Day 2

...

More in depth with python!

But first!

Some repetition

Your previous files are here

>Link to repository

This time, we prepared usb-sticks you can use on the library machines!

https://github.com/cruor99/kivyteach

Python - Variables1

We can store values in a variable, where the values can then be changed and reused

Variables can be added to each other and create new variables by combining one or more other variables

variabler1.py

```
+    python = ‘Is python fun?’, 1 < 2
+    print python
+    a = 2
+    b = 3
+    print a + b
+    c = a + b
+    print ‘This course will run over’, c, ‘weeks’
```

Results:

```
$ python variabler1.py
Is python fun? True
5
This course will run over 5 weeks
```

Python - Variables2

We can add variables directly to strings

variabler2.py

```
navn = ‘Yngve’
alder = 23
hoyde = 174
vekt = ‘det vil jeg ikke si’
print ‘En av lærerne heter’, navn, ‘han er’, alder, ‘år og er’, hoyde, ‘høy. Men om dere spør hvor høy han er vil han si:’, vekt
```

Results:

```
$ kivy variabler2.py
en av lærerne heter Yngve han er 23 år og er 174 høy. men om dere spør hvor høy han er vil han si: det vil jeg ikke si
```

Python - Strings and variables

Mixing strings and variables is very simple in python.

Here, you use {} inside the string, followed by the .format() function

Examples:

```
+   a = “Kjetil”
+   b = “Yngve”
+   print “Vi heter {} og {}”.format(a, b)
```

```
-   print “Vi heter {} og {}”.format(a, b)
+   print “Vi heter {first} og {second}.format(first=b, second=a)
```

Python - Input

Some times, it's nice to have a little input!

In python, you do it like this:

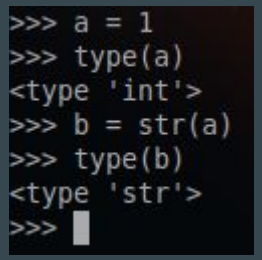
```
+    # -*- coding: utf-8 -*-
+
+    print “Hva heter du?
+    svar = raw_input()
+
+    print “Hyggelig å bli kjent, {}”.format(svar)
```

Python - converting values

In python, it is necessary to have things be of the same type for them to work together. This means that you can't do calculations with strings, and you can't concatenate numbers (integers etc) as strings.

The way this is handled in python is very simple:

```
+    a = 1
+    print type(a) # dette vil gi int som svar
+    b = str(a)
+    print type(b) # dette vil gi string
```



Assignment!

Let's do some repetition!

- Create a script that asks for three numbers, and calculates the average
- Create a script that holds a conversation with yourself!

Python - Methods

Think about methods as prepared actions, waiting to be used

Within methods, you put many instructions and you can repeat them as many times as you want.

```
+    # -*- coding: utf-8 -*-
+
+    def heiNavn(navn):
+        print “Hallo, {}”.format(navn)
+
+    a = raw_input()
+    heiNavn(a)
```

Python - Methods

One important concept in python is “block levels”
This is the level of indentation of a segment of python code, in relation to the rest of the code.
Standard python is that a block should be indented 4 spaces - and it is important to follow that.

```
+    def example():
+    1234newblock = “Dette er en ny blokk”
+    1234sameblock = “Dette er den samme blokken”
+    1234print newblock + sameblock
+
+    newblock = “Dette er en helt ny blokk”
+    print newblock
+    example()
```

Python - Conditionals

Conditionals are often used in programming

In python, as opposed to some other languages, we can and many times should use some english words as keywords, instead of the normal conditionals

```
+    verdi = 1
+    if verdi is 0:
+        print “verdien er null!”
+    else:
+        print “Verdien er: {}”.format(verdi)
```

Python - Loops

Loops are used when you want to repeat things.
There are both “while” loops - “Do this, while this is true” and “for” loops - “Do this for such and such”

```
+    for i in range(10):
+        print i
```

```
+    i = 0
+    while i < 20:
+        print i
+        i += 1
```

Assignment:

- Create a program that writes ten lines, where every other line is supposed to be your front name and then last name.
- Create a program where you are supposed to guess the right number between 1 and 20. You should only have 5 tries, and you should report if the guess is lower or higher

```
+    import random
+    #Use the randrange method to generate the number
+    print random.randrange(1,21)
```

Lists

```
# python
Python 3.5.0 (default, Sep 20 2015, 11:28:25)
[GCC 5.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> liste = [1, 2, 3, 4]
>>> liste.append(5)
>>> print(liste)
[1, 2, 3, 4, 5]
>>> liste.pop()
5
>>> print(liste)
[1, 2, 3, 4]
>>> liste[0]
1
>>> liste[4]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>> liste[3]
4
>>>
```

Loops and lists

You can easily parse through a list, like we've seen with loops previously!

```
+    listen = [“Kari”, “Kåre”, “Olekårejohnnyper”]
+    for navn i listen:
+        print navn
```

Husk encoding i toppen av filen!

```
+    # -*- coding: utf-8 -*-
     listen = [“Kari”, “Kåre”, “Olekårejohnnyper”]
     ...
```

Dicts!

```
$ python
Python 2.7.9 (default, Dec 19 2014, 06:05:48)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.56)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> ordbok = {}
>>> ordbok["Kjetil"] = "92882979"
>>> print ordbok["Kjetil"]
92882979
>>> print ordbok
{'Kjetil': '92882979'}
>>>
```

Assignment!

Create a list of five names

Give all the people a dice roll between 1 and 6

Print the person with the highest number

Store the results in a dict with the name as the key, and dice roll as value