



Tutorium

Skriptsprachen

2009 - Max Maurer



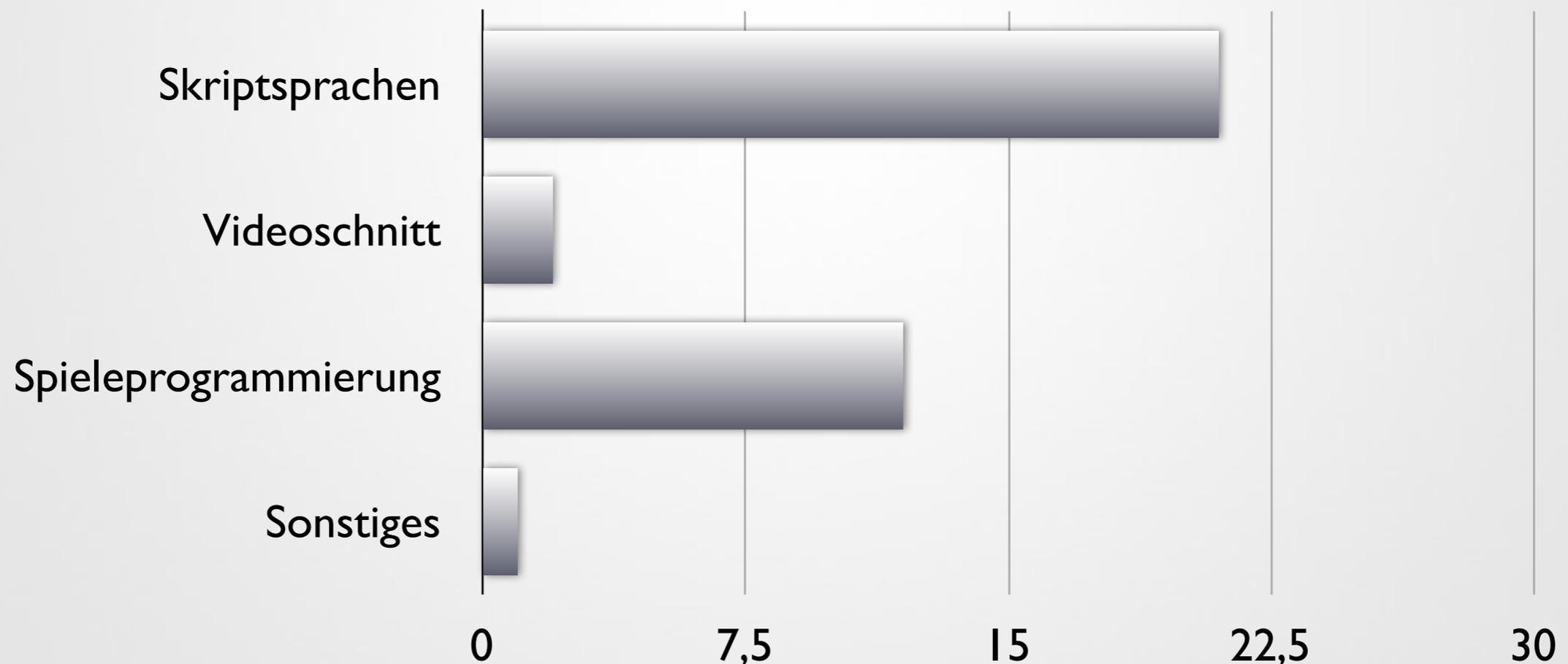
Organisatorisches



Warum...?



- Zusatzveranstaltung zur normalen Lehre
- Themenfindung über Online-Umfrage





Studentenbetreuer...?



- Individuelle Betreuung von Studenten
- Grundsätzlich Lehrstuhlübergreifend
- Teilweise aus Studiengebühren finanziert
- „Zur Verbesserung der Studienbedingungen“
- Ideen für neue Veranstaltungen?



Scheine und Credits



- ... nada ;-)
- freiwillige Zusatzveranstaltung
- Kenntnisse wertvoll für..
 - private Programmierung
 - generelles Programmierverständnis
 - andere Uni-Veranstaltungen



Termine und Ablauf



- Beginn: 2.11.2009
- 5 Termine
 - Jeweils Montags
 - 2.11., 9.11., 16.11., 23.11., 30.11.
- Heute: Überblick und Einführung
- Dann jede Woche 1 Sprache

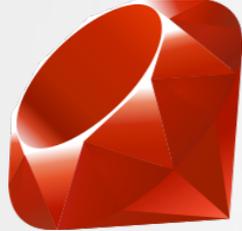


Termine und Ablauf



Überblick

2. November



Ruby

9. November



Python

16. November

Perl

23. November



PHP

30. November



Skriptsprachen (Allgemein)

Literatur:

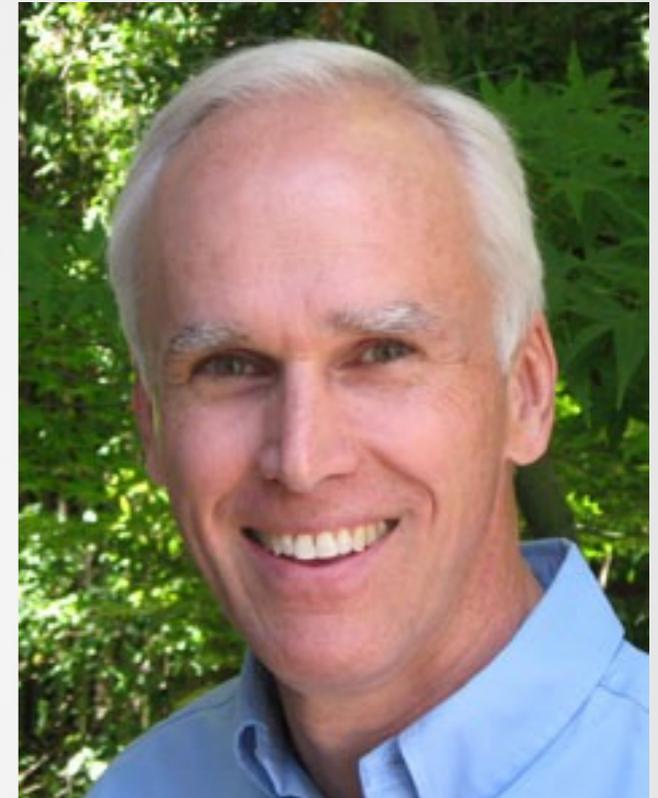
John K. Ousterhout: Scripting: Higher Level Programming for the 21st Century

<http://home.pacbell.net/ouster/scripting.html>



Skriptsprachen

“
[...] scripting languages are designed for gluing: they assume the existence of a set of powerful components and are intended primarily for connecting components together.
”



John K. Ousterhout



Systemprogrammiersprachen

|||00|

- Höhere Programmiersprachen
 - Abstrakter als Assemblersprachen
 - C, C++
- Zweck der Systemsprachen
 - Identisch zu Assembler: Datenstrukturen und Algorithmen entwickeln
 - Aber: Schnellere Entwicklung
 - Starke Typisierung
- Skriptsprachen verfolgen einen anderen Zweck



Sprachvergleich

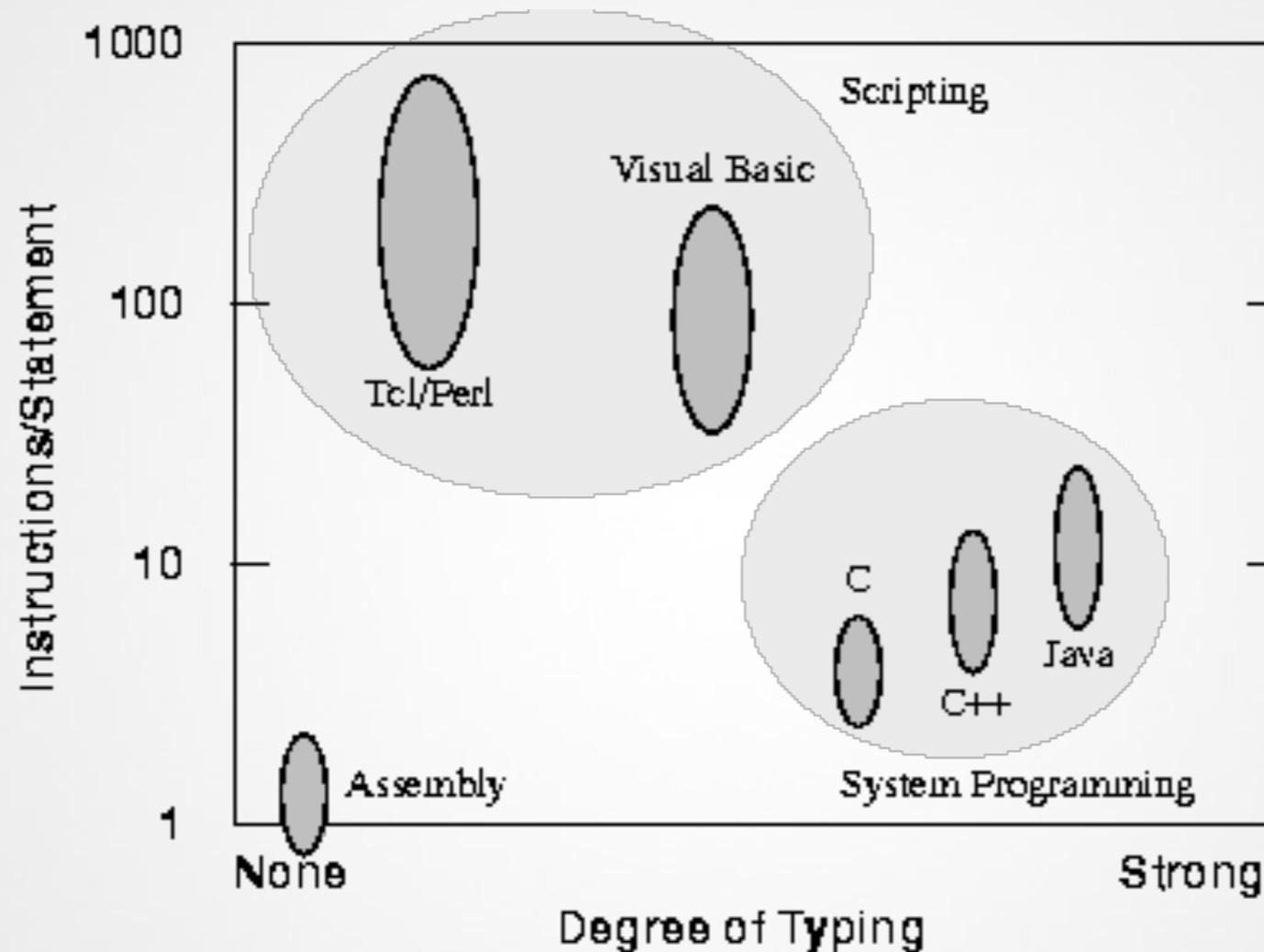


Figure 1. A comparison of various programming languages based on their level (higher level languages execute more machine instructions for each language statement) and their degree of typing. System programming languages like C tend to be strongly typed and medium level (5-10 instructions/statement). Scripting languages like Tcl tend to be weakly typed and very high level (100-1000 instructions/statement).



Skriptsprachen



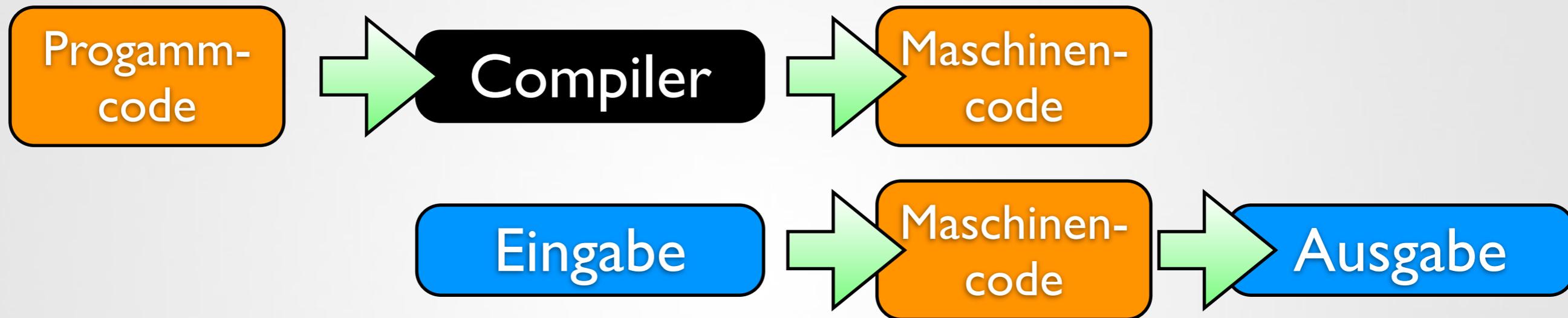
- Klassisch mit anderen Komponenten verwendet
 - Skript benutzt andere Programme geschrieben in höherer Sprache
 - z.B. Unix shell scripts
 - nur „Mittel zum Zweck“
- Heute vollwertige eigene Programmiersprachen
 - ähneln den „Hochsprachen“ immer mehr
 - meistens gut für Einsteiger geeignet
 - Meist leistungsfähig auf einem bestimmten Gebiet



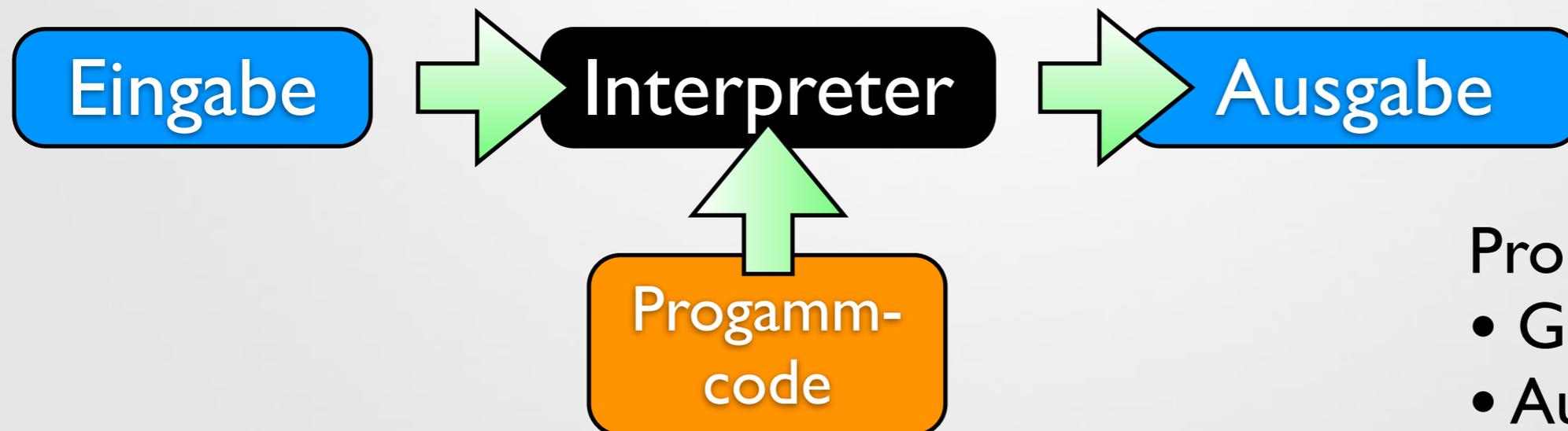
Kompilieren vs. Interpretieren

|||00|

Kompilieren:



Interpretieren:



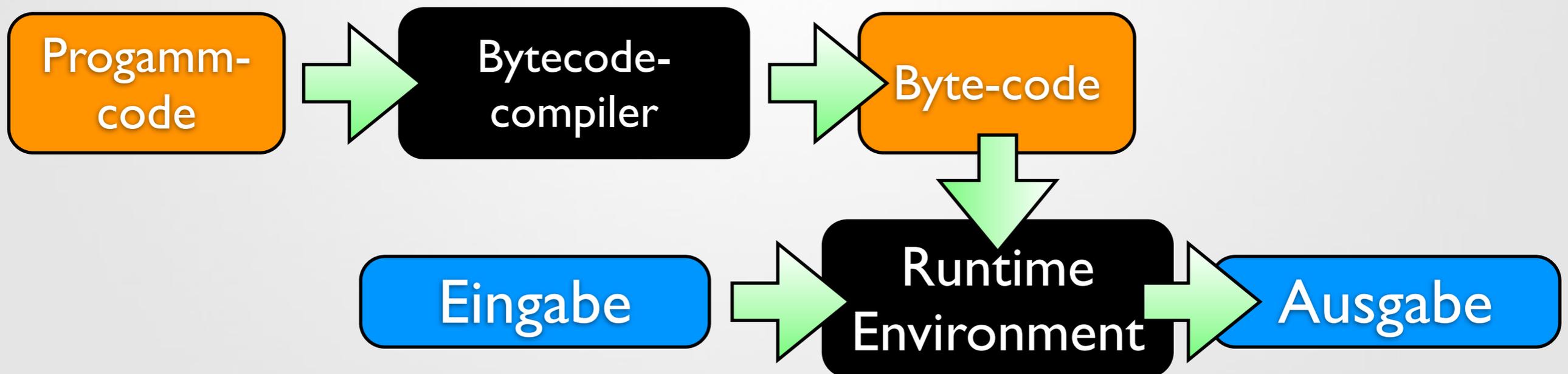
Probleme:

- Geschwindigkeit
- Auslieferung



Bytecode Compiler und Runtime-Environments

- Bytecode Compiler erzeugt „Zwischencode“
 - Noch nicht maschinenausführbar
 - Aber kein Programmcode mehr
- Zur eigentlichen Ausführung wird ein Runtime Environment verwendet (bei Java: JRE)
- Interpretiert Bytecode und macht daraus Maschinencode





Programmierparadigmen

- Programmierstil der verwendet wird
- Viele verschiedene Möglichkeiten
 - Imperativ (Java) vs Funktional (SML)
 - Statisches vs dynamische Typisierung
 - Starke vs. schwache Typisierung
 - Fehlerbehandlung



Imperativ vs. Funktional

- Funktionale Programmierung
 - Programmablauf ist die Auswertung mathematischer Funktionen
 - Beispiel: SML
 - meist Rekursion
- Imperative Programmierung
 - Statusänderung von Variablen erzeugt Code
 - Beispiel: Java
 - meist Iteration

```
fun factorial n =  
    if n = 0 then 1  
    else n * factorial (n-1)
```

```
int factorial(n)  
    sum = n;  
    while (n>0)  
        n=n-1;  
        sum = sum*n  
    return sum
```



Static vs. dynamic typing

- Wo sind die Variablentypen vergeben?
- Statische Typisierung
 - Prüfung der Typen vor der Ausführung des Programmes
 - Typen müssen festgelegt werden
- Dynamische Typisierung
 - Variablentypen werden zur Laufzeit vergeben je nachdem welchen Typ der Wert der Zuweisung hat
 - `a = 3` oder `a = „Hallo“`

```
int a = 40;  
print typeof(a); // int  
a = „Hallo“; -> Fehler!
```

```
a = 40;  
print typeof(a); // int  
a = „Hallo“;  
print typeof(a); // String
```



Strong vs. weak typing

- Strenge oder nachlässige Typprüfung
- Beispiel $a = 3 + „3“$
- Strong typing
 - Ungleiche Typen können nicht zusammen verwendet werden
 - Verwendung erzeugt Fehler (auch bei dynamischen Datentypen)
- Weak typing
 - Anpassung wird vorgenommen ($a = 6$)
 - Problem: $a = 3 + 0.3$



Variablen: Referenzen oder Werte

- Die Zuweisung von Variablen kann entweder als Referenz erfolgen oder als Wert
- Referenz
 - Modifikationen wirken sich auf das Ursprungsobjekt aus
- Wertzuweisung
 - Man arbeitet mit einer „Kopie“ der Variable
 - Modifikationen wirken sich nicht aus
- (De-)Referenzierungsoperatoren

```
a = [1,2,3,4]
b = a;
b[1] = 5;
print b; // 1,5,3,4
print a; // 1,5,3,4
```

```
a = [1,2,3,4]
b = a;
b[1] = 5;
print b; // 1,5,3,4
print a; // 1,2,3,4
```



Fehlerbehandlung

- Methode zur Behandlung von Ausnahmen
- Getrennte Verarbeitung von Fehlern zum Ausgabecode
- Funktion mit Rückgabewert wird im Ablauf unterbrochen
- Der Fehler „bubblet“ durch den Stack nach oben
- Behandlungsmöglichkeit ansonsten Programmabbruch

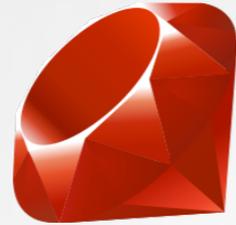
```
function summe(a,b):int {  
    if (typeof(a)!=int) {  
        throw new Exception();  
    }  
    return a+b;  
}  
  
try {  
    print summe(„Hallo“,3);  
} catch (Exception) {  
    print „Uuups“;  
}
```



Skriptsprachen (Vergleich)



Die Sprachen im Vergleich



Ruby



Python

Perl



PHP

Datentypen	dynamisch	dynamisch	dynamisch	dynamisch
Typisierung	strong	strong	weak	weak
Variablen- übergabe	Referenz	Referenz	Wert	Wert
Codestruktur	do/end {	Tabs	}	}
Fehler	begin/ rescue	try/except	eval/\$@	try/catch



Softwarekomponenten



Web-Server



XAMPP

- Skriptsprachen können zur Generierung von Webseiten benutzt werden
- Testen mit Webserver auf einem Testrechner
- XAMPP ist eine Paket aus wichtigen Komponenten (Webserver, Datenbankserver, Runtimes für Skriptsprachen, Grafikwerkzeuge, Analysetools)
- Download unter www.apachefriends.org
- Verfügbar für Windows, Mac OS X und Linux



Was macht ein WebServer?



? Anfrage:
„GET /index.html“

Web Server
= Computer + Software
z.B. Apache



Browser



Datei kommt zurück:
index.html





Webserver + Skriptsprache



? Anfrage:
„GET /index.php“



Web Server

= Computer + Software
z.B. Apache

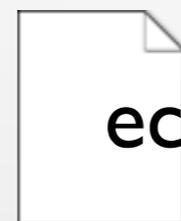


Browser

Interpretiert das Skript
und schickt das Ergebnis
zurück



Ergebnis (nicht Datei!)
kommt zurück!



```
echo „Hallo!“
```



Hallo!



Web-Server



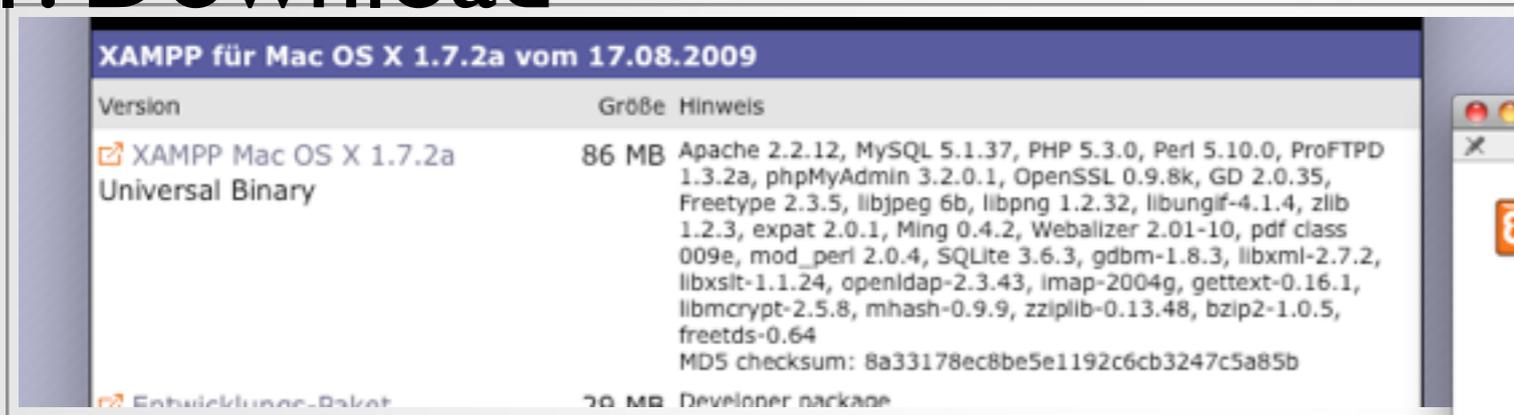
XAMPP

- Skriptsprachen können zur Generierung von Webseiten benutzt werden
- Testen mit Webserver auf einem Testrechner
- XAMPP ist eine Paket aus wichtigen Komponenten (Webserver, Datenbankserver, Runtimes für Skriptsprachen, Grafikwerkzeuge, Analysetools)
- Download unter www.apachefriends.org
- Verfügbar für Windows, Mac OS X und Linux



XAMPP-Installation

1. Download



2. Installation

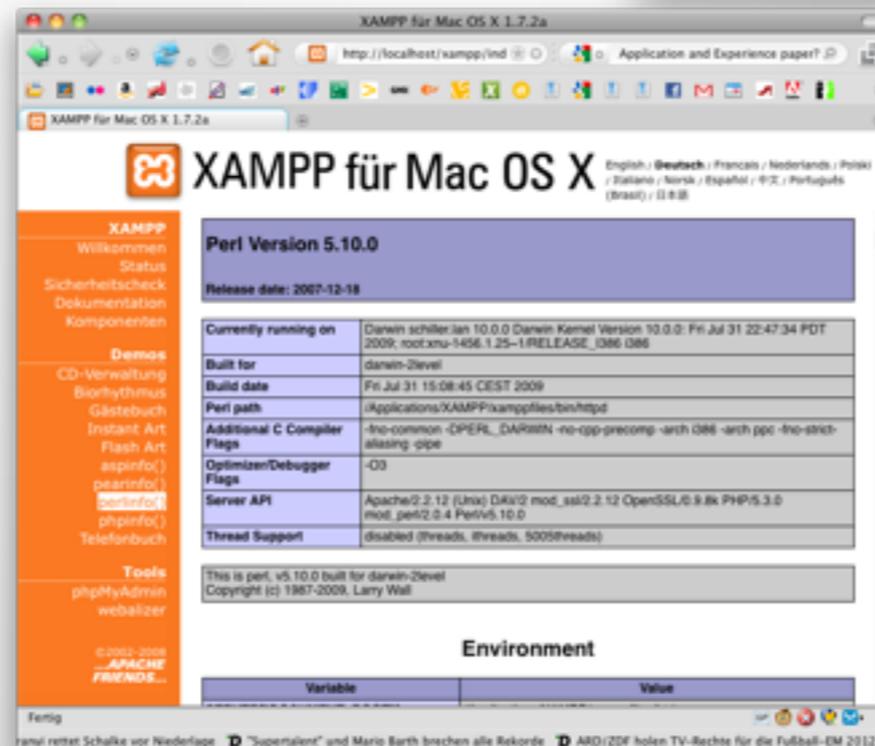


3. Starten



4. Testen

<http://localhost/>





Ruby-Installation

auf dem Mac 



- bereits installiert!
- Vom Terminal aus starten mit „irb“

```
schiller:~ Max$ irb
>> puts "Ruby laeuft ja schon!"
Ruby laeuft ja schon!
=> nil
>> quit
schiller:~ Max$ █
```



Ruby-Installation

unter  Windows

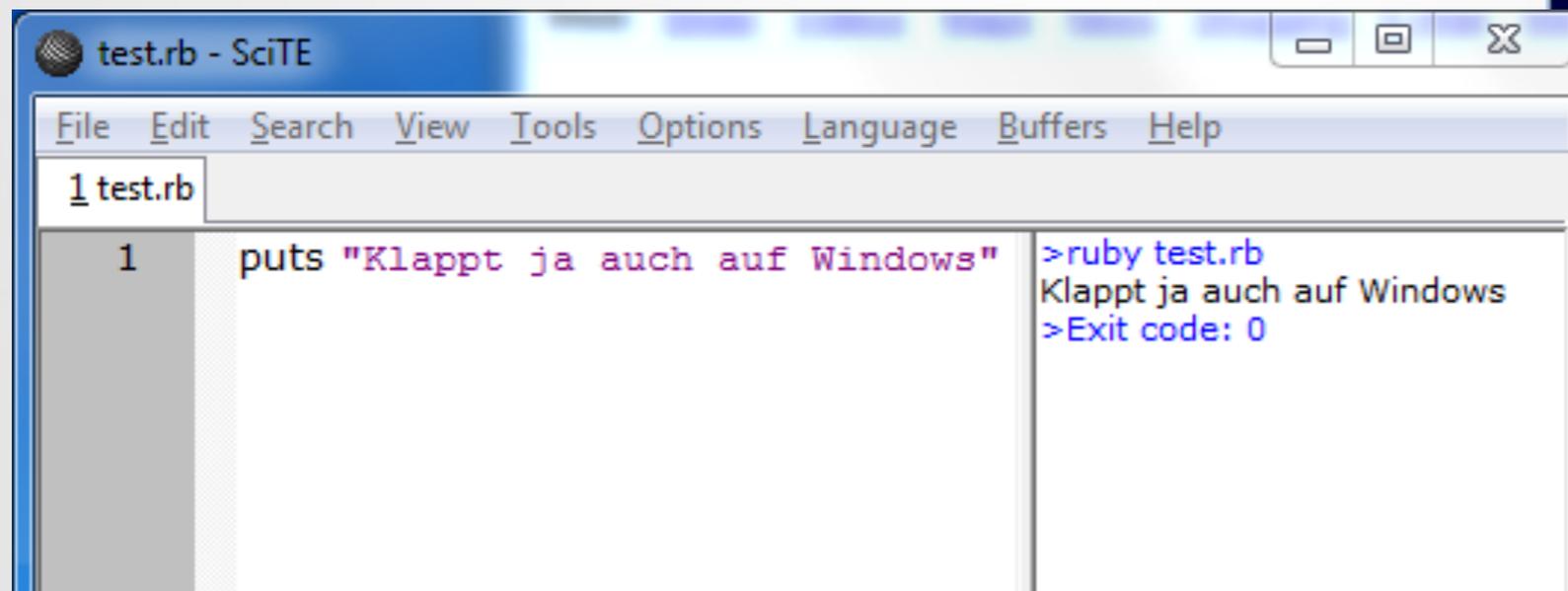


- Download unter
 - www.ruby-lang.org/de/
- installieren
- bringt Editor mit
- Scintilla text editor (SciTE)

Ruby unter Windows

Unter Windows gibt es verschiedene Möglichkeiten, Ruby zu installieren. Die erste besteht darin, einfach das kompilierte Binärpaket zu installieren. Die zweite darin, den One-Click-Installer zu verwenden. Wenn du dir nicht sicher bist, wie du Ruby installieren sollst, ist die zweite Möglichkeit wohl die beste Wahl für dich (neben Ruby bringt der One-Click-Installer noch ein paar zusätzliche Bibliotheken mit).

- [Ruby 1.8.6 One-Click Installer](#)
(md5: 69945dfae60224cc64139c91e000590e) Stabile Version
(empfohlen)
- [Ruby 1.8.7-p](#)





Python-Installation



auf dem Mac 

- auch installiert!
- vom Terminal aus starten mit „python“

```
schiller:~ Max$ python
Python 2.6.3 (r263:75184, Oct 2 2009, 07:56:03)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Python auf dem Mac"
Python auf dem Mac
>>> quit()
schiller:~ Max$
```



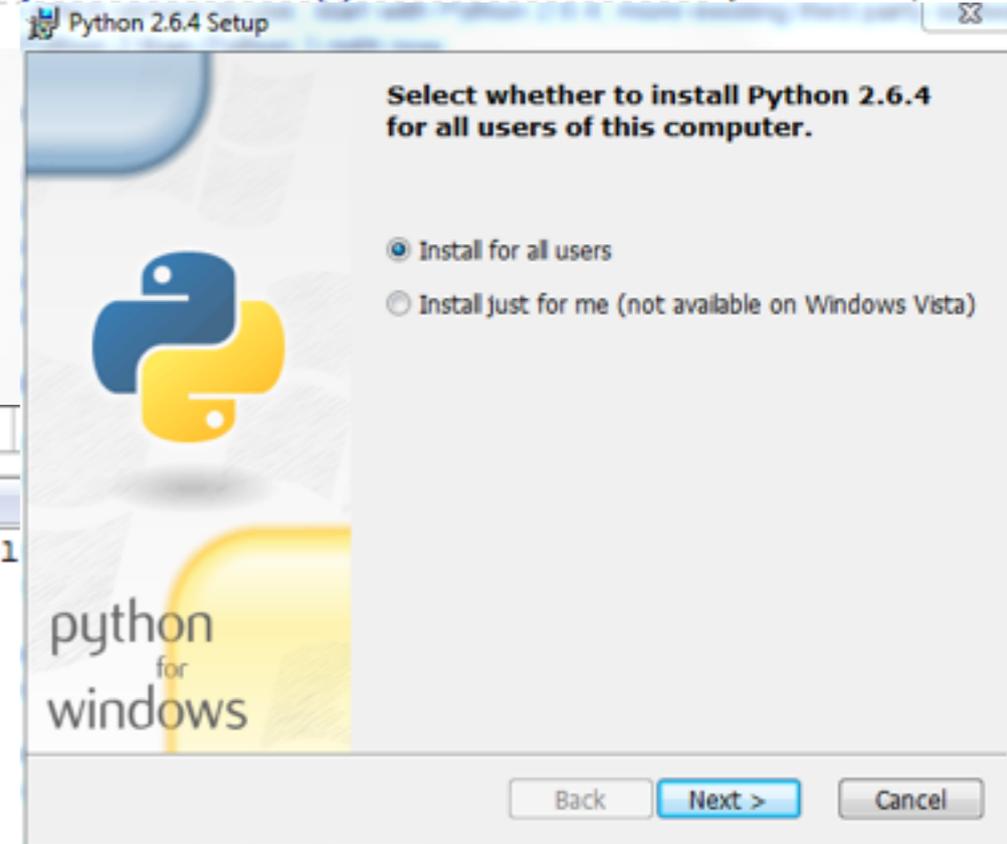
Python-Installation python™

unter  Windows

- Download unter
 - www.python.org
- installieren
- kein Editor in der Distribution

For the MD5 checksums and OpenPGP signatures, look

- [Python 2.6.4 Windows installer \(Windows binary -- d](#)
- [Python 2.6.4 Windows AMD64 installer \(Windows AM](#)
- [Python 2.6.4 Mac Installer Disk Image](#)
- [Python 2.6.4 compressed source tarball \(for Linux, U](#)
- [Python 2.6.4 bzipped source tarball \(for Linux, Unix o](#)



```
Python Shell
File Edit Debug Options Windows Help
Python 2.6.4 (r264:75708, Oct 26 2009, 08:23:19) [MSC v.1500 32 bit (Intel
win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 2.6.4      ==== No Subprocess ====
>>>
Python läuft!
>>>
```



PHP und Perl-Installation



- Interpreter kommen bereits mit XAMPP zusammen und werden installiert
- beides jedoch auch als Einzel-Installationen verfügbar
- Perl
 - unter Windows Strawberry Perl oder ActivePerl
- PHP
 - download unter www.php.net





Editoren



- Skripte sind lediglich Text-Dateien und können prinzipiell mit jedem Editor geschrieben werden
- Editoren bieten jedoch meist zusätzlich Features
 - (vor allem) Syntax-Highlighting
 - Code-Inspector
 - Refactoring
- Zwei Arten von Editoren
 - auf eine Sprache ausgerichtet (Entwicklungsumgebung)
 - unterstützt mehrere Sprachen



PSPad

unter  Windows



- kostenloser Windows-Editor
- mit vielen Features
 - FTP Upload
 - Syntax-Highlighting (Perl, PHP, Python)
 - (noch) kein Ruby
- www.pspad.org



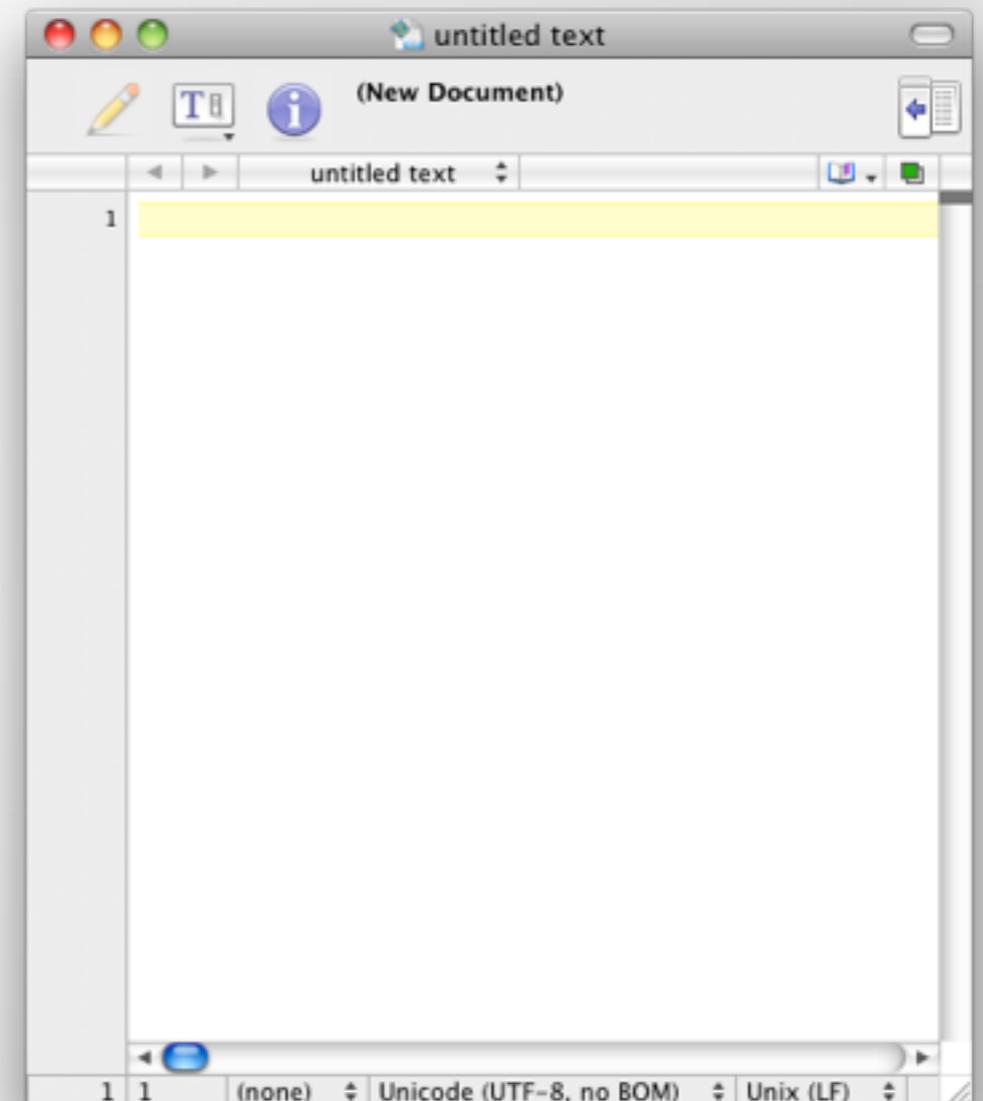


Textwrangler

auf dem Mac 



- kostenloser Mac-Editor
- ähnliche Features
 - FTP Upload
 - Syntax-Highlighting (Perl, Python, Ruby)
 - (anscheinend) kein PHP
- www.barebones.com





Entwicklungsumgebungen

- auch große Entwicklungsumgebungen lassen sich nutzen
- Plug-Ins für viele Sprachen
- Eclipse (ursprünglich Java-IDE)
 - z.B. Pydev
- Netbeans als Bundle für viele Sprachen erhältlich
 - Ruby, PHP
 - Python in einer early access Version





Das Erste Skript im Browser



Ein Python-Skript



Speicherort des Interpreters

```
#!/usr/bin/env python  
HEADER = "content-type: text/html\n\n"
```

```
def seite(text):  
    print HEADER  
    print "<html><body><h1>"+text+"</h1></body></html>"  
    quit()
```

Rückgabebetyp für den Browser

```
seite("Hallo")
```

Funktionsdefinition

Funktionsaufruf



Server falsch konfiguriert



- Wenn nicht schon vorkonfiguriert behandelt der Webserver jede Anfrage als Download-Aufforderung
- Achtung: Code sichtbar!!!

The screenshot shows a Mozilla Firefox browser window with the address bar set to `http://localhost/hallo.py`. The page content is a Python script, indicating that the web server is not configured to serve it as an HTML page. The script is as follows:

```
#!/usr/bin/env python
HEADER = "content-type: text/html\n\n"

def seite(text):
    print HEADER
    print "<html><body><h1>"+text+"</h1></body></html>"
    quit()

seite("Hallo")
```



Server konfigurieren



- Serverkonfiguration legt Dateitypen fest die ausgeführt werden sollen
- Verzeichnisbezogen (früher nur in cgi-bin)
- Bei Apache in der Konfigurationsdatei httpd.conf

```
<Directory "/Applications/XAMPP/xamppfiles/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks ExecCGI Includes
        AddHandler cgi-script .py

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride All

#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all

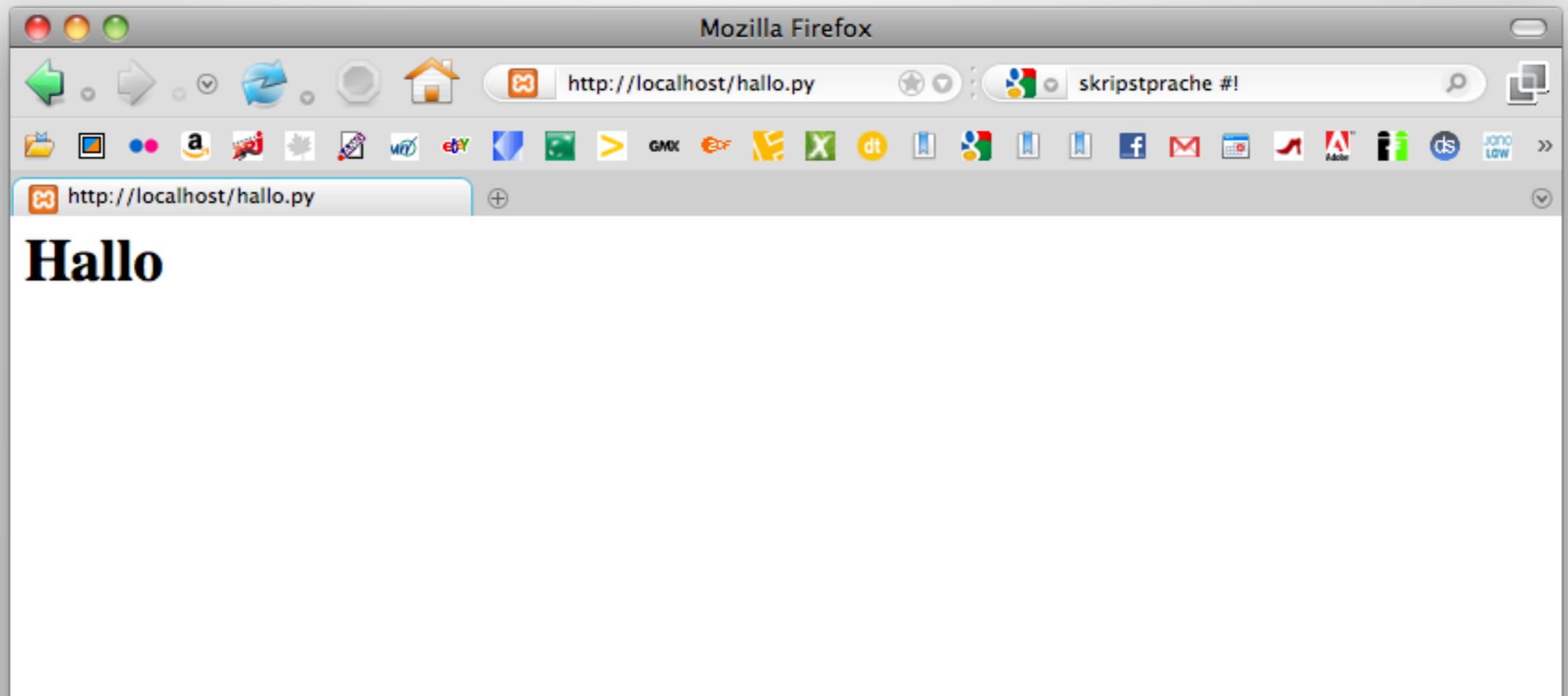
</Directory>
```



Server richtig konfiguriert



- Mit richtiger Konfiguration wird das Skript an den Interpreter aus der ersten Zeile geschickt
- Die Skriptausgabe wird als Antwort geschickt
- Achtung: Dateiberechtigungen (chmod 755)

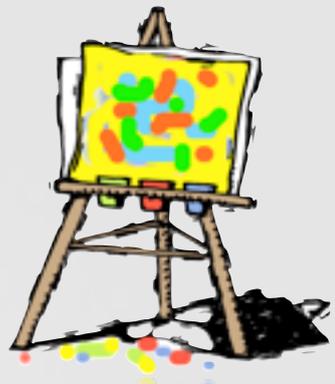




Nächste Woche:
Ruby



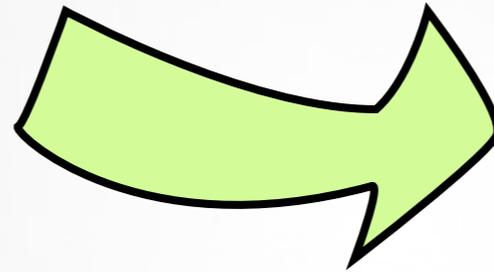
Galerie Baukasten



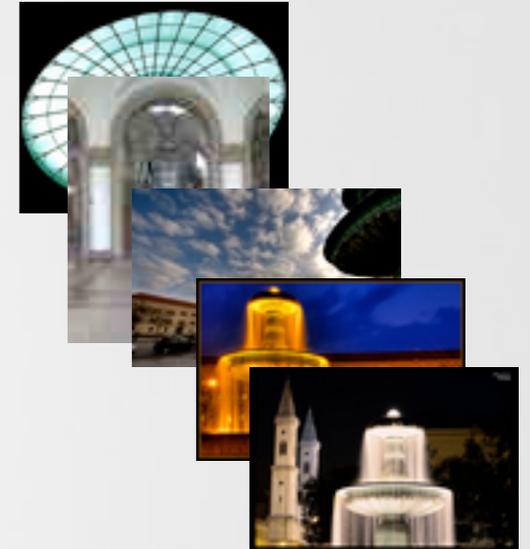
1. Formular anzeigen



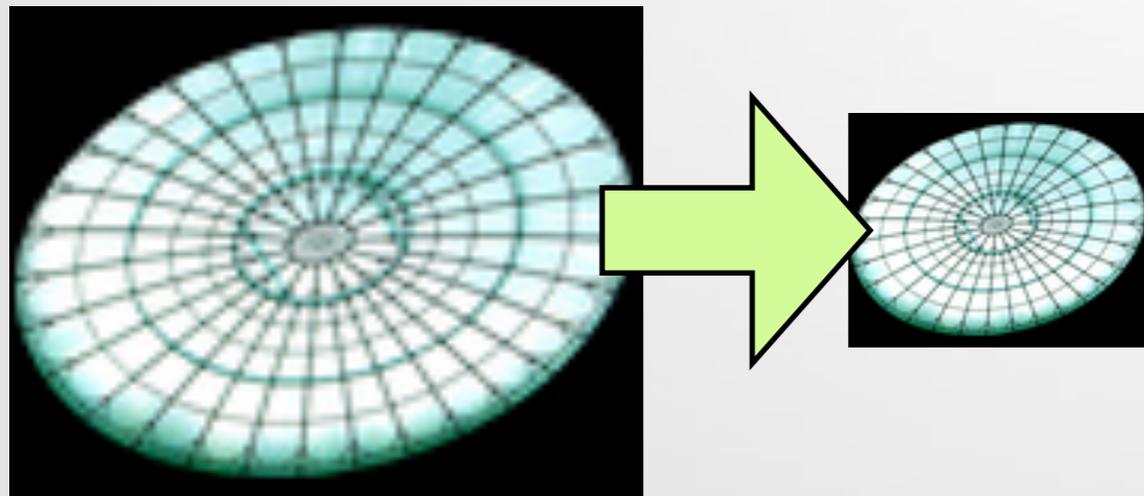
2. Zip-Datei hochladen



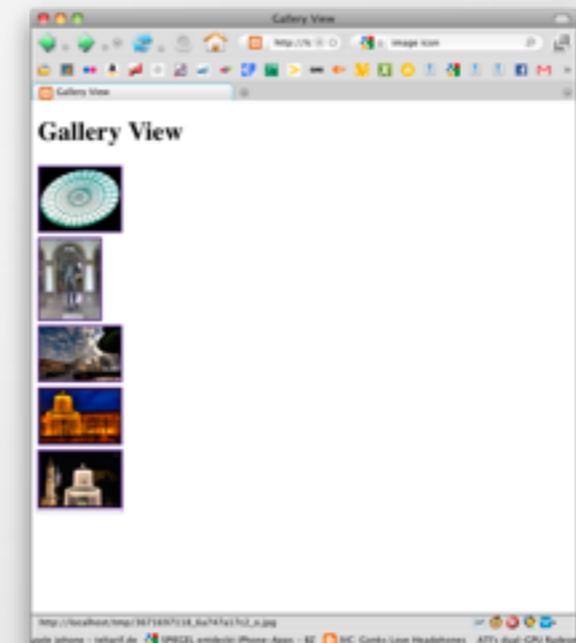
3. Entpacken



4. Thumbnails
rendern



5. Galerie Seite
anzeigen





Bildnachweis



- <http://www.hakstpoelten.ac.at/buchoase/Buch.gif>
- <http://school.discoveryeducation.com/clipart/images/arteasel4c.gif>
- http://de.wikipedia.org/w/index.php?title=Datei:Apple_Logo.svg&filetimestamp=20091016123338
- <http://www.iconarchive.com/icons/visualpharm/hardware/server-256x256.png>
- <http://upload.wikimedia.org/wikipedia/de/9/90/FirefoxLogo3.5.png>
- <http://www.swayze.to/images/notepad-trans.png>
- <http://upload.wikimedia.org/wikipedia/en/e/e8/TextWrangler.png>
- http://upload.wikimedia.org/wikipedia/en/c/c7/Pydev_logo.png
- <http://upload.wikimedia.org/wikipedia/en/2/29/NetBeansLogo.PNG>