



Webframeworks für Python

Vorstellung und Vergleich

Andreas Knöpfle, Bastien Sachs und Tobias Schmid

Institut für Informatik

28. November 2012

Einstiegs-Idee:

viele Logos der Webframeworks für Python

Stichpunkte:

- In Python geschriebene Webframework gibt es wie „Sand am Meer“
- Warum? Einblick gibt unser Vortrag

Gliederung

1. Python – Programmiersprache fürs Web?

2. Webframeworks für Python

3. Fazit

Python – Programmiersprache fürs Web?

- 1991 erstmals erschienen, aktuelle Versionen: 3.3.0 (September 2012), 2.7.3 (April 2012)
- Entwurfsphilosophie betont Programmlesbarkeit
 - ▶ Blöcke durch gleiche Einrückung begrenzt
 - ▶ relativ wenige Schlüsselwörter
- objektorientierte, aspektorientierte und funktionale Programmierung
- dynamische Datentypen, garbage collection
- große, umfangreiche Standardbibliothek “batteries included”

Logo Python

Python – Programmiersprache fürs Web?

eventl. Codebeispiele, Gegenüberstellungen, Erklärung

Web Server Gateway Interface (WSGI)

* Apache (mod_wsgi), nginx (uWSGI)

Bild

Unterteilung Webframeworks für Python

Popular Full-Stack Frameworks, Other Full-Stack Frameworks, Basic Frameworks * Django

* CherryPy, CubicWeb, Flask, Grok, Plone, Pylons, Pyramid, TurboGears, web2py, Zope 2

* Bottle, Karrigell, Nagare, Pyjamas, Quixote, Spyce, Tornado, TwistedWeb, Web.py

* Nicht mehr aktiv: BlueBream, Nevow, Webware

Hauptteil:

Vor- und Nachteile einiger weniger Frameworks aufzeigen

(konkrete) Lösungsansätze für bestimmte Probleme/Vergleichskriterien

Persistenzschicht

- SQL ORM

Persistenzschicht

- SQL ORM
 - Django built-in ORM (Django)

Persistenzschicht

- SQL ORM
 - Django built-in ORM (Django)
 - SQLAlchemy (Grok, Pylons, Zope)

Persistenzschicht

- SQL ORM
 - Django built-in ORM (Django)
 - SQLAlchemy (Grok,Pylons,Zope)
 - Storm (Grok)

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)
 - ▶ DAL (web2py)

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)
 - ▶ DAL (web2py)
- MongoDB ORM

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)
 - ▶ DAL (web2py)
- MongoDB ORM
 - ▶ Django MongoDB Engine (Django)

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)
 - ▶ DAL (web2py)
- MongoDB ORM
 - ▶ Django MongoDB Engine (Django)
 - ▶ Ming (TurboGears)

Persistenzschicht

- SQL ORM
 - ▶ Django built-in ORM (Django)
 - ▶ SQLAlchemy (Grok,Pylons,Zope)
 - ▶ Storm (Grok)
 - ▶ SQLAlchemy (Pylons,TurboGears)
 - ▶ DAL (web2py)
- MongoDB ORM
 - ▶ Django MongoDB Engine (Django)
 - ▶ Ming (TurboGears)
- ...

Templates, I18N, L10N

- Template Engines:

Templates, I18N, L10N

- Template Engines:
 - Python built-in

Templates, I18N, L10N

- Template Engines:
 - Python built-in
 - Django template language

Templates, I18N, L10N

- Template Engines:
 - ▶ Python built-in
 - ▶ Django template language
 - ▶ Cheetah (Django, Turbogears, Pylons)

Templates, I18N, L10N

- Template Engines:
 - ▶ Python built-in
 - ▶ Django template language
 - ▶ Cheetah (Django, Turbogears, Pylons)
 - ▶ Myghty

Templates, I18N, L10N

- Template Engines:
 - Python built-in
 - Django template language
 - Cheetah (Django, Turbogears, Pylons)
 - Myghty
- I18N + L10N

Templates, I18N, L10N

- Template Engines:
 - Python built-in
 - Django template language
 - Cheetah (Django, Turbogears, Pylons)
 - Myghty
- I18N + L10N
 - gettext

Konfiguration, Routing

Formulare, Validierung

Sicherheitsmechanismen

Bootstrapping, Scaffolding, Erweiterbarkeit

Extras: WebServices, Caching, Tests

Kriterienübersicht

- * Vergleichstabellen (Django vs. ...)

Fazit

Je Anforderungen an das Webframework (“Taste”)

...

Quellen der Abbildungen

1. Innenhof Informatik <http://www.flickr.com/photos/bennybenny/3597853896/>

alle URLs aufgerufen am 14. November 2012.

Quellen

<http://www.infoworld.com/d/application-development/pillars-python-six-python-web-frameworks-compared-169442>

<http://wiki.python.org/moin/WebFrameworks>

<http://wiki.python-forum.de/Web-Frameworks>

<http://blog.ianbicking.org/turbogears-and-pylons.html>

Ende der Präsentation

- Vielen Dank für Ihre Aufmerksamkeit.
- offene Fragen?
- Diskussion
 - Kritik, Anregungen