



Triona – Information und Technologie GmbH

PHP vs. Java - ein Vergleich

Meik Forthofer, Joachim Schmidt, 28. November 2008

Einleitung

Das Internet bietet heute dank Web 2.0 mehr interaktive Webseiten und „echte Webanwendungen“ als jemals zuvor. Webseiten dienen schon lange nicht mehr der einfachen Darstellung von statischen oder dynamischen Daten, sondern werden immer mehr zur Konkurrenz von klassischen Desktop-Anwendungen.

Diesem Trend folgend wachsen die Technologien und Programmiersprachen für Webanwendungen stetig und mit großen Schritten.

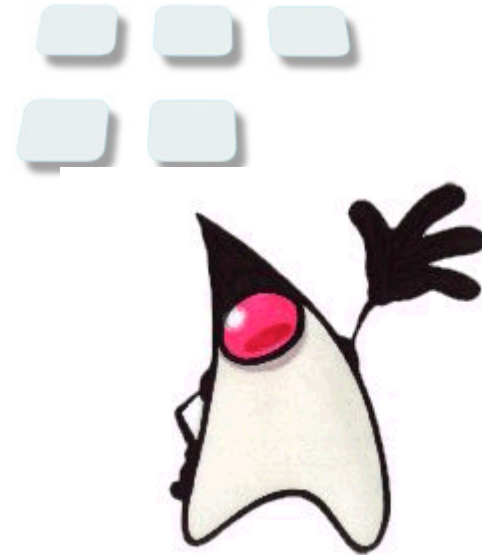
Zwei dieser Technologien wollen wir in diesem Workshop einmal gegenüber stellen und Unterschiede, Gemeinsamkeiten, Vorteile und Nachteile herausarbeiten.



Historie

Was ist Java?

- Java Technologie
 - Besteht aus:
- Java Programmiersprache
- Java Plattform
 - Java Runtime Environment (JRE)
 - Java Plattform, Standard Edition (Java SE)
 - Java Plattform, Micro Edition (Java ME)
 - Java Plattform, Enterprise Edition (Java EE)





Historie

Was ist Java?

• Java

- Urversion auch Oak (Object Application Kernel) genannt
- Entwickelt in 18 Monaten (zwischen 1991 und 1992) unter dem Projektnamen „The Green Project“ und am 23.05.1995 erstmals öffentlich vorgestellt
- Entwickelt von James Gosling und 12 weiteren Personen im Auftrag von Sun Microsystems
- Entwicklung einer kompletten Betriebsumgebung mit eigener virtueller CPU, die auf unterschiedlichsten Endgeräten laufen können soll
- Komplette objektorientiert, einige Konzepte von anderen Sprachen kopiert
- Wird in Bytecode übersetzt und läuft in virtuellen Maschinen



Historie

Was ist Java?

● Hauptziele

- Plattformunabhängige (Steuerungs-) Programme
- Steuerung von z.B. Kaffeemaschinen, Mediengeräten, etc.
- Neue Geschäftsfelder wurden gesucht und gefunden: Internet
- Popularität durch Integration in Netscape-Browser
- Version 1 primär zum Entwickeln von Applets gedacht
- Ab Version 1.1 Möglichkeiten für verteilte Programme und Persistenz
- Ab Version 1.4 besonders für Serverprogrammierung geeignet



Historie

...weitere Entwicklungsschritte

- Java 1.0 (1996)
 - Grundlegende Standardpackages vorhanden
 - Schwerpunkt: Applets
- Java 1.1 (1997)
 - Einige Erweiterungen der Syntax, z.B. innere Klassen und Listener
 - Viele neue Packages, u.A. JavaBeans, JAR Archive, RMI, JDBC, Serialisierung, Internationalisierung

TRIONA

INFORMATION UND TECHNOLOGIE

1991 1996 1998 2002 2004 2006



2002 2003 2004



Historie

...weitere Entwicklungsschritte

- Java 1.2 (1998)
 - Just-In-Time-Compiler (JIT) beschleunigt Ausführung erheblich
 - Umbenennung in „Java 2“ -> J2SE, J2EE, J2ME
 - Weitere neue Packages, u.A. Swing, CORBA, Audio, Drag&Drop
- Java 1.3 (2000)
 - HotSpot-Optimierung verbessert Performance erheblich
 - Wieder deutlich erweiterte API



Historie

...weitere Entwicklungsschritte

- Java 1.4 (2002)
 - Just-In-Time-Compiler (JIT) beschleunigt Ausführung erheblich
 - Umbenennung in „Java 2“ -> J2SE, J2EE, J2ME
 - Weitere neue Packages, u.A. Swing, CORBA, Audio, Drag&Drop



Historie

...weitere Entwicklungsschritte

- Java 5 (2004)
 - Umbenennung in „Java 5“ (Intern weiter Java 1.5)
 - Umbenennen von J2EE -> JEE usw.
 - Umfangreiche Erweiterung der Syntax und der API
 - Z.B. Generics, Annotations, vereinfachte Iterationen, Autoboxing, Enumerations



Historie

...weitere Entwicklungsschritte

- Java 6 (2006)
 - Verbesserungen in den Bereichen Diagnose, Überwachung und Management
 - Verbesserte Darstellung von Desktop-Anwendungen
 - Integration eigener relationaler Datenbank (Java-DB, basierend auf Apache-Derby)
 - Diverse Verbesserungen der JVM
 - Seit 2006 ist Java OpenSource (GPL 2)



Historie

Was ist PHP?

- Personal Home Page Tools (v1.0)

- 1995 von Rasmus Lerdorf entwickelt
- Prozedurale Open Source Skriptsprache
- Protokollierung der Web-Zugriffe auf seinen Online-Lebenslauf
- PHP-Kern mit erweiterter Funktionalität in C geschrieben
- veröffentlichter PHP/FI (FI stand für Form Interpreter) war Perl sehr ähnlich



TRIONA

INFORMATION UND TECHNOLOGIE

1995

1998

2000

2004

2008

2009



Historie

Was ist PHP?

• Hauptziele

- Erfassung der Web-Zugriffe
- leichte und schnelle Erstellung von dynamischen Webseiten und Webanwendungen
- Einfache Kommunikation mit Datenbanken (z.B. MySQL-Datenbank)

TRIONA

INFORMATION UND TECHNOLOGIE

1995 1998 2000 2004 2008 2009

2002 2003 2004



Historie

...weitere Entwicklungsschritte

● PHP 3

- Neuentwicklung von Andi Gutmans und Zeev Suraski
- PHP wird nun als „**Hypertext Preprocessor**“ bezeichnet
- Solide Infrastruktur und vielfältige Erweiterungsmöglichkeiten (Module)

● PHP 4

- Verbesserte Leistung von komplexen Applikationen
- Erste Unterstützung für objektorientierte Syntax
- Verbesserte Modularität des Basiscodes
- Optimierter PHP-Kern (**Zend Engine**)



Historie

PHP heute

- PHP 5
 - **Zend Engine 2** bietet robustes und vollständiges Objektmodell
 - Erfüllt Anforderungen an professionellen Einsatz
- Zend Technologies Ltd. bietet...
 - **Zend Studio:** komplette Entwicklungsumgebung
 - **Zend Framework:** zuverlässig, Komponenten-orientiert
 - **Zend Plattform:** hochverfügbarer PHP-Web Application-Server
 - ...

Historie

PHP Ausblick

● PHP 5.3

- Einführung von Namensräumen (namespace) und late static binding
- neuer, nativer MySQL-Treiber
- Oracle-Treiber mit Hochverfügbarkeitsfunktionalitäten (HA)

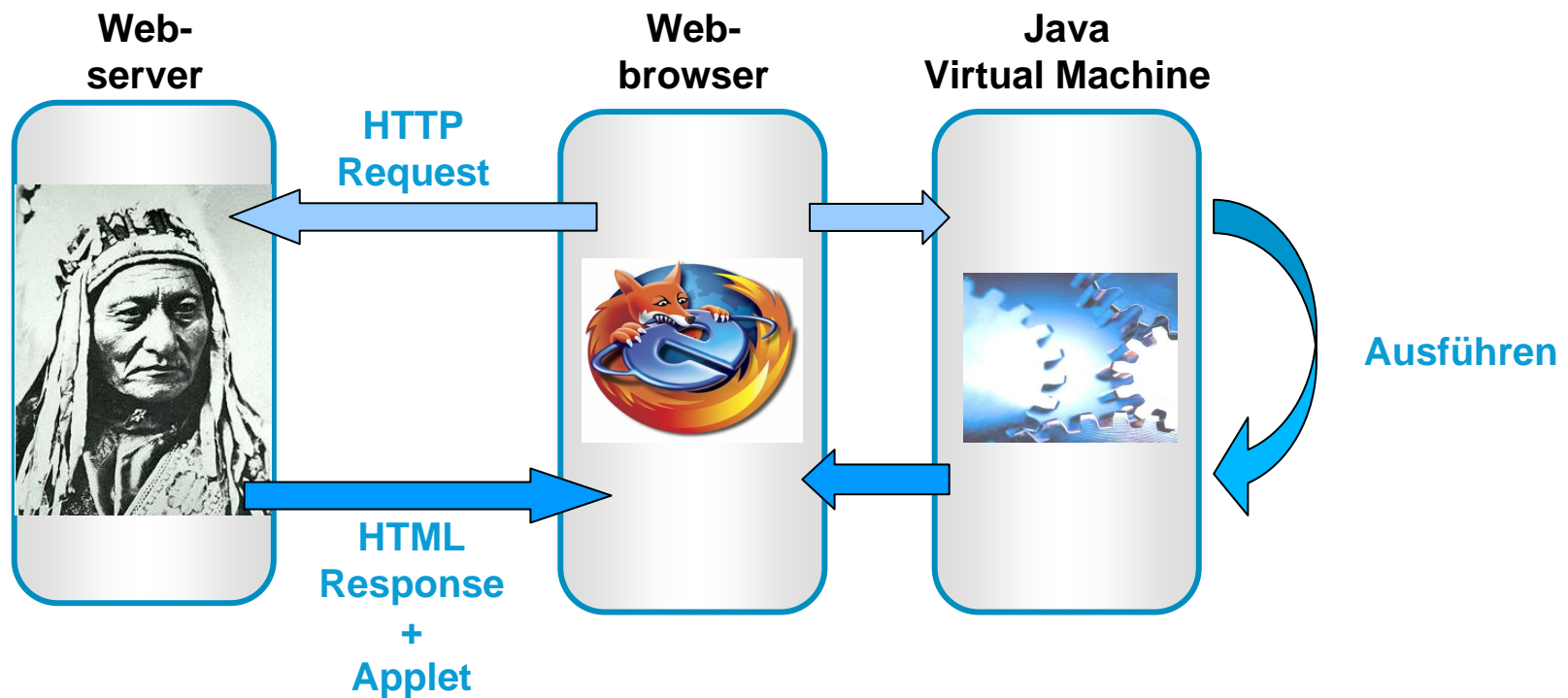
● PHP 6

- PHP 6 befindet sich noch in einer frühen Entwicklungsphase
- Entfernung von „Altlasten“ wie Register Globals, Magic Quotes, Safe Mode
- Unterstützung von Unicode
- neue Datumsroutinen für Zeiten vor 1970



Wie funktioniert Java?

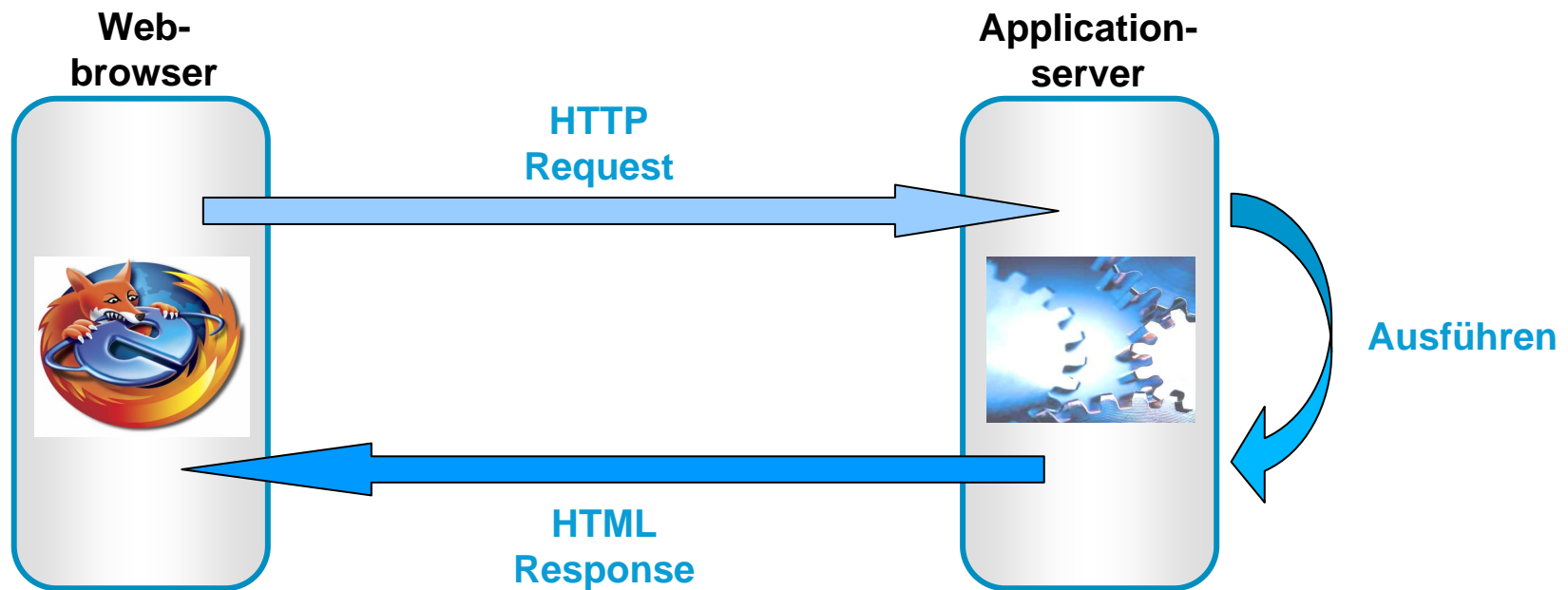
Als Applet clientseitig





Wie funktioniert Java?

Als Anwendung serverseitig





Begriffsdefinitionen

Spracheigenschaften

- Objektorientiert
 - Abstraktion, Strukturierung
 - Klassen und Objekte
 - Kapselung, Zugriffsbeschränkungen
 - Starrer Hierarchiebaum
 - Polymorphismus
 - Vererbung, Interfaces





Wie funktioniert Java?

Syntax - Variablen

- Notation

- Je nach Einsatzzweck unterschiedliche Möglichkeiten
- JSTL, JSF, JSP, ...

- Variablen

- `Typ name = new Typ();`
- `Integer 1`
- `Double 1.2345`
- `String "Hello World"`
- `Array array[3] = 1;`

- Explizite Typumwandlung

- `String wert = new String();`
- `Integer zahl = new Integer();`
- `wert = „3“;`
- `zahl = Integer.valueOf(wert);`



Wie funktioniert Java?

Syntax - Arrays

- `Typ name[] = new Typ[Anzahl];`
 - Schlüssel nur Integer
 - Wert wird explizit typisiert
 - Anzahl muss vorgegeben werden
- Erzeugung mit mehreren Elementen
 - `int array1[] = {2, 3, 5, 7, 11};`
- Zuweisung von Elementen
 - `array1[0] = 13;`



Wie funktioniert Java?

Syntax - Arrays

- Collections-API

- Interface List für numerische Arrays
- Interface Map für assoziative Arrays
- Interface Set für Liste von nicht doppelt vorkommenden Elementen

- ArrayList

- Beliebige Anzahl von Elementen
- Spezielle Methoden wie isEmpty(), indexOf() oder size() und Iteratoren
- Hinzufügen an beliebigen Stellen möglich



Wie funktioniert Java?

Hallo Welt!

Kurze Variante

```
<%  
    out.println(„Hallo Welt“);  
%>
```

HTML-konforme Variante

```
<html>  
  <head>  
    <title>PHP-Test</title>  
  </head>  
  <body>  
    <% out.println(„Hallo Welt“); %>  
  </body>  
</html>
```



Wie funktioniert Java?

Einfache Formularbehandlung

- Simpler Zugriff auf Inhalt der Variablen `name` und `state` mittels `request`-Objekt

```
<%  
if (request.getParameter(„name“) == null) { %>  
  <form action=„test.jsp“ method=„post“>  
    name: <input type=„text“ name=„name“/><br/>  
    state: <input type=„text“ name=„state“/><br/>  
    <input type=„submit“ value=„Abschicken“/>  
  </form>  
  
  <%  
  } else {  
    out.println(„Hallo“ +request.getParameter(„name“) + „ aus “  
      + request.getParameter(„state“) + „ !“;  
  }  
  %>
```



Wie funktioniert Java?

JSP, EL, JSTL, JSF, Beans, MVC

- Eine Trennung zwischen Layout und Logik muss in Java nicht umständlich über eine „Template-Engine“ erledigt werden.
- Java bietet mehr als eine vordefinierte Möglichkeit diese Trennung zu vollziehen.
- Eine Umsetzung des MVC-Patterns steht im Vordergrund
- Template-Engines zum dynamischen Ändern des Aussehens gibt es natürlich auch



Wie funktioniert Java?

Zusammenspiel Java -> JSTL, EL

JSP-Datei

```
<ul>

<c:forEach var=„marke“
items=„${bean.marken}“>
  <li>
  <c:out value=„${marke.name}“>
  </li>
</c:forEach>

</ul>
```



Backing Bean

```
Public class bean {

Private List marken;

Public List getMarken() {
  return marken;
}
}
```



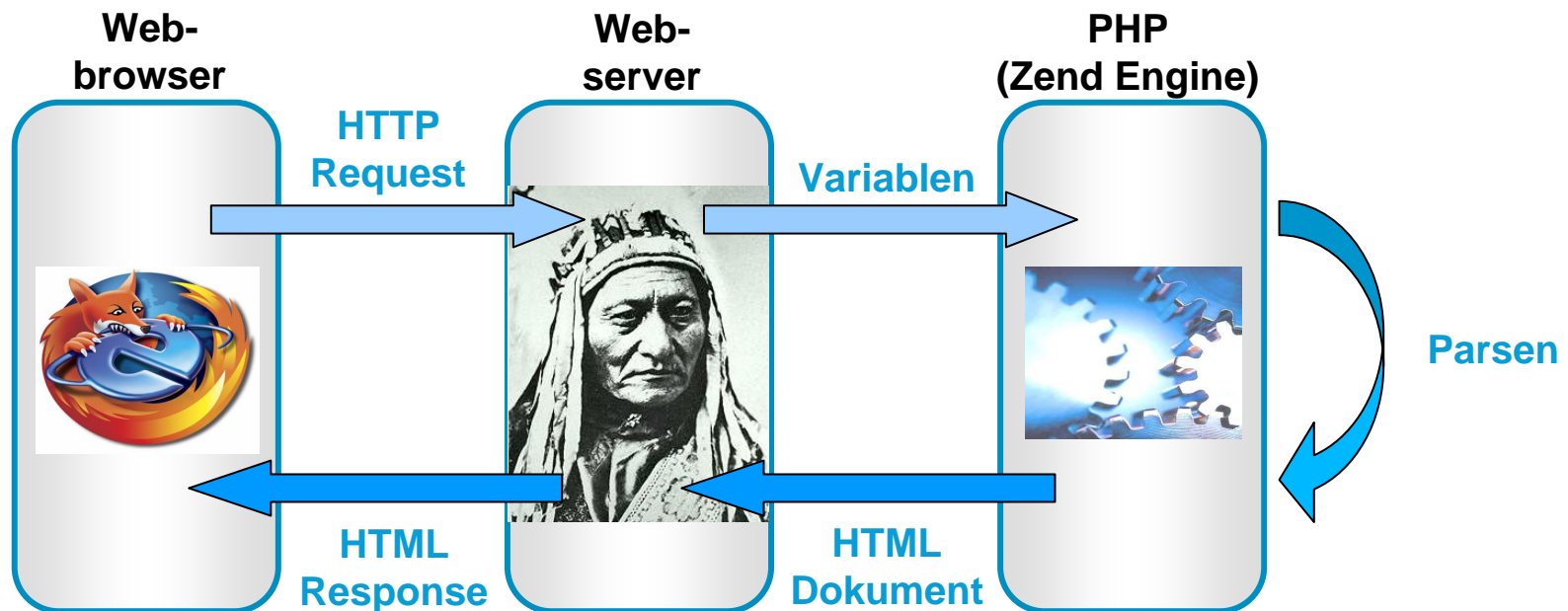
Wie funktioniert Java?

Vorteile

- Java 6
 - Flexible Struktur durch unterschiedliche Komponenten und Frameworks
 - Hoch skalierbar und performant für große Projekte
 - Strikte Trennung nach dem MVC Pattern
 - Verarbeitung auf optimierten Anwendungsservern
 - Wird vorkompiliert und zur Laufzeit nur noch ausgeführt
 - HighLevel Sprache, keine Scriptsprache
 - Für komplexe und Migrations-Projekte eher geeignet
 - Nicht nur für Webanwendungen

Wie funktioniert PHP?

Anwendung serverseitig

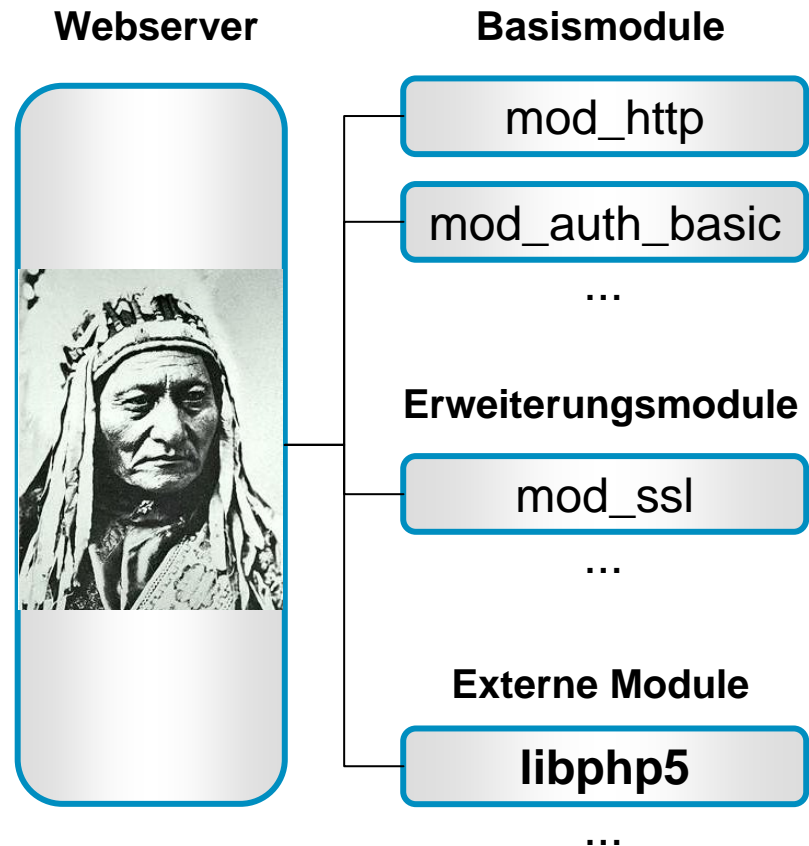


php Wie funktioniert PHP?

PHP als dynamisches Modul

• Modularität

- Apache-Kern für wichtigste Aufgaben
- Basismodule liegen kompiliert vor und bieten grundlegende Funktionalität
- Erweiterungsmodule können zur Laufzeit in Apache geladen werden (shared objects), Bestandteil des Apache-Softwarepakets
- Externe Module (z.B. PHP) nicht in Apache-Distribution enthalten (shared obj./statisch)
- **Vorteil:** Dynamische Module werden bei Nichtverwendung wieder entladen (Ressourcenfreigabe)



Begriffsdefinitionen Spracheigenschaften

- **Prozedural**
 - Ansatz: Programme bestehen aus kleineren Teilproblemen (Prozeduren)
 - Die Anweisung ist der kleinste Schritt (atomar)
 - Programm schreitet von Anweisung zu Anweisung (lat. *procedere*)
- **Skriptsprache**
 - Implizit deklarierte Variablen
 - Dynamische, automatische Typumwandlung
 - Datentypen werden zur Laufzeit festgelegt

Wie funktioniert PHP?

Syntax - Variablen

● Notation

- `<?php` Anweisung an den Parser zu Beginn jeder PHP-Datei, ab jetzt zu interpretieren
- `?>` Ende des Parsens

● Variablen

- `$varname`
- Integer `1`
- Double `1.2345`
- String `"Hello World"`
- Array `$array["inhalt"] = 1`

● Automatisches Typecasting

- `$a = 1 // Integer`
- `$b = "2" // String`
- `$c = $a + $b // $c == 3`
- `$d = $a . $b // $d == "12"`

Wie funktioniert PHP?

Syntax - Arrays

- **array([Schlüssel =>] Wert , ...)**
 - Schlüssel kann entweder Integer oder String sein
 - Wert kann beliebigen Typ besitzen
 - Numerische oder assoziative Arrays verwendbar
- **Erzeugung mit mehreren Elementen**
 - `$array1 = array(2, 3, 5, 7, 11); // numerisch`
 - `$array2 = array("one"=>1, "two"=>2); // assoziativ`
- **Zuweisung von Elementen**
 - `$array1[0] = "13";`
 - `$array2["one"] = 1;`

Wie funktioniert PHP?

Hallo Welt!

Kurze Variante

```
<?php  
    echo 'Hallo Welt';  
?>
```

HTML-konforme Variante

```
<html>  
  <head>  
    <title>PHP-Test</title>  
  </head>  
  <body>  
    <?php echo 'Hallo Welt'; ?>  
  </body>  
</html>
```

Wie funktioniert PHP? Einfache Formularbehandlung

- Simpler Zugriff auf Inhalt der Variablen `name` und `state` mittels `$_POST`/`$_GET`

```
<?php
if (!isset($_POST['name'])) { ?>
    <form action="test.php" method="post">
        name: <input type="text" name="name"/><br/>
        state: <input type="text" name="state"/><br/>
        <input type="submit" value="Abschicken"/>
    </form>

    <?php
    } else {
        echo "Hallo ".$_POST['name']. " aus ".$_POST['state']. " !";
    }
?>
```

Wie funktioniert PHP?

Smarty, die Template Engine

- **Template Engine**
 - Unterstützt den Webdesigner, um typische Anforderungen an die Präsentationsschicht schnell und effektiv zu lösen
 - Designziel in Anlehnung an MVC: Trennung von Businesslogik und Präsentationscode
 - Grad der Trennung je nach Anforderung flexibel umsetzbar
 - Features: Caching, Modifiers, Filters, Plugins
 - Tag-basierte Syntax

Wie funktioniert PHP?

Zusammenspiel PHP -> Smarty

PHP-Datei *start.php* ruft...

```
<?php
// Code
// ...

/**
 * Smarty Ausgabe
 */
$smarty->assign('$HTML_TITLE', 'Workshop');
$smarty->display('start.tpl');
?>
```

Template-Datei *start.tpl* auf!

```
<html>
<head>
<!-- SMARTY -->
{if !empty($HTML_TITLE)}
    <title>{$HTML_TITLE}</title>
{/if}
</head>
<body>
<...>
```

Wie funktioniert PHP?

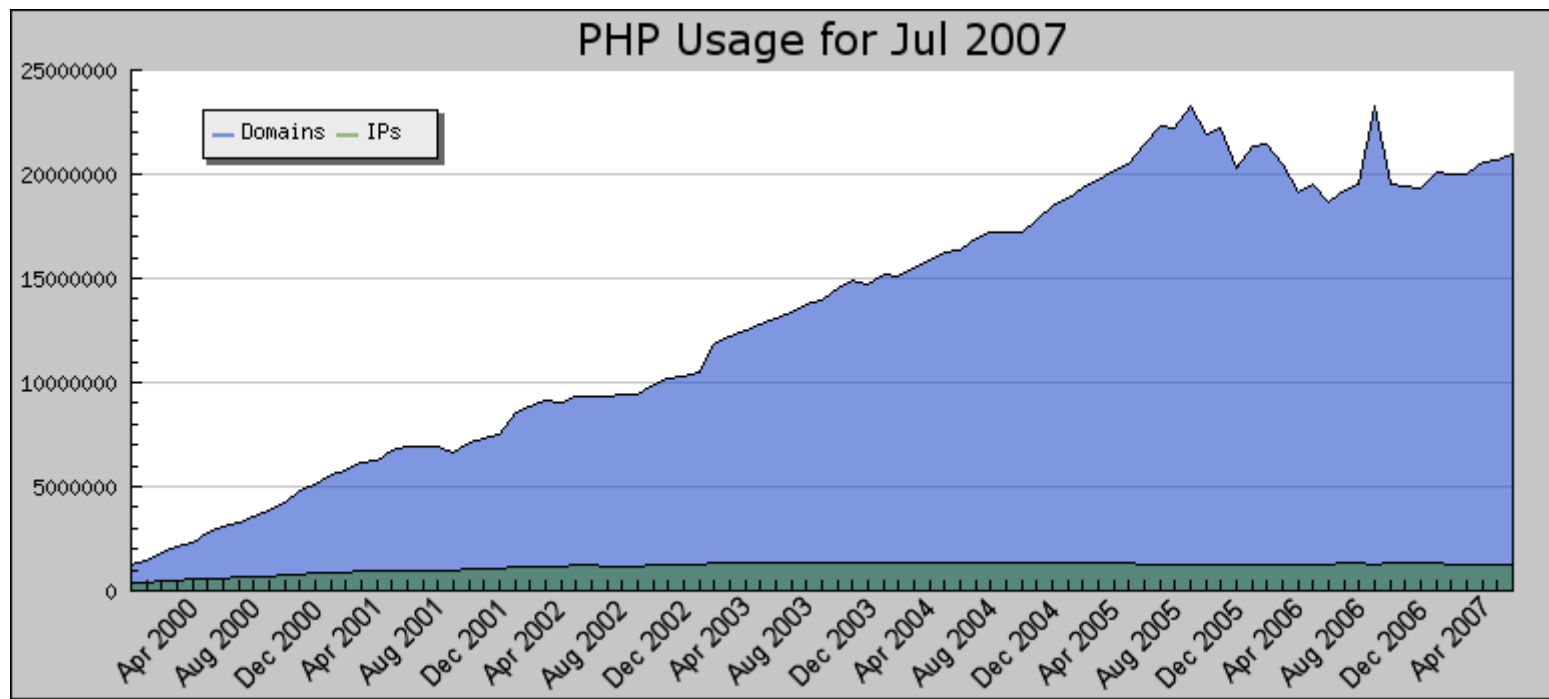
Vorteile

- PHP 5
 - optimal kombinierbar mit HTML durch direkte Integration
 - Code wird zur Laufzeit übersetzt, keine Kompilierung notwendig
 - viele vordefinierte, optimierte Funktionen
 - einfache Datenbank-Anbindung (auch persistent)
 - Performante Abarbeitung bei Installation als Apache-Modul
 - Kombination von prozeduraler und objektorientierter Programmierung

Vergleich

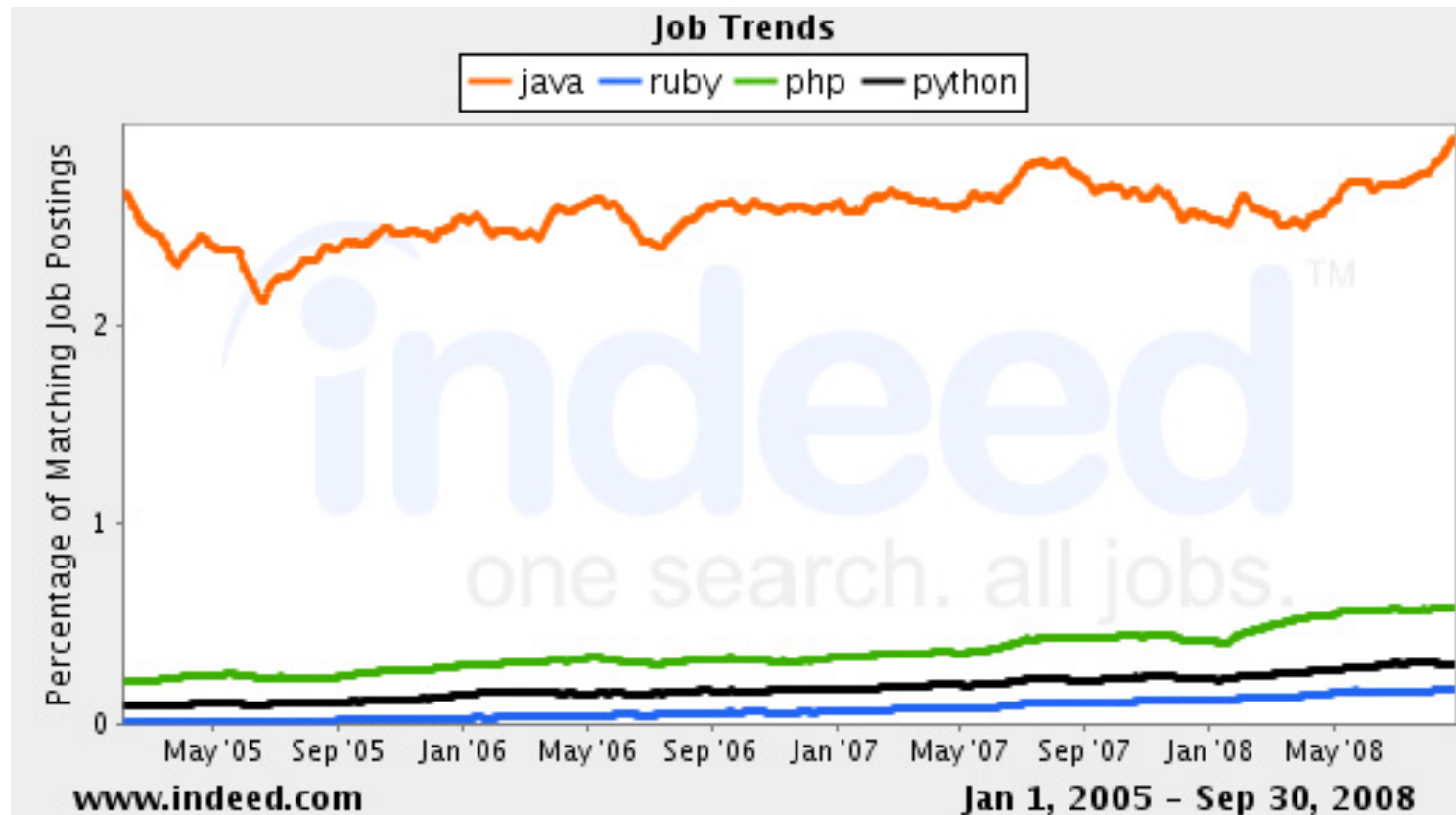
Popularität PHP

- Juli 2007: 20,917,850 domains, 1,224,183 IP Adressen



Vergleich

Jobs im Umfeld, PHP und Java



Vergleich

Popularität

- TIOBE Programming Community Index
 - Indikator für die Popularität einer Programmiersprache
 - Auswertung nur anhand der Trefferquote der zu suchenden Sprache über populäre Suchmaschinen

Position Nov 2008	Position Nov 2007	Delta in Position	Programming Language	Ratings Nov 2008	Delta Nov 2007	Status
1	1	=	Java	20.299%	-0.24%	A
2	2	=	C	15.276%	+1.31%	A
3	4	↑	C++	10.357%	+1.61%	A
4	3	↓	(Visual) Basic	9.270%	-0.96%	A
5	5	=	PHP	8.940%	+0.25%	A



Wie funktioniert PHP?

Weitere Fragen?



Wie funktioniert Java?

Weitere Fragen?

