

PPS-Design Eigener Homepage

4. Praxisveranstaltung

"Client Side Scripting: Java Applets"

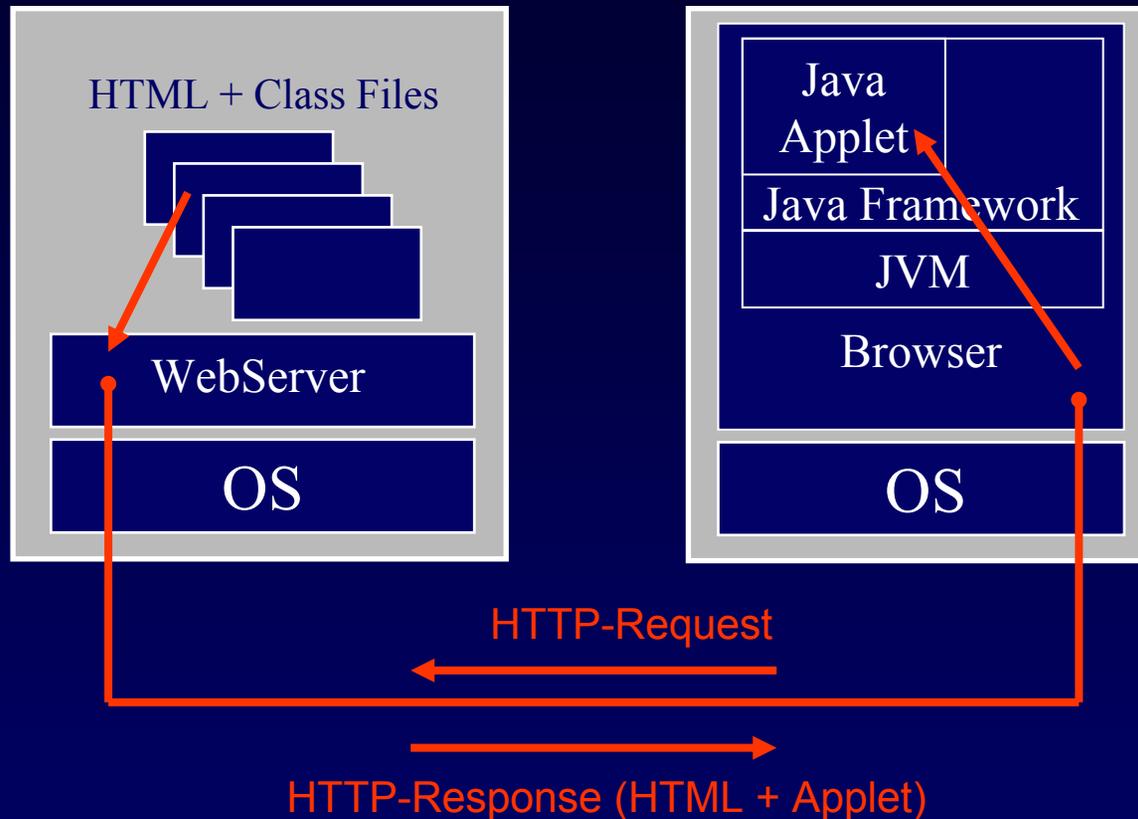
Überblick

1. Java Applets

- Architektur
- Einbettung des Applets in HTML
- Programmierung eines Applets
- Sicherheitsrisiken

2. JavaScript

Architektur



- Applets sind Java Programme, die in Webseiten eingebunden werden können.
- Der Interpreter ist im Browser integriert

Einbettung des Applets in HTML

- Ein Applet wird mittels des `applet`-Tag in eine Webseite integriert:
- Beispiel einer Webseite mit Applet:

```
<h1>Einfaches Applet</h1>
```

```
<p>In diesem Test wird ein einfaches  
HelloWorld-Applet ausgef&uuml;hrt.
```

```
<p><applet code=SimpleApplet.class  
width=500 height=100  
alt="Cannot load applet">
```

```
No applet support.
```

```
</applet>
```

*Classfile
des Applets*

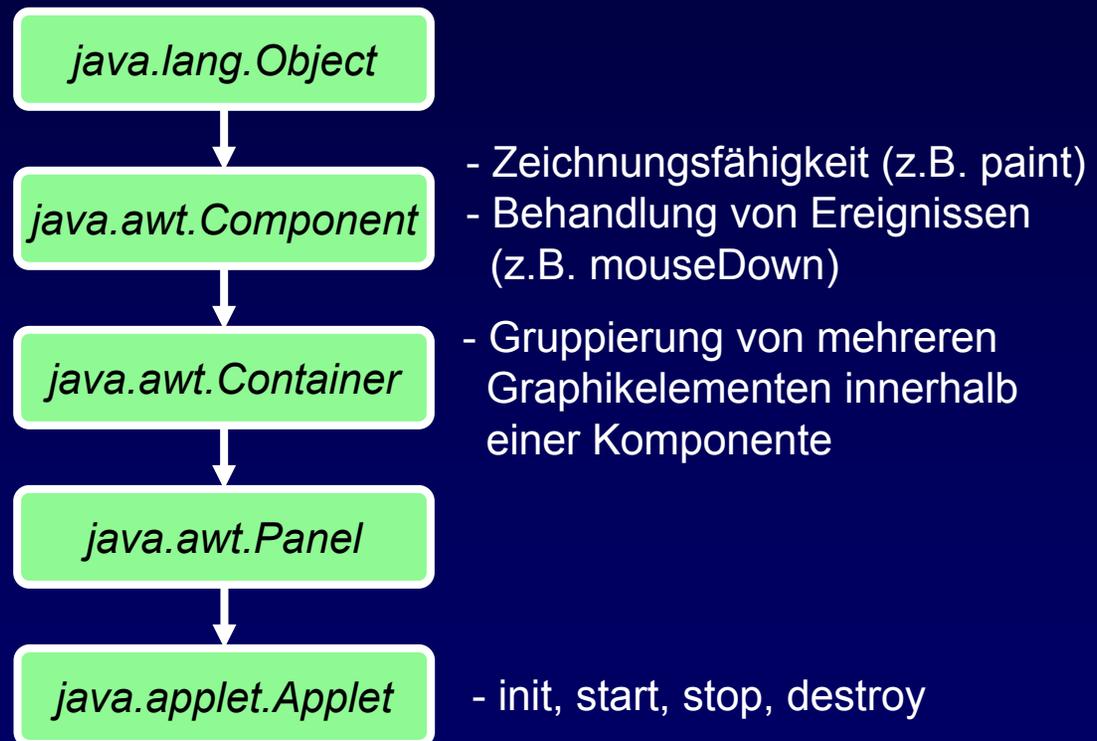
*Dimensionen
des Applets*

*Alternativer Text, falls
Browser Classfile nicht
laden oder ausführen kann*

*Alternativer Text, falls Browser
applet Tag nicht versteht*

Programmierung eines Applets

- Ein Applet ist eine Java-Klasse, die (direkt oder indirekt) von der Klasse `java.applet.Applet` abgeleitet ist.



Programmierung: SimpleApplet

```
import java.applet.*;
import java.awt.*;

public class SimpleApplet extends Applet {
    private Font font;
    public void init() {
        font = new Font("Helvetica", Font.ITALIC, 24);
    }
    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.setFont(font);
        g.drawString("Hello world.", 25, 50);
    }
}
```

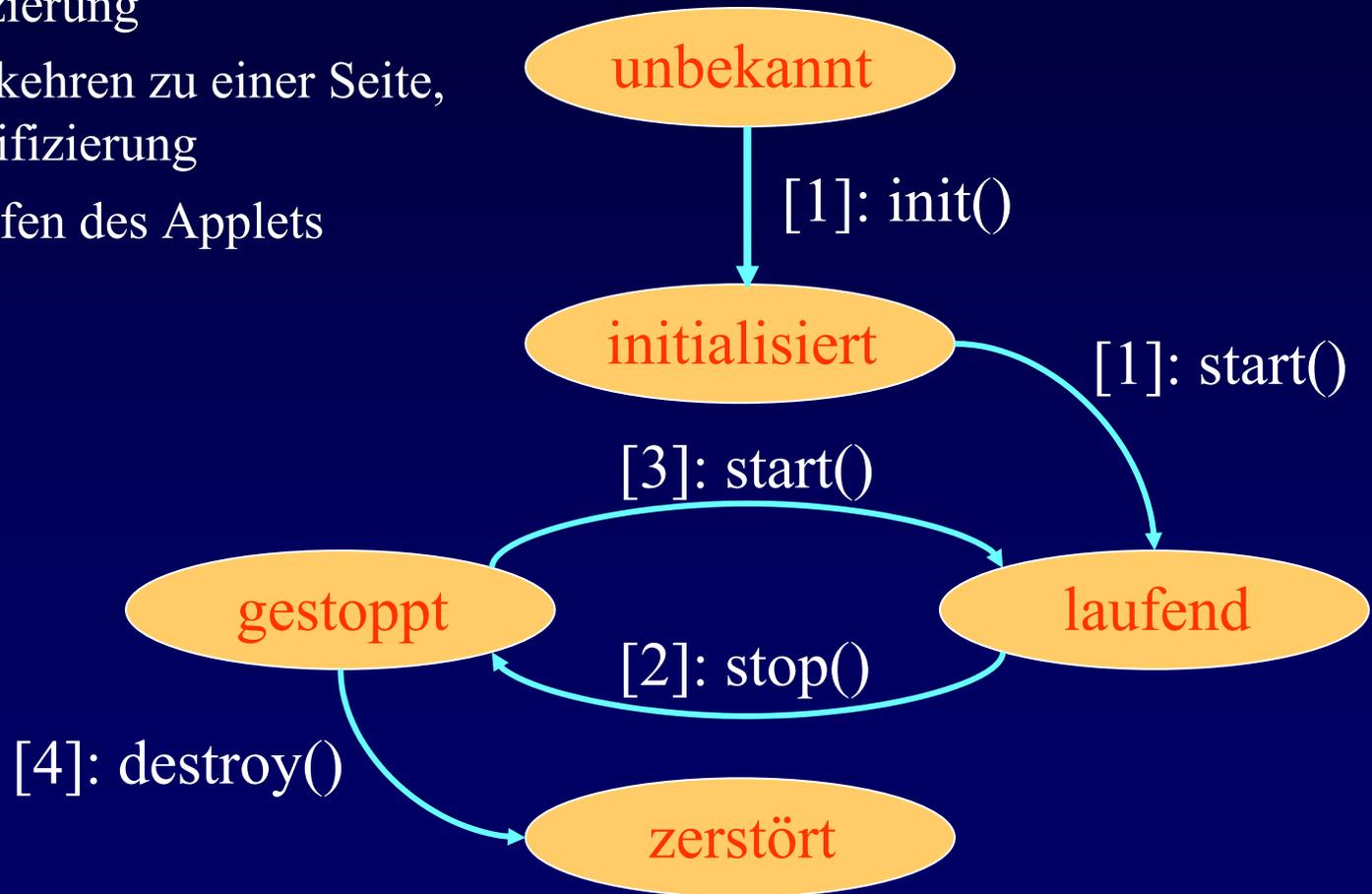
Lebenszyklus eines Applets

[1] = Laden des Applets

[2] = Laden einer neuen Seite,
Iconifizierung

[3] = Zurückkehren zu einer Seite,
Deiconifizierung

[4] = Verwerfen des Applets



Lebenszyklus eines Applets

- Der Browser ruft die folgenden Methoden auf:
 - **init**: Um das Applet zu initialisieren, wenn es das erste Mal geladen wird (oder nach einem Reload)
 - **start**: Um dem Applet anzuzeigen, dass es im Browser sichtbar ist (auch aufgerufen nach **init**)
 - **stop**: Um dem Applet zu signalisieren, dass es im Browser nicht mehr sichtbar ist wegen Scrolling oder Seitenwechsels (auch aufgerufen vor **destroy**)
 - **destroy**: Um das Applet wieder aus dem Browser zu laden (unloading) und Ressourcen wieder freizugeben

Zeichnen innerhalb eines Applets

- Ein Applet implementiert die Zeichnungsfunktionen in der `paint` Methode:

```
class SimpleApplet extends Applet {  
    ...  
    public void paint(Graphics g) {  
        g.setFont("Helvetica", Font.BOLD, 32);  
        g.drawString("my text", x, y);  
        ...  
    }  
}
```

- Wichtig:
 - Die `paint` Method wird vom Browser aufgerufen und sollte nie direkt aufgerufen werden
 - Um das Zeichnen des Applets zu forcieren, verwendet man die Methode `repaint` welche indirekt `paint` aufruft

Beispiel Applets

- SimpleApplet.java
- LifeCycle.java

Behandlung von Ereignissen

- Applets reagieren auf Ereignisse, indem sie den entsprechenden Event-Handler implementieren, z.B.:

```
public class Scribble extends Applet implements
    MouseListener, MouseMotionListener {
    private int last_x=0, last_y=0, x=0, y=0;
    private Graphics g;
    ...
    public void init() {
        addMouseListener(this); addMouseMotionListener(this);
        g = getGraphics();
    }
    public void mousePressed(MouseEvent event) {
        last_x = event.getX(); last_y = event.getY();
    }
    public void mouseDragged(MouseEvent event) {
        x = event.getX(); y = event.getY();
        g.drawLine(last_x, last_y, x, y);
        last_x = x; last_y = y;
    }
}
```

Parametrisierung von Applets

- Applets lassen sich durch Parameter im HTML-Code konfigurieren:

```
<p><applet code=ParameterDemo.class width=500
        height=100 alt="Cannot load applet">
  <param name=text value="Irgendein Text.">
  No applet support.
</applet>
```

```
<p><applet code=ParameterDemo.class width=500
        height=100 alt="Cannot load applet">
  <param name=text value="Ein anderer Text.">
  <param name=background value="FF0000">
  <param name=foreground value="FFFFFF">
  No applet support.
</applet>
```

Parameterübergabe an Applet

- Im HTML Code ([ParameterDemo.html](#)) benutzen wir unsere Parameter `text`, `foreground` und `background`
- Applet ([ParameterDemo.java](#)) kann die Parameter mit [getParameter](#) lesen:

```
public void init() {
    font = new Font("Helvetica", Font.ITALIC, 24);
    text = getParameter("text");
    fgColor = getColorParameter("foreground",
                                Color.black);
    bgColor = getColorParameter("background",
                                Color.white);

    setForeground(fgColor);
    setBackground(bgColor);
}
```

Sicherheitsrisiken

- Applets sind ein potentielles Sicherheitsrisiko, da Applets beliebigen Code enthalten können
- **Potentielle Gefahren:**
 - Lokale Dateien lesen, modifizieren oder löschen
 - Gefälschte E-Mails versenden
 - Weitergabe von sensiblen Benutzerinformationen
 - ...
- Daher: JVM, welche innerhalb des Browsers läuft, schränkt Möglichkeiten eines Applets ein

Einschränkungen für Applets

- **Restriktionen für das lokale Filesystem:**
 - Keine Dateien lesen oder schreiben
 - Keine Programme starten, keine Bibliotheken laden
 - Prüfen auf die Existenz einer Datei ist nicht erlaubt
 - Keine Verzeichnisse durchsuchen oder erzeugen
- **Restriktionen für Netzwerkoperationen:**
 - Keine Netzwerkverbindungen aufbauen (ausser zum Rechner von dem Applet ursprünglich stammt)
- **Restriktionen für Benutzerinformationen:**
 - Gewisse Benutzerinformationen lassen sich mit `getProperty` nicht lesen (`user.name`, `user.home`, `user.dir`)

Fähigkeiten eines Applets

- Applets, welche *vom lokalen Filesystem geladen* wurden (von Folder in CLASSPATH), besitzen normalerweise *keine Restriktionen*
- Netzwerkverbindungen zum Rechner, von welchem das Applet ursprünglich stammt, aufbauen.
- Audio Clip Dateien spielen:
 - Beispiel 1 ([HTML](#), [Applet](#))
 - Beispiel 2 ([HTML](#), [Applet](#))
- Public Methoden anderer Applets in der gleichen Webseite aufrufen

Exceptions bei unerlaubten Zugriffen

- Unerlaubte Zugriffe erzeugen *Exceptions*, z.B.:

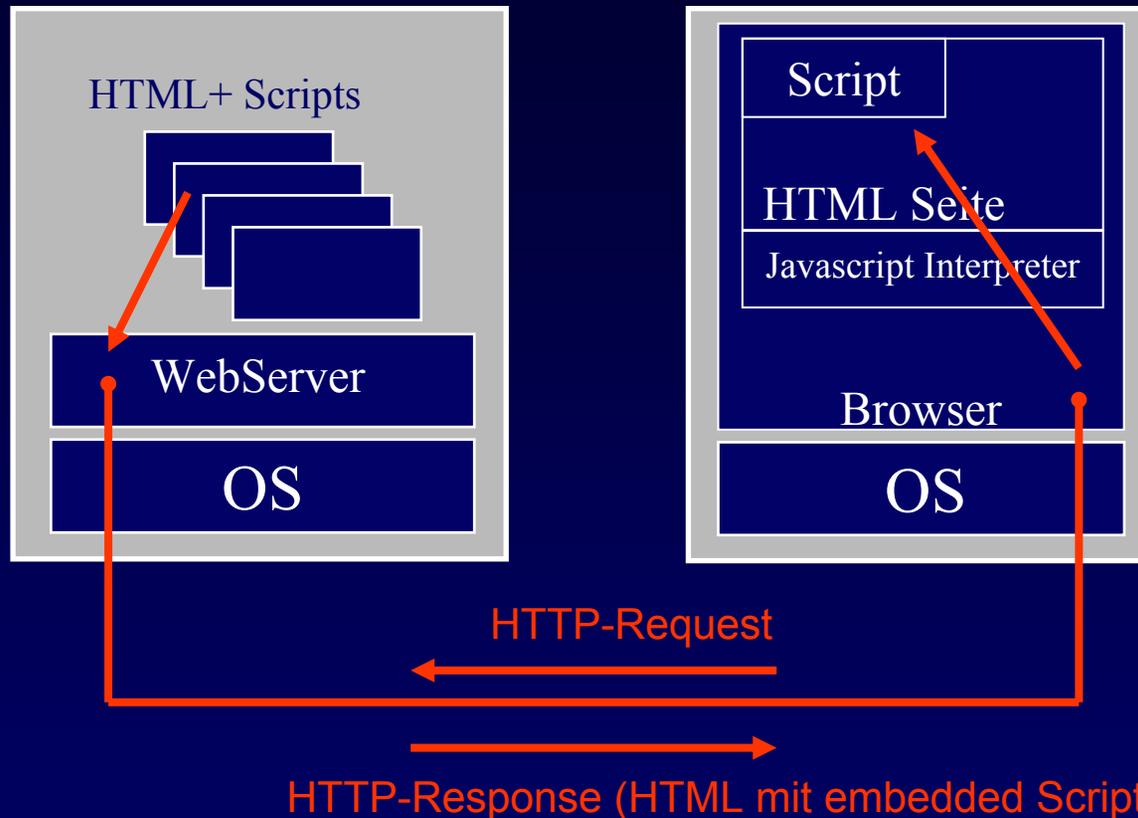
```
public void init() {  
    ...  
    try {  
        Properties p = System.getProperties();  
        msg1="Name: " +p.getProperty("user.name");  
        msg2="Home: " +p.getProperty("user.home");  
    }  
    catch (Exception e) {  
        msg1 = "Ouch: " + e.getMessage();  
        msg2 = "";  
    }  
}
```

JavaScript

- Crossplatform, object-oriented language
- Besteht aus drei Mengen:
 - Core set
 - Server-side JavaScript (erweitert das core set mit server-side Funktionen)
 - Client-side JavaScript: erweitert das core set und bietet Objekte an, mit den man einen Browser manipulieren kann.

Unter dem Gebrauch von client-side JavaScript kann man auf user-input, wie z.B. mouse clicks, page navigation events und input in Formen, reagieren

Architektur



- Skripte werden embedded in der HTML Seite vom Webserver dem Browser geliefert und werden vom Browser's (darum client-side) JavaScript Interpreter exekutiert.

Einbettung von Scripts

- Man gebraucht das `<SCRIPT>` HTML Tag, um JavaScript code (statements, functions) einzubetten
- Man gebraucht JavaScript Event Handlers als HTML Attributen (z.B. in Forms)
- Man kann auch ein File mit JavaScript source file referenzieren.

```
<HEAD><TITLE>A Simple Document</TITLE>
<SCRIPT>
function update(form) {
alert("Form being updated")
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myform" ACTION="foo.cgi" METHOD="get" >Enter a value:
<INPUT TYPE="text" NAME="text1" VALUE="" SIZE=20 >
Check if you want:
<INPUT TYPE="checkbox" NAME="Check1" CHECKED
onClick="update(this.form)" > Option #1
<P>
<INPUT TYPE="button" NAME="button1" VALUE="Press Me"
onClick="update(this.form)" >
</FORM>
</BODY>
```

JavaScript Funktion

Event Handlers

"Alltag" Gebrauch von JavaScript

- Event Handling (Validierung, Animationen)
- Navigation ("Inter-Frame" Communication, Bearbeitung von mehreren Frames auf einmal, client-side image maps, cookies, status bar interaction)

Javascript "Flavours":

- JavaScript (Netscape)
- Jscript (Microsoft)
- ECMA-262 Specification

Referenzen

Java Applets:

- <http://java.sun.com/applets/> (resources)
- Java Tutorial :
<http://java.sun.com/docs/books/tutorial/index.html>
- Java-World Artikel ueber Applet Security:
<http://www.javaworld.com/javaworld/jw-12-2000/jw-1215-security.html>

JavaScript:

- [JavaScript Manual / Reference](#)
- [Client-Side Scripting and HTML \(W3C\)](#)

Danke fuer Ihre Aufmerksamkeit !



Fragen