Einstieg in Python (GFU Cyrus Köln)

Release 0.02

Altan Tosun

August 15, 2011

CONTENTS

1 Ein paar Sphinx-Tipps **3**
- 1.1 Sphinix-Installtion . . . 3
- 1.2 Sphinx-Arbeitsverzeichnis einrichten . . . 3
- 1.3 Python-Code einfügen . . . 3
- 1.4 Links einfügen . . . 3
- 1.5 Versionierung mit Mercurial . . . 4
- 1.6 Zeigen von Pythoh-Code . . . 4

2 Python Packages und Module **5**
- 2.1 Allgemeines . . . 5
- 2.2 Elemente des Moduls benutzen . . . 5
- 2.3 Eine Funktion in einem Modul implementieren . . . 7

3 Indices and tables **9**

Python Module Index **11**

Index **13**

Contents:

CHAPTER

ONE

EIN PAAR SPHINX-TIPPS

1.1 Sphinix-Installtion

Wie installiere ich Sphinx in Python?

1.2 Sphinx-Arbeitsverzeichnis einrichten

1.2.1 Benutzung von sphinx-quickstart

Wie erstelle ich ein Sphinx-Arbeitsverzeichnis?

```
$ sphinx-quickstart
```

1.2.2 Antworten auf die Fragen von Sphinx.

1.3 Python-Code einfügen

So zum Beispiel kann man in Sphinx Python-Code einfügen

```
>>> 2+5
7
>>> def f():
        return 77
...
>>> f()
77
>>>
```

1.4 Links einfügen

Python ist *Eigentum* der Python Foundation, eine US-Amerikanische **gemeinnützige** Stiftung.

1.5 Versionierung mit Mercurial

Wichtig ist nur 'source' zu versionieren. Siehe `.hgignore` für den Ausschluß des Ordners 'build'.

1.6 Zeigen von Pythoh-Code

Python-Code aus dem Python-Package `examples` in die Sphinx-basierte Dokumentation zu zeigen. Some simple Tk based utilities.

`examples.tk_simple.`**`tk_message`**(*title='message window'*, *geometry='200x80+200+200'*, *message='Hello from TK:'*)

Displays a message in a Tk window.

`examples.tk_simple.`**`tk_readstring`**(*title='input window'*, *geometry='200x80+200+200'*, *prompt='input:'*)

A simple Tk window for reading a string.

Returns the string from the string input field.

Finish with 'RETURN' or click on 'Ok'.

CHAPTER

TWO

PYTHON PACKAGES UND MODULE

2.1 Allgemeines

Ein Python-Modul ist kurz gesagt eine Datei mit der Endung '.py', die Python-Code enthält.

Ein Python-Package ist ein Ordner mit Python-Modulen, mindestens mit dem Modul '__init__.py'.

So ist der Ordner 'examples' ein Python-Package, weil er das Modul '__init__.py' enthält.

Ein Modul enthält:

1. Funktionen
2. Klassen
3. Modulweit bekannte Variablen

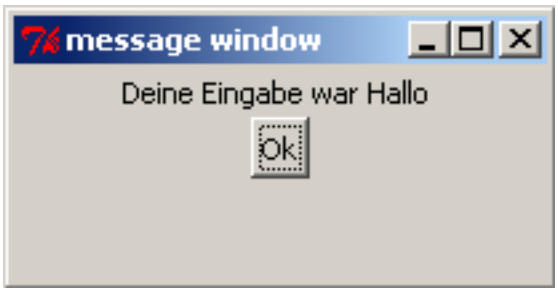
Beispiel: das Modul 'examples.tk_simple'

2.2 Elemente des Moduls benutzen

2.2.1 Aufruf mit Default-Parameter

Importieren einer Funktion aus einem Modul und die Funktion ohne Parameter aufrufen.

```
>>> from tk_simple import tk_message
>>> tk_message () # aufruf der Funktion
>>>
```



Es greifen dann die Vorgabe-Parameter (default parameters).

Die Vorgabeparameter von tk_message sind:

title message window

geometry 200x80+200+200

message Hello from TK

Dies kann aus der Deklaration der Funktion entnommen werden:

`examples.tk_simple.`**`tk_message`**(*title='message window', geometry='200x80+200+200', message='Hello from TK:'*)

Displays a message in a Tk window.

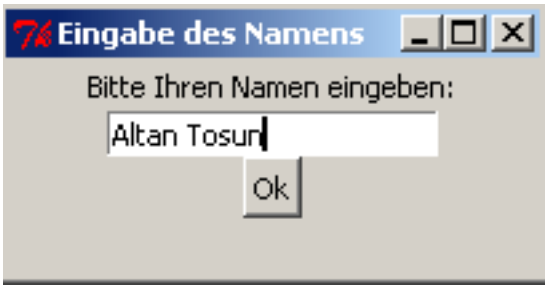
2.2.2 Aufruf mit nicht-Default-Parametern

Aufruf der Funktion mit veränderten Parametern:

```
>>> tk_message(title='Nachricht aus Köln', message='Kölle Alaaf!')
```

Aufruf der Funktion tk_readstring (wobei das Ergebnis in der Variablen 'name' von Python abgefangen wird):

```
>>> name = tk_readstring(title='Eingabe des Namens', prompt="Bitte Ihren Namen eingeben:")
```



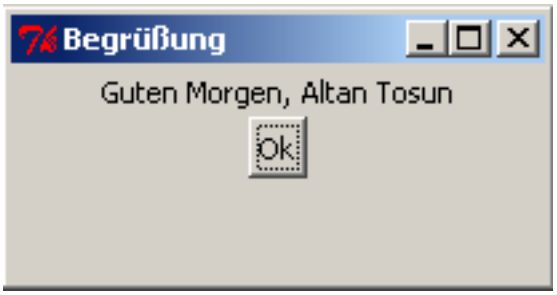

```
>>> name
'Altan Tosun'
>>>
```

Der Inhalt der Variablen 'name' kann dazu benutzt werden, eine neue Variable 'greetings' zu definieren mit einem ausgearbeiteten Inhalt.

Der Inhalt der neuen Variaablen 'greetings' kann wiederum dazu benutzt werden, um ein Message-Fenster auszugeben.

```
>>> greetings = "Guten Morgen, " + name
>>> greetings
'Guten Morgen, Altan Tosun'
>>> tk_message(title='Begrüßung', message=greetings)

>>> tk_message(title='Begrüßung', message=greetings)
>>>
```



2.3 Eine Funktion in einem Modul implementieren

Übungsaufgabe:

Ein neues Modul tk_dialogs anfangen Eine Funktion tk_dialogs implementieren, die folgendes macht:

- den Namen einer Person abfragt
- eine Begrüßung ausgehend vom Namen zusammenstellt
- die zusammengestellte Begrüßung ausgibt

Benutzung der Funktion tk_message und tk_readstring vom Modul tk_simple

Lösung:

`examples.tk_dialogs.`**`tk_greeting`**`()`
Fragt den Namen und dann begrüßt

Tip: Testen Sie die neue Funktion in einem interakziven Python-Interpreter etwa so:

Importieren Sie nichtt aus dem Modul, sondern das Modul:

```
>>> import tk_dialogs
```

Rufen Sie die testende Funktion vom Modul aus durch den Punkt-Operator.

```
>>> tk_dialogs.tk_greeting()
```

Wenn Sie etwas in der Funktion oder sonst im Modul korrigieren, laden Sie das Modul neu:

```
>>> reload(tk_dialogs)
<... insert Meldung aus der SHELL ...>
```

Testen Sie die korrigierte Version:

>> tk_dialogs.tk_greeting()

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

e

examples.tk_simple, 4

INDEX

E

examples.tk_simple (module), 4

T

tk_greeting() (in module examples.tk_dialogs), 7
tk_message() (in module examples.tk_simple), 4
tk_readstring() (in module examples.tk_simple), 4