

# web2py

batteries included

EDU|days 2019

Donau-Universität Krems

Martin Weissenböck

2. – 3. April 2019

# Addiere zwei Zahlen

```
import java.util.*;
public class einausgabe {
    public static void main(String[] args) {
        Scanner ScIn= new Scanner(System.in);
        System.out.print("1. Zahl: ");
        int i=ScIn.nextInt();
        System.out.print("2. Zahl: ");
        int j=ScIn.nextInt();
        int s=i+j;
        System.out.println(
            "Die Summe von "+i+" und "+j+" ist "+s);
    }
}
```

# Oder so?

```
i = input("1. Zahl: ")  
j = input("2. Zahl: ")  
print "Die Summe von %d und %d ist %d" % (i, j, i+j)
```

Die Antwort ist...



# web2py – das Framework!

- Laden Sie:  
[https://mdipierro.pythonanywhere.com/examples/static/web2py\\_win.zip](https://mdipierro.pythonanywhere.com/examples/static/web2py_win.zip)
- Nach dem Laden in Verzeichnis entpacken – fertig!  
Vorschlag: c:\web2py

Ein komplettes Framework („Batteries included“)  
zum Erstellen einer Website.

Umsetzung des Prinzips

**Trennung in „Model - View - Controller“**

# Ein php-Beispiel

```
<html><body><h1>Records</h1><?
mysql_connect(localhost,username,password);
@mysql_select_db(database) or die( "Unable to select database");
$query="SELECT * FROM contacts";
$result=mysql_query($query); mysql_close();

$i=0;
while ($i < mysql_numrows($result)) {
    $name=mysql_result($result,$i,"name");
    $phone=mysql_result($result,$i,"phone");
    echo "<b>$name</b><br>Phone:$phone<br /><br /><hr /><br />";
    $i++;
}

?>
</body></html>
```

# ... und nun in Python mit web2py

```
def index():  
    return HTML(BODY(H1('Records'),  
                    db().select(db.contacts.ALL)))
```

*Und damit das wirklich beeindruckend ist, bleibt die restliche Seite leer.*

web2py starten



# Los geht's!

- Nicht vergessen:  
nach dem Download: entpacken
- Starten über `web2py.exe` (windows) oder `web2py.app` (osx)

# Let's start!

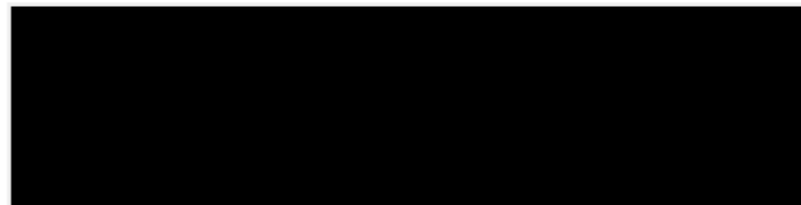
## WEB2PY

Version 2.17.1-stable+timestamp.2018.08.06.01.02.56  
Created by Massimo Di Pierro, Copyright 2007-2018

Server IP:  Local (IPv4) (127.0.0.1)  
 Local (IPv6) (::1)  
 Public (fe80::6c31:e010:ec37:1871%19)  
 Public (192.168.0.84)  
 Public (0.0.0.0)

Server Port:

Choose Password:



# Ein Passwort festlegen:


- Choose Password:  
123 reicht bei lokaler Installation
- start server

# Die Startseite > admin

## Willkommen bei web2py!

Wie sind Sie hier her gelangt?

1. web2py wird erfolgreich ausgeführt
2. Sie haben die URL `/welcome/default/index` besucht
3. Welche die Funktion `index()` in der Datei `web2py/applicat`
4. Die Ausgabe der Datei ist ein "dictionary", welches vom "\
5. Sie können diese Anwendung verändern und Ihren Bedürf

 admin

Online Beispiele

web2

# Beispiel: DVD-Sammlung verwalten

## Neue einfache Anwendung

Name der Anwendung:

# Modelle

# Modelle

- Beschreibung der Datenbanktabellen:
- Name der Tabelle: dvd
- Name der Felder:
  - Titel
  - Bewertung
  - Datum

# Datenbank-Tabelle

- Modelle: db.py <Bearbeiten>
- Am Ende einfügen.  

```
db.define_table('dvd',  
    Field('titel'),  
    Field('bewertung', 'integer'),  
    Field('datum', 'datetime'),  
)
```
- „Speichere Datei“ **nicht vergessen!**



# Modelle

Datenbankadministration

graph model

Bearbeiten



db.py

Bearbeiten




menu.py

Erstellen

# Tabelle „dvd“ erstellen

```
152 # -----
153 # after defining tables, uncomment below to enable
154 # -----
155 # auth.enable_record_versioning(db)
156
157 db.define_table('dvd',
158     Field('titel'),
159     Field('bewertung', 'integer'),
160     Field('datum', 'datetime'),
161 )
162
```

# Speichern nicht vergessen!

Speichere Datei:  Gespeicherter Datei-

**Breakpoint aktivieren/deaktivieren**

```
1 # -*- coding: utf-8 -*-
```

```
2
```

```
3
```

```
# -----
```

# Weiter mit „Bearbeiten“

Seite

Bearbeiten

Über

Datei gespeichert auf Tue Sep 25 2012

# Weiter mit „Datenbankadministration“

## Modelle

Datenbankadministration

sql.log

graph model

Bearbeiten



db.py definiere Tabellen dvd

Bearbeiten



menu.py

Erstellen

# db.dvd > Neuer Eintrag

## Verfügbare Datenbanken und Tabellen

Tables	Hooks
» <b>db.auth_user</b>	Neuer Eintrag
» <b>db.auth_group</b>	Neuer Eintrag
» <b>db.auth_membership</b>	Neuer Eintrag
» <b>db.auth_permission</b>	Neuer Eintrag
» <b>db.auth_event</b>	Neuer Eintrag
» <b>db.auth_cas</b>	Neuer Eintrag
» <b>db.dvd</b>	Neuer Eintrag

# Ein paar DVDs eintragen

Datenbank **db** Tabelle **dvd**

Neuer Eintrag

Titel

Bewertung

Datum

Submit

# Die erste DVD

## Neuer Eintrag

Titel

Ghost Busters

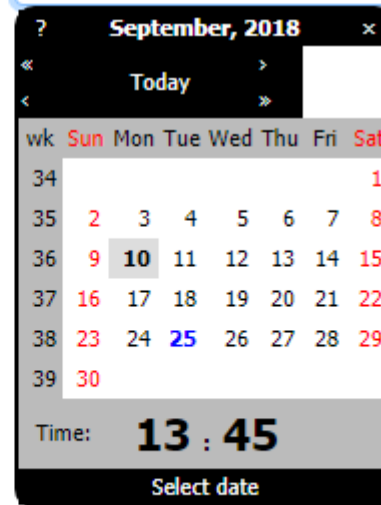
Bewertung

1

Datum

2018-09-10 13:45:00

Submit





# Alle DVDs anzeigen

3 ausgewählt

<b>dvd.id</b>	<b>dvd.titel</b>	<b>dvd.bewertung</b>	<b>dvd.datum</b>
1	Ghost Busters	1	2018-09-10 13:45:00
2	Gladiator	2	2018-09-10 14:00:00
3	Wiki und die ...	3	2018-09-11 08:00:00

# Eintrag ändern

- In der Spalte dvd.id auf die Nummer des Datensatzes klicken
- Bewertung auf 2 ändern
- Speichern

# Eintrag ändern

Datenbank **db** Tabelle **dvd** id des Eintrags **3**  
Diesen Eintrag editieren

Id

3

Titel

Wiki und die starken Männer

Bewertung

2

Datum

2018-09-11 08:00:00

Auswählen um zu löschen

Submit

**Controller**

# Controller

- To control = steuern
- Sammlung von Python-Dateien
- Jede Python-Datei enthält Python-Funktionen
- Jede Funktion ist für die Steuerung eines bestimmten Ablaufs zuständig

# Funktion zum Zeigen der DVDs

- In der Datei default.py einfügen:
- ```
def zeigen():  
    alledaten = db(db.dvd.id>0).select()  
    return dict(zeigedaten=alledaten)
```

# Controller „default.py“ bearbeiten

## Controllers

Test

crontab

Bearbeiten



appadmin.py stellt zur Verfüg

Bearbeiten



default.py stellt zur Verfügung

# Funktion „zeigen“ einfügen

```
# -----  
# This is a sample controller  
# this file is released under public domain and you c  
# -----  
  
def zeigen():  
    alledaten = db(db.dvd.id>0).select()  
    return dict(zeigedaten=alledaten)  
  
# ---- example index page ----  
def index():  
    response.flash = T("Hello world")  
    return dict(message=T('Welcome to web2py!'))
```



# Aufruf einer web2py-Webseite

- Struktur (vereinfacht):  
Prot://Hostadresse:Port/app/controller/function
  - Prot = Protokoll: http
  - Hostadresse: localhost oder 127.0.0.1
  - Port: 8000
  - app = Application = Anwendung: DVD
  - controller: default
  - Function= Funktion: zeigen
- Aufruf daher:  
<http://localhost:8000/DVD/default/zeigen>

# Inhalt der Tabelle „dvd“

← → ↻ ⓘ 127.0.0.1:8000/DVD/default/zeigen

Startseite

web2py

## Zeigen

| <b>dvd.id</b> | <b>dvd.titel</b> | <b>dvd.bewertung</b> | <b>dvd.datum</b>    |
|---------------|------------------|----------------------|---------------------|
| 1             | Ghost Busters    | 1                    | 2018-09-10 13:45:00 |
| 2             | Gladiator        | 2                    | 2018-09-10 14:00:00 |
| 3             | Wiki und die ... | 2                    | 2018-09-11 08:00:00 |

**View = Ansicht**

# Ansicht erzeugen

- Ansichten <Erstellen>  
Erzeuge Datei mit Dateinamen:
- default/zeigen <Erstellen>
- <Bearbeiten> default/zeigen.html
- Ändern auf

```
  {{extend 'layout.html'}}  
  <h1> DVD-Liste </h1>  
  {{=zeigedaten}}
```
- „Speichere Datei“ **nicht vergessen!**
- <Versuche view>

# View erzeugen

## Ansichten

Die Präsentationsschicht, Views sind auch bekannt als Vorlagen/Templates

Download layouts from repository

# View erzeugen

- Auf „Erstellen“ klicken

Erstellen

Erzeuge Datei mit Dateinamen:

default/zeigen|

Erstellen

# Datei „default/zeigen.html“

zeigen.html ✕

Speichere Datei:  Gespeicherter Datei-Hash:   
**bearbeite Controller:** `default.py`

```
1  {{extend 'layout.html'}}
2  <h1> DVD-Liste </h1>
3  {{=zeigedaten}}
4
```

So schaut es aus

# DVD-Liste

<b>dvd.id</b>	<b>dvd.titel</b>	<b>dvd.bewertung</b>	<b>dvd.datum</b>
1	Ghost Busters	1	2018-09-10 13:45:00
2	Gladiator	2	2018-09-10 14:00:00
3	Wiki und die ...	2	2018-09-11 08:00:00



# Schönere Ausgabe

- web2py enthält viele Werkzeuge zur Vereinfachung der Programme
- Beispiel: Verwenden von SQLFORM.grid
- Ändern des Controllers

```
# -----  
# This is a sample controller  
# this file is released under public domain  
# -----  
  
def zeigen():  
    alledaten = SQLFORM.grid(db.dvd)  
    return dict(zeigedaten=alledaten)
```

# Verwenden von SQLFORM

## DVD-Liste

Id	Titel	Bewertung	Datum	
1	Ghost Busters	1	2018-09-10 13:45:00	<input type="button" value="Ansicht"/>
2	Gladiator	2	2018-09-10 14:00:00	<input type="button" value="Ansicht"/>
3	Wiki und die star...	2	2018-09-11 08:00:00	<input type="button" value="Ansicht"/>

Export:

# SQLFORM erlaubt...

- Sortieren (Klick auf den Spaltenkopf)
- Ändern der Sortierreihenfolge (nochmals Klick auf den Spaltenkopf)
- Suchen nach verschiedenen Kriterien
- Anzeigen aller Details eines Datensatzes
- Ändern eines Datensatzes
- Hinzufügen und Löschen von Datensätzen

# Layout ändern

- Layout aussuchen und mit get laden, z. B. Commission

## Ansichten

Download layouts from repository

Bearbeiten



`__init__.py`

Bearbeiten



`appadmin.html` erweitert `layout.html`

# Layout installieren

## Plugins ?

Keine Plugins vorhanden

Download plugins from repository

Upload

- Plugin aus dem Downloadverzeichnis wählen
- Seite neu laden

# So sieht es aus:

Startseite   Meine Seiten   Diese App »   web2py.com »   Dokumentation »

Community »

## DVD { by Free CSS Templates }

### DVD-Liste

Search Clear

3 records found

<u>Id</u>	<u>Titel</u>	<u>Bewertung</u>	<u>Datum</u>
1	Ghost Busters	1	2018-09-10 13:45:00 <a href="#">Ansicht</a>
2	Gladiator	2	2018-09-10 14:00:00 <a href="#">Ansicht</a>
3	Wiki und die star... 2		2018-09-11 08:00:00 <a href="#">Ansicht</a>

Export: [CSV](#) [CSV \(hidden cols\)](#) [HTML](#) [JSON](#) [TSV \(Spreadsheets\)](#) [TSV \(Spreadsheets, hidden cols\)](#) [XML](#)

### Aliquam tempus

Mauris vitae nisl nec metus placerat perdiet est. Phasellus dapibus semper urna. Pellentesque ornare, orci in consectetur hendrerit, volutpat.

### Calendar

March 2008

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

[« Feb](#) [Apr »](#)

### Turpis nulla

# Detailansicht

## DVD-Liste

3 records found

<u>Id</u>	<u>Titel</u>	<u>Bewertung</u>	<u>Datum</u>	
1	Ghost Busters	1	2018-09-10 13:45:00	<a href="#">Ansicht</a>
2	Gladiator	2	2018-09-10 14:00:00	<a href="#">Ansicht</a>
3	Wiki und die star...	2	2018-09-11 08:00:00	<a href="#">Ansicht</a>

Export:[CSV](#)[CSV \(hidden cols\)](#)[HTML](#)[JSON](#)[TSV](#) ([Spreadsheets](#))[TSV](#) ([Spi](#)

# Das Buch



# The Book

<http://www.web2py.com/book>

- 00 Preface
- 01 Introduction
- 02 The Python language
- 03 Overview
- 04 The core
- 05 The views
- 06 The database abstraction layer
- 07 Forms and validators

# The Book

- 08 Emails and SMS
- 09 Access Control
- 10 Services
- 11 jQuery and Ajax
- 12 Components and plugins
- 13 Deployment recipes
- 14 Other recipes
- 15 Helping web2py

Ein wenig Theorie

# Das Datenmodell

- db = Instanz der Klasse DAL  
Verbindung zur Datenbank, „Connector“
- Abstrakte Zwischenschicht
- Wechsel zu einer anderen Datenbank  
(theoretisch) einfach möglich

# Datenbank Tabelle anlegen

- `db.define_table` = Aufruf der Methode zum Erzeugen einer neuen Tabelle
- `db.define_table('dvd' ...`  
dvd = Name der Tabelle
- `Field('titel')`  
Spalte 'titel', Typ (Standard): 'string'
- `Field('bewertung', 'integer')`  
Spalte 'bewertung', Typ: 'integer'
- Viele weitere Parameter möglich

# Die Abfrage im Controller

```
db(db.dvd.id>0).select()
```

- `db.dvd` oder `db['dvd']` ist die Tabelle
- `db.dvd.id` oder `db.dvd['id']` ist das Feld `id`
- `db.dvd.id>0` heißt in `web2py` "Query"  
Entspricht der `WHERE`-Klausel von `SQL`
- `db(db.dvd.id>0)` ist ein "Set"
- `db(db.dvd.id>0).select()` liefert "Rows" = "Zeilen"
- Vergleich mit `PHP`: `$query="SELECT * FROM dvd WHERE id>0";`

# View = Ansicht

- HTML mit eingefügten Python-Befehlen, in `{ { und } }` eingeschlossen
- `{{extend 'layout.html'}}`  
lädt den "Rahmen" für die Applikation
- `<h1> DVD-Liste </h1>` Normaler HTML-Befehl
- `{{=zeigedaten}}`  
wird in einen Python-print-Befehl umgewandelt. Controller liefert ein dict. Geschrieben wird der Wert, dessen Key `zeigedaten` ist

# DVD-Liste erweitern

- <DIESE APP> <CONTROLLER>

- Neue Funktion:

```
def dvdlisteerweitern():
```

```
    form = SQLFORM(db.dvd)
```

```
    if form.process().accepted:
```

```
        response.flash = 'form accepted'
```

```
    elif form.errors:
```

```
        response.flash = 'form has errors'
```

```
    else:
```

```
        response.flash = 'please fill out the form'
```

```
    return dict(form=form)
```

- „Speichere Datei“ **nicht vergessen!**



# Ansicht hinzufügen

- <Erstellen> Erzeuge Datei mit Dateinamen
- default/dvdlisterweiteren

```
  {{extend 'layout.html'}}  
  <h1>DVD Liste erweitern</h1>  
  {{=form}}
```
- „Speichere Datei“ **nicht vergessen!**

# DVD bearbeiten

- Und in der Ansicht default/dvdbearbeiten.html anlegen:

```
{{extend 'layout.html'}}  
<h1>DVD bearbeiten</h1>  
{{=grid}}
```

- Neue Funktion im Controller hinzufügen:

```
def dvdbearbeiten():  
    grid = SQLFORM.grid(db.auth_user, user_signature=False)  
    return dict(form=form)
```

# DVD bearbeiten

## DVD bearbeiten

+ ADD RECORD

SEARCH

CLEAR

Id	Titel	Bewertung			
1	The Jungle Book	1	 ANSICHT	 EDIT	 DELETE
2	Enigma	1	 ANSICHT	 EDIT	 DELETE
3	Flug 7500	2	 ANSICHT	 EDIT	 DELETE

Export:

CSV

CSV (HIDDEN COLS)

HTML

JSON

TSV (SPREADSHEETS)

TSV (SPREADSHEETS, HIDDEN COLS)

XML

# Mögliche Erweiterungen

- Personen anlegen
- Verleihliste
- Bewertungsliste

# web2py-Projekte

- web2py wurde zwar für Schulungszwecke entwickelt, erlaubt aber auch sehr große und anspruchsvolle Applikationen
- Dutzende Tabellen mit vielen Tabellenspalten
- Hunderte von Funktionen in vielen „Controllern“
- Viele Applikationen, die gemeinsam ein Paket bilden

# Eigene Erfahrungen

- Initiative „SCHUL.InfoSMS“  
(<http://www.infosms.org>):  
Verbesserung der Kommunikation zwischen Schule unter Eltern mit elektronischen Mitteln
- Komplette in web2py und somit in Python realisiert
- Interessenten an der Initiative sind sehr willkommen: [office@infosms.org](mailto:office@infosms.org)

# SCHUL.infosms

- Seit 11 Jahren in Betrieb
- An über 100 Schulen in Verwendung
- Verständigung der Eltern per SMS, E-Mail, Telegram und App
- Auch für lange Texte (nicht per SMS)

# SCHUL.infoservice

- Die Erweiterung durch das Messenger-Programm Telegram und mit einer App
- Möglichkeit der bidirektionalen Kommunikation
- Einsatz von Bots
- Viele Anwendungen und Erweiterungen, Vorschläge der Schulen



# Artikel über InfoSMS

In der aktuellen Ausgabe der PCNEWS gibt es drei Beiträge zum Thema InfoSMS:

- Argumente für InfoSMS:  
[http://d.pcnews.at/\\_pdf/n1580009.pdf](http://d.pcnews.at/_pdf/n1580009.pdf)
- Unterschrift der Eltern - der Vorschlag wurde vom Wiener Stadtschulrat genehmigt.  
[http://d.pcnews.at/\\_pdf/n1580010.pdf](http://d.pcnews.at/_pdf/n1580010.pdf)
- Umfragen - ein neuer Baustein für rasche Meinungsumfragen  
[http://d.pcnews.at/\\_pdf/n1580011.pdf](http://d.pcnews.at/_pdf/n1580011.pdf)

# Weitere Artikel

- **Klassenkassa.**  
Damit werden Beträge verwaltet  
[http://d.pcnews.at/\\_pdf/n1560007.pdf](http://d.pcnews.at/_pdf/n1560007.pdf)
- **Chatbots**  
Zum Thema "dialogfähige Programme"  
[http://d.pcnews.at/\\_pdf/n1550004.pdf](http://d.pcnews.at/_pdf/n1550004.pdf)

# Weitere Schritte mit web2py

# Wie geht es weiter?

- Applikation auf einen öffentlich zugänglichen Server stellen:
  - Eigener Server – Installationsanleitungen vorhanden
  - Gratisserver, vor allem zum Testen
- Handbuch lesen:
  - <http://web2py.com/book> (6. Auflage)
  - [https://dl.dropboxusercontent.com/u/18065445/web2py/web2py\\_manual\\_5th.pdf](https://dl.dropboxusercontent.com/u/18065445/web2py/web2py_manual_5th.pdf) (5. Auflage)

# Zusammenfassung: warum web2py?

# Anwendungsbereiche

- web2py wurde für Lehrzwecke entwickelt
  - Eigene Einstellungen für Vortragende
- web2py kann auch für große „echte“ Projekte verwendet werden

# Didaktische Komponenten

- Python  
Der Sourcetext liefert viele Anregungen
- HTML und CSS  
Auch am Beispiel unterschiedlicher Vorlagen
- Datenbanken  
Modellierung, Abfragen, unterschiedlichen  
Datenbankprogramme im Vergleich
- Webservices

Danke für Ihr Interesse!

Viel Erfolg beim Einsatz von  
Python und web2py

Martin Weissenböck