start!

1 Installing Python

First, of course, you need to install Python. We recommend that you use Miniconda which is a package management system and makes managing a Python installation easy.

- Download Miniconda from: conda.io/miniconda.html

- Install and add it to your path (refer to the documentation for that or, if you are having trouble, come to the walk in session)

- Then run the following commands in your terminal/command line:

  ```
  conda install pip numpy matplotlib scipy
  conda install pandas jupyter scikit-learn scikit-image
  ```

- You can of course also download Python from the official website and use pip

2 Hello you!

To take first steps, we will for now work in the *Python Shell*. To open it, open your terminal/command line, type in "python" and press return. It should give you an output similar to this:

```
Python 3.6.3 |Anaconda, Inc.  (default, Oct 15 2017, 03:27:45) [MSC
v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If this is similar to what you're seeing, then you're good to go. (Otherwise, please check again if Python was installed correctly).
Now type the following and hit return:

```
print("Hello World!")
```

This will give you the following output:

```
Hello World!
```

Now modify the statement to create the following outputs:

1. `Hi, my name is [your name]!`

2. `What's up?`

3. *Bonus:*
   `So what's "Python"?`

4. *Bonus:*
   `Hello`
   `World`

To solve the two bonus tasks, you might have to do some googling. It's good to get used to it right away because programmers have to google. A lot. Therefore, there will probably be more exercises where we let you ask the internet for help. When you're finished, please write all your statements that succesfully printed the given sentences into a file and save it using the name *hello.py*. Feel free to play around with more statements.
To exit the Python shell, use the following command:

```
exit()
```

Solution (hello.py):

```python
# from the original print("Hello World!") just change what's inside the
    brackets
print("Hi, my name is [your name]!")
print("What's up?")

# Bonus
print("So what's \"Python\"?") # use \ to cancel the meaning of "
print("Hello\nWorld") # \n will add a line break
```

3 Fun with Turtles

In the late 20th century it became very popular to teach programming with a turtle. Those turtles were little robots that could move around - check one out here [1] Did you notice that it drew on the sheet of paper? No? Watch again then.

---

[1] Check this for more info. The concept is still popular today, although turtles are not enough anymore (https://studio.code.org/s/frozen/stage/1/puzzle/17)

Since we can not afford to buy a turtle for everyone, Python allows us to draw with a virtual turtle. Open your Python interpreter again, as you did in the previous exercise.

Note for Windows users: Please use this command instead this time:
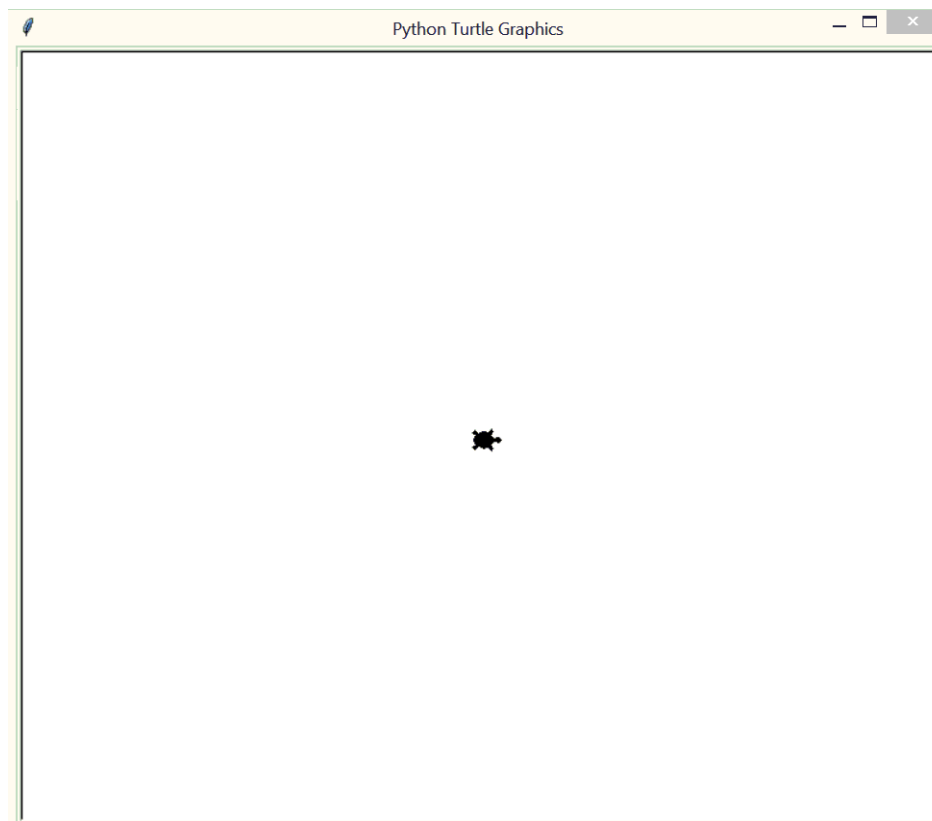
```
python -i
```

To use the turtle, you first have to tell Python that we need it:

```
import turtle
```

To show our turtle, use:

```
turtle.shape('turtle')
```

You should see a window like this:

Now you can let your turtle move around and draw with the following commands:

```
turtle.forward(100)
turtle.right(90)
```

```
turtle.backward(100)
turtle.left(90)
```

Another note for Windows users: You might notice the turtle window is sometimes unresponsive in the background. Don't worry - as soon as you're entering a new command, it will show up in the foreground again.

If you want to reset what your turtle drew, you can use:

```
turtle.reset()
```

In Germany little kids learn to draw the *Haus vom Nikolaus* (St. Nicholas' house) in one stroke. It looks like this:
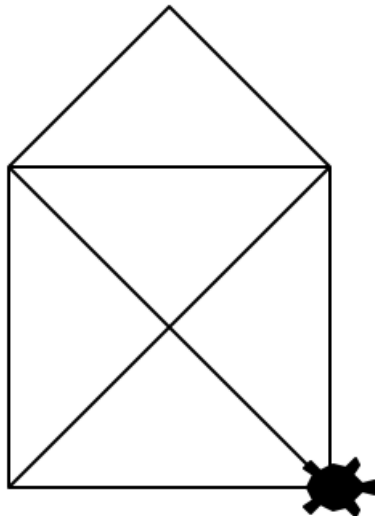
Can you give your turtle commands to draw something like this?
Store your successful commands in a file names *nicholas.py* for your submission.

Solution (nicholas.py):

```python
import turtle

turtle.shape('turtle')

# left edge
turtle.left(90)
turtle.forward(100)

# roof
```

```
turtle.right(45)
turtle.forward(70.71)
turtle.right(90)
turtle.forward(70.71)
turtle.right(135)
turtle.forward(100)

# cross and right edge
turtle.left(135)
turtle.forward(141.42)
turtle.right(45) # !
turtle.backward(100)
turtle.right(45)
turtle.forward(141.42)

# bottom edge
turtle.left(135)
turtle.forward(100)

# keep the window open (not needed for the homework)
turtle.done()
```