



# Hochschule Augsburg University of Applied Sciences

## Diplomarbeit Studienrichtung Informatik

**Jorge Andrés Cuartas Monroy**  
Vergleich von Pythonframeworks  
und Implementierung einer  
Webschnittstelle für Mikrocontroller

Verfasser der Diplomarbeit:  
Jorge Andrés Cuartas Monroy  
Rehmstr. 7  
866161 Augsburg  
Telefon: 0176 21232896  
jorge.cuartas@googlemail.com

Fakultät für Informatik  
Telefon: +49 821 5586-3450  
Fax: +49 821 5586-3499

Hochschule Augsburg  
University of Applied Sciences  
Baumgartnerstraße 16  
D 86161 Augsburg

Telefon +49 821 5586-0  
Fax +49 821 5586-3222  
<http://www.hs-augsburg.de>  
poststelle@hs-augsburg.de

Erstprüfer: Prof. Dr. Hubert Högl  
Zweitprüfer: .....  
Abgabe der Arbeit: .....

## Sie dürfen:

- Das Werk bzw. den Inhalt vervielfältigen, verbreiten und öffentlich zugänglich machen
- Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen

## Zu den folgenden Bedingungen:

- Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Keine kommerzielle Nutzung — Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

## Wobei gilt:

- Verzichtserklärung — Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.
- Sonstige Rechte — Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:
  - Die gesetzlichen Schranken des Urheberrechts und sonstigen Befugnisse zur privaten Nutzung
  - Das Urheberpersönlichkeitsrecht des Rechteinhabers
  - Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel Persönlichkeitsrechte abgebildeter Personen.
- Hinweis — Im Falle einer Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für dieses Werk gelten. Am einfachsten ist es, an entsprechender Stelle einen Link auf diese Seite einzubinden.

## Hinweis

Sie können den Lizenztext unter <http://creativecommons.org/licenses/by-nc-sa/3.0/deed.de> nachlesen.

# Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Augsburg, den Tag. Dezember Jahr

---

Jorge Andres Cuartas Monroy

# Inhaltsverzeichnis

<b>1</b>	<b>Anwendung</b>	<b>6</b>
1.1	Überblick . . . . .	6
<b>2</b>	<b>Webframeworks</b>	<b>8</b>
2.1	Definition . . . . .	8
2.2	Warum Python . . . . .	8
2.3	Fullstack vs. Glue . . . . .	9
2.4	Weitere Unterscheidungsmerkmale . . . . .	10
2.4.1	Funktionalität . . . . .	10
2.4.2	Dokumentation . . . . .	10
2.5	Bekannte Webframeworks im Pythonumfeld . . . . .	11
2.5.1	Django . . . . .	11
2.5.2	Pylons . . . . .	11
2.5.3	Zope . . . . .	11
2.5.4	web2py . . . . .	11
2.6	Webframeworks anderer Programmiersprachen . . . . .	12
2.6.1	Überblick . . . . .	12
2.7	Arbeiten ohne Framework . . . . .	12
<b>3</b>	<b>Django</b>	<b>13</b>
3.1	Einführung . . . . .	13
3.2	Installation . . . . .	13
3.3	Dokumentation . . . . .	13
3.4	Datenbanksysteme . . . . .	13
3.5	Templatesystem . . . . .	13
3.6	JavaScript . . . . .	13
3.7	Sicherheitsaspekte . . . . .	13
3.8	Durchführung der Anwendung . . . . .	13

<b>4</b>	<b>TurboGears</b>	<b>14</b>
4.1	.....	14
<b>5</b>	<b>Pylons</b>	<b>15</b>
5.1	Einführung .....	15
<b>6</b>	<b>Zope</b>	<b>16</b>
6.1	Einführung .....	16
<b>7</b>	<b>Web2Py</b>	<b>17</b>
7.1	Einführung .....	17
<b>8</b>	<b>Ergebnisse / Übersicht</b>	<b>18</b>
8.1	Bewertungskriterien .....	18

# 1 Anwendung

## 1.1 Überblick

Im Zuge der Diplomarbeit soll neben einem Vergleich zu Pythonwebframeworks auch eine Anwendung für des Rechnertechniklabor der HS Augsburg geschrieben werden. Dabei soll der Zugriff auf Mikrocontroller ermöglicht werden, in Form einer Webanwendung. Die Anwendung soll den Benutzern den Zugang auf die Mikrocontroller ermöglichen. Dabei sollen sie die einzelnen Boards reservieren können und aus Performance Gründen nur jeweils einem einzelnen Benutzer gleichzeitig den Zugang auf ein Board ermöglicht werden. Darüber hinaus wird folgende Funktionalität von der Anwendung aus zugänglich:

- Power on/off
- Reset
- Statusinformationen abrufen
- Reservierung
- Webupload der Root-Filesysteme

Weiterhin bekommt der Benutzer die Möglichkeit Statusinformationen abzurufen, die auf der Weboberfläche angezeigt werden. Darüber hinaus auch ein Webupload von Rootfilesystemen möglich, die auf dem Board mit Hilfe von Skripten auf die Boards automatisch aufgespielt werden. Zu den Arbeiten gehört weiterhin die Installation entsprechender Frameworks und nötige Software für den Reibungslosen Ablauf der Applikation. Die Anwendung wird gegebenenfalls unter verschiedenen Technologien implementiert. Nach der Reservierung der Boards bekommt der Benutzer einen Zugang mit Passwort, damit dieser über SSH auf das Board zugreifen kann. Das Passwort wird nach Reservierung jeweils neu erstellt und nach dem Abmelden aus der Userverwaltung oder nachdem

der User nicht mehr auf dem Board arbeiten will der Zugang gesperrt, bis ein anderer das Board reserviert.

Für jedes der hier vorgestellten Webframeworks wird diese Anwendung entwickelt. Es soll nach dem Durchlesen dieser Arbeit möglich sein, ein Vergleich zu ziehen, welches Framework die beste Unterstützung bietet, um Webprojekte im Pythomumfeld zu erstellen. Normalerweise denkt man heutzutage bei Webentwicklung hauptsächlich an *PHP* als Hauptsprache, für die es unzählige Frameworks und Anwendungen<sup>1</sup> gibt. Auch wird oft *Ruby on Rails* im Zusammenhang gebracht, bei der es eine große Entwicklergemeinde existiert und ebensoviele bekannte Projekte<sup>2</sup>.

So soll aufgezeigt werden, dass auch der Pythonentwickler seine hart erarbeiteten Kenntnisse auch im Webumfeld einsetzen kann und keine neue Programmiersprache dafür lernen muss. Mit Django, Zope, Pylons, web2py gibt es genug Alternativen für die Entwicklungen von Anwendungen in der Python-Welt. Diese werden anhand einer Beispielanwendung vorgestellt und untersucht.

---

<sup>1</sup>[magentocommerce.com](http://magentocommerce.com), [atmail.com](http://atmail.com)

<sup>2</sup>unter anderen [twitter.com](http://twitter.com), [shopify.com](http://shopify.com), [github.com](http://github.com)

## 2 Webframeworks

### 2.1 Definition

Webframeworks sind im Grunde eine Sammlung von Funktionalität, die es dem Webentwickler erlaubt Anwendungen zu schreiben. Die Funktionalität besteht aus verschiedenen Libraries, die den Zugriff auf Datenbanken, die Erstellung von Templates für die Ausgabe in HTML oder die Erstellung von Klassen und Methoden für die Geschäftslogik.

Webanwendungen, die sich auf Datenbanken stützen können mit Webframeworks relativ schnell erstellt werden, dabei helfen OR-Mapper<sup>1</sup>, die aus Klassenkonstrukte SQL<sup>2</sup> für eine ausgewählte Datenbank erstellen. Die Mapper bieten darüberhinaus eine eigene Sprache, um auf die Klassen bzw. Informationen in der Datenbank zuzugreifen. Der Vorteil liegt darin, dass der Code unabhängig von einer Datenbank erstellt werden kann. Die Geschäftslogik kann abhängig vom Framework in einer beliebigen Programmiersprache geschrieben werden, meistens Java, PHP oder Python. Um die Ausgabe zu generieren bringen die Frameworks eine Templatesprache mit. Diese ermöglicht die einfache Erstellung von HTML aufgrund der Mischung von HTML und Templatecode. Dieser Code dient überwiegend zur Darstellung von Information, die aus der Geschäftslogik stammt.

### 2.2 Warum Python

Die Programmiersprache Python ist eine sehr flexible Programmiersprache. Sie kann in verschiedenen Gebieten eingesetzt werden. Unter anderem bei der Verarbeitung von XML, zur Verbindung von Datenbanken, in der Netzwerk Programmierung, im Webbereich auch die Erstellung von Programmen mit grafischer Oberfläche (wxPython, PyQt) ist möglich. In dieser Arbeit wird vor allem der Webbereich genauer analysiert und verschiedene Webframeworks am Beispiel einer Anwendung verglichen.

Python ist nicht an ein bestimmtes Programmierparadigma gebunden und je nach

---

<sup>1</sup>Objekt Relationale Mapper

<sup>2</sup>Structured Query Language



Aufgabe kann ein Paradigma, wie z.B. Objektorientierung, Funktionaler Programmierung oder ein aspektorientierter Ansatz gewählt werden.

Im Jahre 1991 wurde die Sprache veröffentlicht. Sie wurde im *Centrum voor Wiskunde en Informatica* in Amsterdam entwickelt. Dabei gilt *Guido van Rossum* als Erfinder der Sprache. Python wurde unter anderem von C, LISP und Smalltalk beeinflusst. Wobei Python selbst wiederum Ruby, Boo oder Groovy beeinflusst hat.

Ja, Python ist gewachsen und hat sich zu einer ausgereiften Sprache samt eigenem Ökosystem entwickelt.[1]

Im Vergleich zu z.B. PHP[3] gibt es zwar auch viele Python-Webframeworks[4], diese sind aber weniger bekannt, mit der Ausnahme von z.B. Django oder TurboGears. Es werden diese zwei beim Vergleich eingesetzt, zusätzlich auch neuere oder weniger bekannte Frameworks im Pythonökosystem.

## 2.3 Fullstack vs. Glue

Frameworks werden hauptsächlich unterschieden zwischen Fullstack- und Glueframeworks.

Ein Fullstackframework bringt alle möglichen Komponenten mit, um eine Webanwendung schreiben zu können. Unter anderem sind das ein OR-Mapper für die Datenbank, eine eigene Templatesprache oder ein URL-Dispatcher<sup>3</sup> für lesbare Links. Der Vorteil eines solchen Frameworks ist, dass eine klare Linie vorgegeben wird, um eine Anwendung schreiben zu können. Dies führt in der Regel zu schnellen Ergebnissen. Vertreter dieser Art sind z.B. Django.

Auf der anderen Seite gibt es Frameworks, die aus einzelnen unabhängigen Komponenten bestehen. Dabei sind diese oft lose gekoppelt und können leicht ausgetauscht werden. So können für diese Frameworks immer die neuesten Komponenten benutzt werden, die von Gluecode<sup>4</sup> zusammengehalten werden. Es ist aber ein größerer Aufwand nötig, um diese losen Teile zusammenzubringen. Das bekannteste Framework dieser Art in Python ist TurboGears.

---

<sup>3</sup><http://www.w3.org/Provider/Style/URI>

<sup>4</sup>[http://en.wikipedia.org/wiki/Glue\\_code](http://en.wikipedia.org/wiki/Glue_code)

## 2.4 Weitere Unterscheidungsmerkmale

### 2.4.1 Funktionalität

Wenn ein neues Produkt das Interesse geweckt hat, liegt es meistens an der Funktionalität, die es anbietet. Für eine Webapplikation ist es wichtig, dass auf eine Art und Weise der Client auf die Dateien mittels Browser zugreifen kann. Als Schnittstelle bietet sich im Pythonumfeld WSGI<sup>5</sup> an, damit der Webserver mit der Webapplikation kommunizieren kann, die wiederum HTML an den Client liefert. Für einfache Anwendungen reicht es aus HTML-Code mit Hilfe von Python zu generieren und diesem dem User zur Verfügung zu stellen. Soll die Anwendung aber komplexere Ausmaße haben und zum Beispiel eine Userverwaltung mit LogIn Möglichkeit bieten ist es ratsam ein Framework auszuwählen, welches diese Funktionalität mit einfachen Handgriffen bietet. Auch die Kommunikation zu Datenbanken ist ein wichtiger Bestandteil, da viele Webapplikationen Klassen bereitstellen, mittels derer Objekte generiert werden, die miteinander interagieren. Daraus entstehen meistens Informationen die persistent gehalten werden müssen. Beispiele sind Webshops, Blogs/Wikis oder soziale Netzwerke. Bei der Entwicklung sollte auch berücksichtigt werden, dass Webdesigner leicht im Entwicklungsprozess integriert werden können. Dabei sollte wichtig sein, dass CSS und JavaScript (oder JavaScript-Frameworks), die als Standard bei der Gestaltung gelten eingesetzt werden können. Dabei spielen Templatesprachen für die Erstellung des HTML-Codes eine große Rolle, denn viele Webframeworks bieten Templatesprachen, die sich an Pythoncode anlehnen aber nur geringe logische Funktionalität bieten und im Grunde nur für die Darstellung der einzelnen Seiten konzipiert sind.

### 2.4.2 Dokumentation

Viele Projekte und im Speziellen hier Webframeworks werden hauptsächlich wegen ihrer Funktionalität ausgewählt. Ein weiterer Grund kann auch ihre Popularität sein auf Grund eines Hypes. Ein sehr wichtiger Aspekt, um überhaupt mit einem Framework oder Programmiersprache anzufangen, ist die Lernhürde die manchmal hoch sein kann. Deswegen ist es wichtig und sollte eine Priorität der Entwickler sein die Frameworks sehr gut zu dokumentieren und Anfängertutorials zu anbieten.

---

<sup>5</sup><http://wsgi.org/wsgi/>

## 2.5 Bekannte Webframeworks im Pythonumfeld

### 2.5.1 Django

Django ist ein High-Level Python Webframework, das eine schnelle Entwicklung fördert und ein sauberes und pragmatisches Design verfolgt.

Django wurde designed, um zwei Herausforderungen zu meistern: Intensiver Deadlines eines Newsrooms und den strengen Anforderungen der erfahrenden Webentwickler, die das Framework geschrieben haben, gerecht zu werden. Django ermöglicht es performante und elegante Anwendungen, schnell zu entwickeln. [6]

### 2.5.2 Pylons

Pylons verbindet Ideen aus der Welt von Ruby, Python und Perl. Dabei ist es ein sehr flexibles Python-Webframework und verhalf dem WSGI-Standard zum Durchbruch. Das Ziel von Pylons ist es, die Webentwicklung schnell, flexibel und einfach zu gestalten.[5]

### 2.5.3 Zope

Die Zope Softwarebibliothek ermöglicht die komponentenbasierte Entwicklung von Web-Anwendungen in der objektorientierten Programmiersprache Python. In der komponentenbasierten Programmierung werden komplexe Anwendungen mit Hilfe wiederverwendbarer Komponenten erstellt. Eine Komponente stellt dabei die Implementierung einer bestimmten genau spezifizierten Funktionalität dar.[7]

### 2.5.4 web2py

web2py wurde inspiriert von Ruby on Rails und, wie Rails, liegt der Fokus in schneller Webentwicklung und folgt dem Model View Controller Muster. web2py unterscheidet sich von Rails, weil es auf Python basiert. Weil es eine umfassende Web-basierte Administrationsoberfläche bereitstellt, darüber hinaus beinhaltet es Bibliotheken, um mit mehr Protokollen zurechtzukommen und ist auf der Google App Engine lauffähig.[8]

## 2.6 Webframeworks anderer Programmiersprachen

### 2.6.1 Überblick

Zur Erstellung von Webapplikationen gibt es viele Möglichkeiten dies zu bewerkstelligen. Der einfachste Ansatz ist, eine Programmiersprache zu lernen, mit deren Hilfe HTML-Code erzeugt und dem Client zur Verfügung gestellt wird.

Der erste Kontakt wird sicherlich mit PHP gemacht. Leicht zu erlernen und mit einer Vielzahl von Funktionen. Darüber hinaus existieren eine Menge Frameworks, die den Entwickler bei der Arbeit unterstützen. Unter anderen Zend, CakePHP oder Symfony.

Andere Programmiersprachen, wie Ruby, Java oder Perl ermöglichen ebenfalls fürs Web Programme zu schreiben und einem breiten Publikum zur Verfügung zu stellen. Darunter bekannte wie Ruby on Rails, Spring oder Jifty. Zwar liegt der Fokus auf Python, dennoch ist es wichtig andere Möglichkeiten der Webentwicklung für einen besseren Überblick kurz vorzustellen.

## 2.7 Arbeiten ohne Framework

Die Erstellung einer Anwendung ohne ein entsprechendes Framework ist sehr mühsam. Der Entwickler hat zwar alle Möglichkeiten eine entsprechende Anwendung zu schreiben. Ihm wird aber vieles abgenommen. Viele Frameworks bringen Unterstützung in Domänen, wie z.B. Sicherheit, Benutzerverwaltung und Applikationsadministration. Diese müssen, falls gewünscht neu geschrieben werden und lenke somit von der eigentlichen Applikation ab. Wenn also eine Applikation eine gewissen Komplexität erreicht hat oder erreichen soll, ist es ratsam sich mit einem Framework auseinander zu setzen und von erfahrenen Entwicklern, die diese Arbeit bereits geleistet haben zu profitieren. Die meisten Frameworks arbeiten nach dem Prinzip „Don't repeat your self“ oder „KISS“ keep it simple stupid. Somit werden immer wiederkehrende Aufgaben, die mit komplexen Projekten verbunden sind bereits vorgefertigt geliefert und der Entwickler kann sich sofort dem eigentlichen Projekt widmen.

## 3 Django

### 3.1 Einführung

### 3.2 Installation

### 3.3 Dokumentation

### 3.4 Datenbanksysteme

### 3.5 Templatesystem

### 3.6 JavaScript

### 3.7 Sicherheitsaspekte

### 3.8 Durchführung der Anwendung

## 4 TurboGears

### 4.1

# 5 Pylons

## 5.1 Einführung

## 6 Zope

### 6.1 Einführung



# 7 Web2Py

## 7.1 Einführung

# 8 Ergebnisse / Übersicht

## 8.1 Bewertungskriterien

- Dokumentation
- Installation
- Templatesystem / Darstellung
- Unterstützung von Javascript
- Unterstützung von CSS
- Datenbankbindung
- Handling
- Lizenz

# Literaturverzeichnis

- [1] Farid Hajji, Das Python Praxisbuch - Der große Profi-Leitfaden für Programmierer, 2008
- [2] Python.org, <http://www.python.org/about/>, 28.07.2010
- [3] phpwact.org, [http://www.phpwact.org/php/mvc\\_frameworks](http://www.phpwact.org/php/mvc_frameworks), 28.07.2010
- [4] python.org, <http://wiki.python.org/moin/WebFrameworks>, 28.07.2010
- [5] Pylons, <http://pylonshq.com/>, 29.07.2010
- [6] Django, <http://www.djangoproject.com/>, 29.07.2010
- [7] zope.de, <http://www.zope.de/uber-zope>, 29.07.2010
- [8] web2py.com, <http://www.web2py.com/examples/default/what>, 29.07.2010