

Inhaltsverzeichnis

0.1	Files	2
0.2	Mainpage	2
0.3	Classes	3
0.4	Functions	3
0.5	Variables	4

Doxygen Guide

This is a small doxygen guide of how to use doxygen correctly in your code.
If you have some issues, ideas or suggestions: Please contact the author of this document!

Based on the "Doxygen Guide".¹

0.1 Files

The first line of letters should be the file (source and header files) description in this form:

```
/**
 * @file      filename.ext
 * @brief     a short description of the content
 * @author    name
 * @date      date YYYY-MM-DD
 */
```

The name of the author should be in the form of the first letter of the first name and all letters of the last name. For example the name "Klaus Mustermann". The short name is "kmustermann". The description should be completely written in lower case!

0.2 Mainpage

To define a source file as mainpage (shown at the first side of the doxygen documentation) you have to involve the Mainpage-Tag in the file, like this:

```
/**
 * @file      filename.ext
 * @brief     a short description of the content
 * @author    name
 * @date      date YYYY-MM-DD
 *
 * \mainpage This is my main page
 *
 * And this is my so necessary description
 */
```

¹<http://www.stack.nl/~dimitri/doxygen/>

```
    of my project.  
**/
```

0.3 Classes

The class description should be written in front of the class declaration:

```
/**  
    @class      classname  
    @brief      short description of class  
**/  
class classname  
{  
    //code to be executed  
};
```

0.4 Functions

The function description should be in front of the function implementation:

```
/**  
    @fn      classname :: functionname  
    @brief      short description of function  
**/  
classname::functionname()  
{  
    //code to be executed  
}
```

For destructor description you have to put the ' ' -symbol in front of the function name.

If you have some in- and out-parameters, you have to describe that parameter also in the first description comment block:

```
/**  
    @fn      functionname  
    @brief      short description of function  
    @param      inparaname      short description  
    @return     outparaname      short description  
**/  
outparaname functionname (float inparaname)  
{  
    //code to be executed
```

```
}
```

0.5 Variables

To add a description to variables, you have to choose this way:

```
class classname : public motherclass
{
    protected:
    private:
        // Variables
        std::string mMyVar;  //!< short variable description

        //more code goes here
}
```