Welcome to today's meeting

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1
/DTD/xhtml1-transitional.dtd">
<html>
```

Obroni Computer Club – Programming in Python

Joachim Breitner

SOS Hermann Gmeiner International College

September 5th 2006

Today's topics

1. Prelude
   - Review of the last meeting

2. Our programming project

3. Using the command line

4. My first Python program
   - The Hello World program

Prelude

1 **Prelude**
  - Review of the last meeting

2 Our programming project

3 Using the command line

4 My first Python program
  - The Hello World program

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation

Project

Command line

Python
Hello World

Do you still remember?

Last week, we had a look at how software is made, what Free Software is and why it is important.

Do you still remember?

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Last week, we had a look at how software is made, what Free Software is and why it is important.

We also started the "Geek Points" competition, with points awarded to things like posting blog entries, contributing to the wiki and finding out simple and not-so-simple things, including the Greeting Challenge, through research.

Do you still remember?

Last week, we had a look at how software is made, what Free Software is and why it is important.

We also started the "Geek Points" competition, with points awarded to things like posting blog entries, contributing to the wiki and finding out simple and not-so-simple things, including the Greeting Challenge, through research.

So far, nothing in that direction has happened.

Our programming project

What do we want to do?

When learning programming, it always helps to have a certain goal, that is a program that we would like to have in the end. Ideas?

What do we want to do?

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation

Project

Command line

Python
Hello World

When learning programming, it always helps to have a certain
goal, that is a program that we would like to have in the end.
Ideas?

My proposal, as you liked the `talk` program:

Programming Goal

We write a simple multi-user chat program for us here on the
local network, which we then can extend as we wish.

(But until we can do that we need to learn a few basics)

Now playing

A Python Lovestory

23 minutes

`http://www.ibiblio.org/obp/pyBiblio/pythonvideo.php`

Using the command line

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Logging in

As I was not yet cleared to put Linux on these machines, we will have to start using only the command line, both for editing and for running the programs.

To log in, use the program `putty` in `K:\Others\OCC\`, enter `occ` as the hostname. The security warning, that might pop up, can be dismissed with "Ok". Enter then your username and password. **It is normal that you can't see the password when you type it.**

You may want to login twice: Once for editing the files, once for testing your program.

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Mini-Command-Line-Introduction

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

A few interesting commands:

Command	What it does
mkdir *dir*	Creates a directory
cd *dir*	Goes to that directory
ls	Lists files in the current directory
cat *file*	Outputs the file
chmod +x *file*	Makes a file executable
./*file*	Runs a file
rm *file*	Deletes a file
nano *file*	Edits a file

The Editor

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

There are many text editors for the command line, some extremely powerful (most notably `vim` and `EMACS`). For now, we use a simle one: "nano".

You start it with "nano *file*", for example "nano test.py".

The bottom lists a few actions. `^X` for example means "press Control-X". Important ones are `^O` and `^X`.

Putting it together

The steps for programing are:

1. Create a directory for your programming stuff, go into it
2. Use the editor to write the source code
3. Save it to a file (you can leave it open)
4. Make that file executable
5. Run the program
6. Make more changes to the program
7. Save it again (important)
8. Run the program again
9. Repeat until satisfied
10. Serve with chocolate sauce

My first Python program

Let's look at this code:

```
1  #!/usr/bin/python
2
3  print 'Hello World!'
```

OCC
Programming

Joachim
Breitner

Prelude
 Recapitulation
Project
Command line
Python
 Hello World

Let's look at this code:

```
1  #!/usr/bin/python
2
3  print 'Hello World!'
```

The file consists of two elements:

1. The first line tell the operating sytem, in this case Linux, what program (/usr/bin/python) is to be used to run this file. Python itself ignores this file, as it starts with "#"

2. The last line is a *statement*, telling to do something. Printing means "printing to the screen", to to paper. We have to put quotes around "Hello World!" to tell Python that this is actually a text, and not some program code. (compare **print print** to **print** 'print'!)

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Exercise: more print statements

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Now, try it yourself! Also try to create a program that has the
following output:

```
Hi you

how are you?
```

Exercise: more print statements

OCC
Programming

Joachim
Breitner

Prelude
Recapitulation
Project
Command line
Python
Hello World

Now, try it yourself! Also try to create a program that has the following output:

```
Hi you

how are you?
```

Solution:

```
1  #!/usr/bin/python
2
3  print 'Hi you'
4  print
5  print 'How are you?'
```

Any Questions?

Good bye 'till next time

```
</html>
```