Flash Tutoriat SS 2007

ActionScript OOP /Flash Dynamisch
XML / PHP & MySQL / Remoting

Von Patrick Heneise und Florian Weil

Objektorientierte Programmierung

- Warum OOP in Flash?
  - Quelltexte werden überschaubarer
  - Änderungen können einfacher vorgenommen werden / Bessere Wartung
  - Dynamische Erzeugung von Objekten zur Laufzeit
  - Klasse können für neue Projekte wieder verwendet werden!!!

OOP in Flash - Syntax

- Importieren von Klassen
  - Mit import flash.display.BitmapData;
  - Eigene Klassen liegen direkt im Verzeichnis der .Fla Datei
  - Extra Klassenstruktur für eigene Klassen (erweitern des Classpath unter Einstellungen)
  - Bsp.: import ordnername.Klassename;
- Verwalten und Import von Klassenpaketen
  - import flash.display.* -> ganzes Paket
  - Ordnerstruktur aufbauen -> besser Übersicht:
    - Land.Domain.Paketname
    - Bsp: de.derhess.gui

Eigene Klasse schreiben

- Dateiname muss Klassename sein mit Dateiendung .as
  - Bsp. Bild.as
- Klassenstruktur

```
class Bild {
        public var attribute:Number;

        function Bild(parameter:Datentyp) {
                this.attribute = 5;
        }

        public function macheWas() {
        }
}
```

Attribute und Zugriffe

- Public
  - Zugriff auf Funktionen und Attribute ausserhalb der Klasse
  - Bsp. public var variablennamen:Typ;
  - Bsp. public function funktionsname():RückgabeTyp;
- Private
  - Zugriff auf Funktionen und Attribute nur innerhalb der Klassen
  - Bsp. private var variablennamen:Typ;
  - Bsp. private function funktionsname():Void;
- Static
  - Zugriff nur aus der Klasse direkt heraus
  - Bsp. static public function funktionsname():RückgabeTyp;
  - Bsp. static public var variablenname:Typ;
  - Bsp. Klassenname.funktionsname(parameter:Typ);

(Mehrfach) Vererbung

- Einfache Vererbung
  - Mit extends im Klassenkopf
  - Bsp. class Bild extends MovieClip { … }
  - Mit der Funktion super() kann der Konstruktor der Mutterklasse aufgerufen werden (Beispiel an der Tafel bei Bedarf)
- Mehrfachvererbung über Interfaces
  - Dateiname Produkt.as und im Quellcode:
    - interface Produkt { public function tuwas(); }
    - Schreibwaren Klasse als Beispiel
    - class Schreibwaren implements Produkt { … }

Mehrfach Vererbung

- Extrem Beispiele
  - class AbKlasse extends Basisklasse implements Schnittstelle
  - class Basisklasse implements Schnittstelle1, Schnittstelle2

- Wird verwendet bei Erstellung von komplizierten Anwendungen und Frameworks
- Anwendung bei Design Patterns

- Mehr Infos zu Objektorientierter Programmierung mit Flash in dem Buch “Object-Oriented ActionScript For Flash 8“ von Friends of ED

Eventhandling

- Eigene Events werfen
  - mx.utils.Delegate Class (http://www.adobe.com/devnet/flash/articles/eventproxy.html und http://www.galileodesign.de/openbook/actionscript/actionscript_6_26_001.htm)
  - mx.events.EventDispatcher (http://www.asnative.de/artikel/eventdispatcher.html und http://www.adobe.com/devnet/flash/articles/creating_events.html)
- Einsatz bei Komponentenentwicklung, Frameworks und Rich Internet Application Entwicklung
- EventDispatcher benutzen
  - Komfortabler, soll besser sein
  - Mag ich mehr (meine subjektive Meinung)

EventDispatcher

```
// Meldeamt.as import mx.events.EventDispatcher;
class Meldeamt {
	private var city:String;
	private var dispatchEvent:Function;
	public var addEventListener:Function;
	public var removeEventListener:Function; public

	function Meldeamt(city:String) {
		EventDispatcher.initialize(this);
		this.city = city;
	}

	public function sendeBescheid():Void {
		this.dispatchEvent({type:"onBescheid", target:this});
	}

	public function getCity():String { return this.city; }
}
```

Quelle: www.asnative.de

EventDispatcher Teil 2

- Listener Erstellen
  - var einwohner:Object = new Object();
  - einwohner.onBescheid = function(ereignisObjekt) {
    trace("Brief vom Meldeamt " + e.target.getCity());
    };
- Anwendung:
  - var amt:Meldeamt = new Meldeamt("Berlin");
    amt.addEventListener("onBescheid", einwohner);
    amt.sendeBescheid();
- Trace Ausgabe:
  - Brief vom Meldeamt Berlin

Quelle: www.asnative.de

EventDispatcher Teil 3

- Benutzerdefinierte Variablen hinzufügen
  - this.dispatchEvent({type:"onBegruessung", target:this, betrag:110.00});
  - Listener.onBegruessung = function (eventObjekt) {
    trace(„Betrag: “ + eventObjekt.betrag);
    }

- Saubere Lösung ist die Erstellung eines EventObjekts
  - class EventBescheid {
    public var type:String;
    public var target:Object;

    public function EventBescheid(target:Object) {
    this.type = "onBescheid"; this.target = target; }
    }
  - public function sendeBescheid():Void {
    this.dispatchEvent(new EventBescheid(this));
    }

Quelle: www.asnative.de

Geschafft !!!

- Yeah, der objektorientierte Part ist fertig!!

Flash und XML

- Flash und XML ist gut für:
  - dynamische Menüleiste
  - einfache mehrsprachige Websites
  - Bildergalerien
  - Einfache News Systeme

- Allgemein: für dynamische Inhalte mit mittlerer Menge
- Ziemlich hoher Datenoverhead

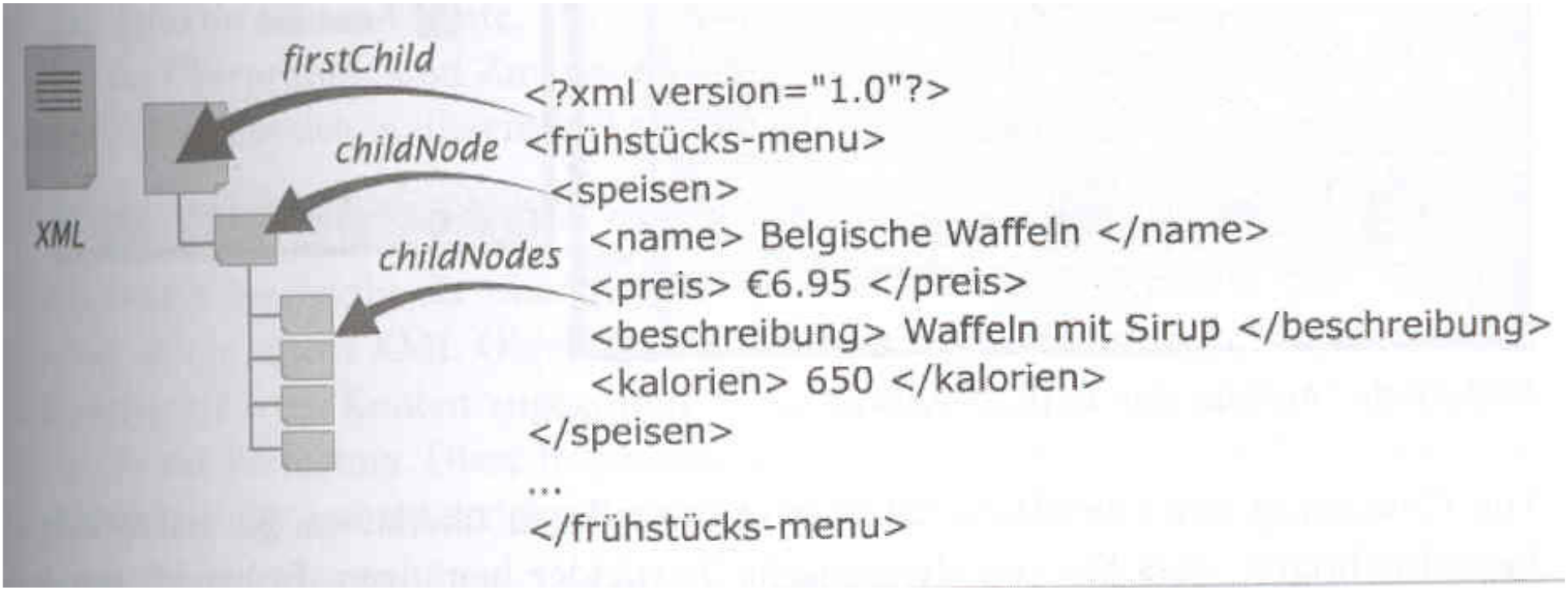
Grundlagen XML

- XML ist eine Markup-Language.
- Grundidee von XML: logischer Aufbau von Datenstrukturen (in Objektform)

- Grundaufbau eines XMLs:

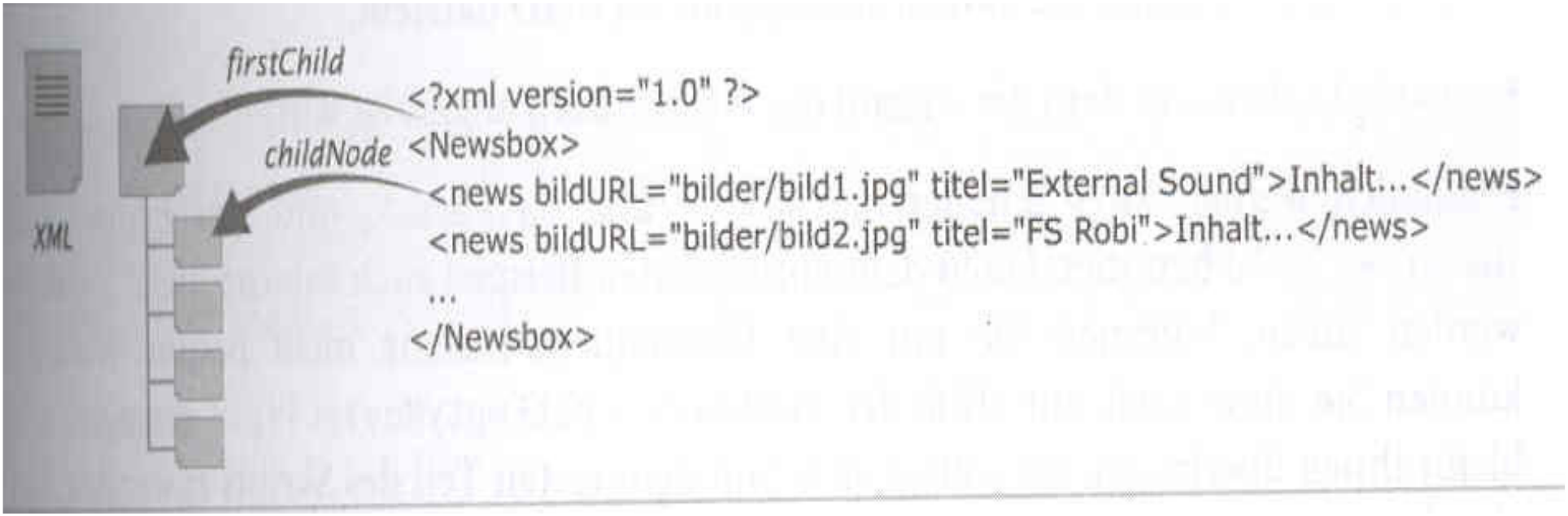
```
<?xml version="1.0" ?>

<news>
        <eintrag>News 1</eintrag>
        <eintrag>News 2</eintrag>
        <eintrag>News 3</eintrag>
        ...
        <eintrag>News n</eintrag>
</news>
```

Grundlagen XML



Quelle: ActionScript Praxisbuch von Matthias und Caroline Kannengiesser

Grundlagen XML: Attribute



Quelle: ActionScript Praxisbuch von Matthias und Caroline Kannengiesser

Grundlegendes in Flash

- Für XML Verarbeitung benötigt man:
  - Die XML Klasse (erbt vom XML Node)
  - XML-Node Klasse
- XML Objekt
  - Für das Laden einer XML Datei
  - Neue Tags erzeugen
- XML Node
  - Zuständig für die Struktur (anhängen, löschen)
  - Zum navigieren zwischen den verschiedenen Tags

Navigieren durch das XML mittels ActionScript

- Navigieren durch das XML
  - childNodes
  - parentNode
  - previousSibling
  - nextSibling
- Zugriff auf die Daten
  - nodeValue -> klappt nur richtig bei Textknoten
  - nodeName -> klappt nicht bei Textknoten
- Attribute
  - xmlobj.xmlnode.attributes.attributname Zugriff auf die Attribute

Flash liest XML - Beispielcode

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<design>
        <foto id="1">
                <dateiname>demo/foto/arbeitsamt.jpg</dateiname>
                <ort>Frankfurt</ort>
                <zeit>Fruehling 2006</zeit>
                <beschreibung>Der Arbeitsamt Bau in Frankfurt. Sicherheit, Sauberkeit und Ordnung ist hier das oberste Gebot</beschreibung>
        </foto>

        <foto id="2">
                <dateiname>demo/foto/bungalow.jpg</dateiname>
                <ort>Frankfurt</ort>
                <zeit>Fruehling 2006</zeit>
                <beschreibung>Ein neues Mietshaus am Frankfurter Mainhafen in der Nahaufnahme</beschreibung>
        </foto>
</design>
```

Flash liest XML - Beispielcode

```
var datenXML:XML = new XML();
datenXML.ignoreWhite = true;

var wurzel:XMLNode = new XMLNode();
datenXML.load(aktivesXML);

datenXML.onLoad = function(success) {
wurzel = datenXML.firstChild;      // Gallerie Knoten
// Lese alle Fotos aus
for(var i:Number = 0; i < wurzel.childNodes.length; i++) {
    foto = wurzel.childNodes[i];          // Foto Knoten
    fotos_array[i] = new scrollerItem(foto.childNodes[0].firstChild.nodeVale,
                                      foto.childNodes[1].firstChild.nodeValue,
                                      foto.childNodes[2].firstChild.nodeValue,
                                      foto.childNodes[3].firstChild.nodeValue);
    }
};
```

XML-Bearbeitung mit Flash

- Erstellung von Tags
  - xmlObj.createElement("elementname")
  - xmlObj.createTexteNode("Inhalt")
- Einfügen von Tags
  - xmlNode.appendChild(XMLnode)
  - xmlNode.insertBefore();
- Löschen von Tags
  - xmlNode.removeNode();

Anmerkung für die XML-Verarbeitung in Flash

objXML = new XML();
objXML.ignoreWhite = true;
und die XML-Datei in UTF8 codieren, wegen den Umlauten

- Zum Speichern der (veränderten) XML-Struktur in eine externe Datei
  - objXML.contentType = "text/xml"; sicherheitshalber setzen
  - LoadVars Objekt erzeugen
  - XML-Struktur in einen String umwandeln -> XmlNode.toString()
  - mit LoadVars.sendAnLoad() an ein PHP Skript schicken
  - In PHP mittels fopen() und fwrite() die Daten speichern

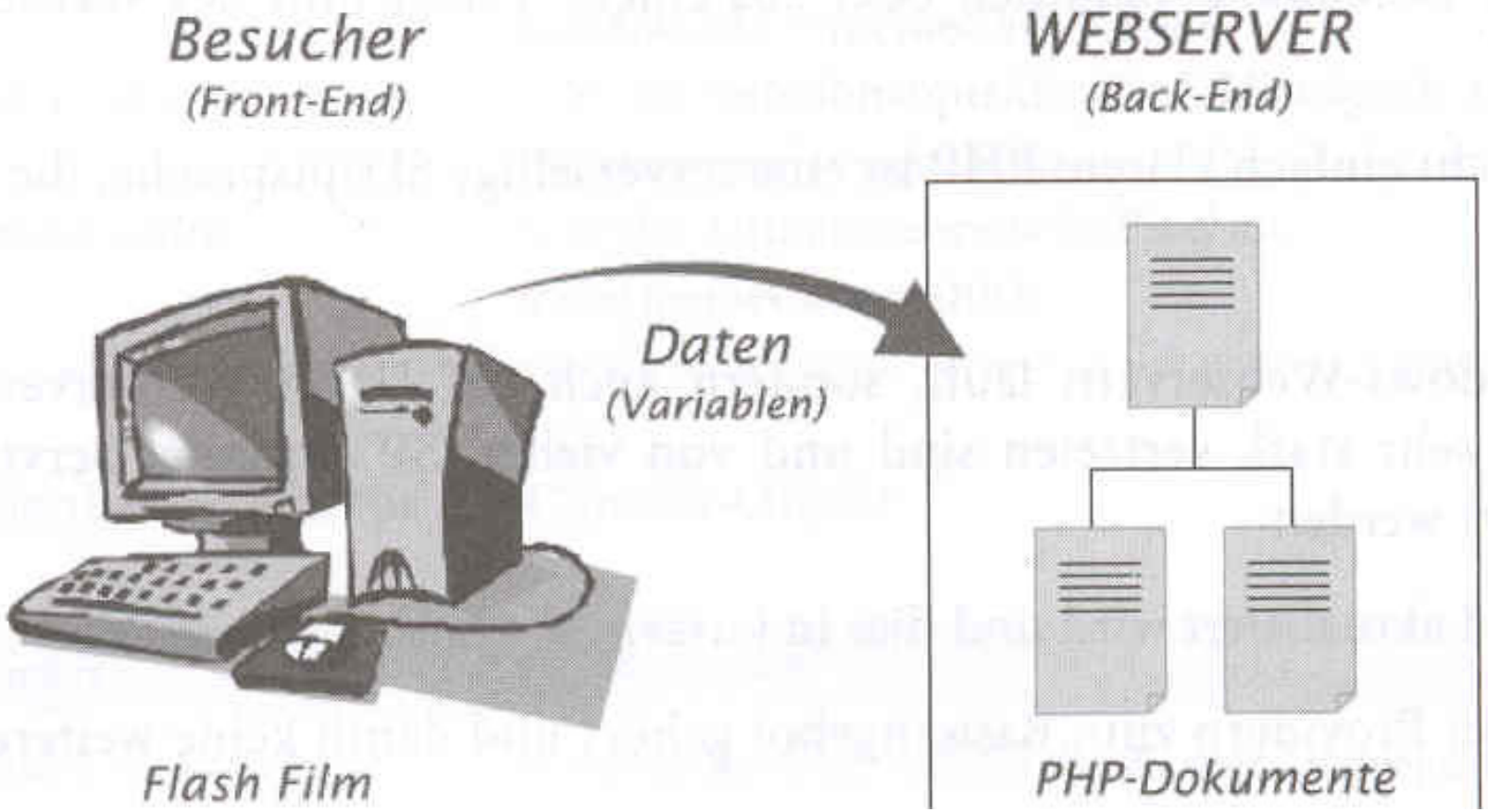
- Diese Lösung ist besser als die xmlObj.send() Methode, wegen einer besseren Kontrolle über die Speicherung der Daten.

Abschluss Flash und XML

Yeah!!!

Erster Part geschafft!

Flash Kommunikation mit PHP



Quelle: ActionScript Praxisbuch von Matthias und Caroline Kannengiesser

Flash Kommunikation mit PHP

- Flash Kommunikation mit PHP
- Grundlegendes:
  - Flash ruft eine Serverseitiges Skript auf (hier: PHP)
- mit Hilfe einer Serverskript Sprache kann man:
  - auf Datenbanken zugreifen und bearbeiten
  - Dateien bearbeiten und schreiben
  - Stark dynamische Inhalte abrufen (Highscoreliste)
- Der Datenaustausch erfolgt immer über HTTP GET oder POST Methoden
  - immer POST Methode verwenden (weil unbegrenzte Anzahl Zeichen, erschwertes URL-Rewriting)

Exkurs: PHP Grundlagen

- <?php .... ?> Code Anfang von PHP
- vor jede variable ein $ -> $variale
- Strings in '...' und mit '.' verknüpfen -> 'Hallo '.'Welt!';
- Zugriff auf Objekte $this->attribut;
- Text Ausgabe mit PHP durch echo('String: '.variable);
- Top Nachschlagewerk für PHP unter http://de3.php.net/download-docs.php

PHP & MySQL Grundlagen

- Prinzip:
  - $dbHost = 'localhost'; // Host der DB
  - $dbUser = 'root'; // DB-Benutzer
  - $dbPasswort = ''; // DB-Passwort
  - $dbDBName = 'testbank'; // Name der DB
  - $dbLink = 0;

- Conncecten mit der Datenbank
  - $dbLink = mysql_connect($dbHost,$dbUser,$dbPasswort);

- Datenbank auswählen
  - mysql_select_db($dbDBName,$dbLink); // Gibt ein Boolean Wert zurück

PHP & MySQL - Anfrage

// SQL Code schreiben

- $query = 'SELECT * FROM `blog` ORDER BY `datum` DESC';

// Anfrage zur Datenbankschicken

- $resultarray = mysql_query($query,$dbLink);
- //speichert ein Ergebnis ab

// Ergebnis verarbeiten

- while ($data = mysql_fetch_assoc($resultarray)) {
- echo 'Spalte1: '.$data['spaltenname']....;
- }

// Datenbankverbindung schliessen (Perfomance)

- mysql_close($dbLink);

PHP & MySQL - Besser mit Klassen Arbeiten

```
//importieren Datenbankklasse
include_once('php/db.class.php');

$database = new db();
$database->dbConnect();

$query = 'SELECT * FROM `blog` ORDER BY `datum` DESC';
$database->dbquery($query);

// Verarbeitung Datenbankanfrage
while ($data = mysql_fetch_assoc($database->resultarray)) {
    echo 'Spalte1: '.$data['spaltenname']....;
}

$database->dbClose();
```

Flash Kommunikation mit PHP

- ActionScript 1-Methode
  - loadVariablesNum("phpurl", 0, "POST");
    - phpurl:String -> Pfad des PHP Skriptes ("phpurl.php?cache=" + (new Date().getTime()) verhindert Browser Cache)
    - 0:Number -> Levelzahl (0=Hauptzeitleiste) an diesem Ort werden die Variablen gespeichert
    - POST:String -> Übermittlungs HTTP-Methode
- ActionScrip 2-Mehtode:
  - LoadVars-Objekt
  - besser weil:
    - komfortableres versenden und empfangen von Daten
    - Bessere Variablenverwaltung
    - Eventhandling für den Datentransfer
- Generell guter Link zum Thema Flash und PHP:
  - http://flashhilfe.de/forum/tipps-tricks/php-daten-in-flash-laden-loadvars-object-etc-33971-33971.html

Die LoadVars Klasse

var lv = LoadVars()

- Variablen hinzufügen:
  - lv.variable1 = "Hallo";
  - lv.variable2 = "Welt";

- Variablen versenden und empfangen:
  - lv.send("phpurl","_self","POST"):Boolean;
  - lv.load("phpurl"):Boolean;
  - lv.sendAndLoad("phpurl",ziel_lv,"POST");

- Ereignisse
  - lv.onLoad = function (sucess) { trace(sucess); }
  - lv.onHTTPStatus = function(httpStatus:Number) { trace(httpStatus); }

Datenverarbeitung auf der PHP Seite

Zugriff auf die Daten über die Globale Variable

- $_POST['variable1'] oder $_GET['variable1']

- Ausgabe wieder an Flash durch
  - echo “&inhalt_1=$meineDaten_1&inhalt_2=$meineDaten_2“;

- Zugriff in Flash auf die Variablen
  - ziel_lv.inhalt_1;
  - ziel_lv.inhalt_2;

Flash & PHP Codebeispiel:

Actionscript Code:

```
senden_lv = new LoadVars();
senden_lv.name = "Florian";
senden_lv.veranstaltung = "Flash";

empfangen_lv = new LoadVars()
empfangen_lv.onLoad = function (erfolgreich:Boolean) {
	if(erfolgreich) {
		if(empfangen_lv.ergebnis == "ok") trace("Datenübertragung erfolgreich");
	} else {
		trace("Fehler bei der Datenübertragung");
	}
}

senden_lv.sendAndLoad("testskript.php",empfangen_lv,"POST");
```

PHP Code (testskript.php):

```
<?php
$name = $_POST['name']; // Eigentlich vorher prüfen
$veranstaltung = $_POST['veranstaltung']; // Eigentlich vorher prüfen

if($name != '' && $veranstaltung != '')
	echo '&ergebnis=ok';
else
	echo '&ergebnis=fehler';
?>
```

Worklflow Flash und PHP

- PHP Skripte nicht mit Flash testen, normale echos als Ausgabe als Debugging Methode
- Für eine Variablenanzahl von kleiner 10 - 15 Load Vars Benutzen
- Für Variablenanzahl größer 15 die Daten als XML ausgeben
  - (Bessere Struktur, Objektähnliche Repräsentation)
- Für viel Datenverkehr (Anzahl Datensätze > 100) zu Flash Remoting greifen

Einblick Flash Remoting

- Technische Umsetzung über PHP und AMF Datenschnittstelle (Binärstrom)
- Prinzip: Aufruf von PHP Funktionen über Flash, effektiver Datenverkehr
- - Was braucht man:
  - Flash Remoting PlugIn für Flash
  - AMFPHP und einen Webserver
- Vorgehen
  - Aufbau von AMFPHP zeigen (Tafel) und am Laptop
  - Service Browser
  - Method Table
  - Generierung von ActionScript Klassen