Web-Programmierung (WPR)

Vorlesung IV.

JavaScript

Manfred Gruner

mailto:wpr@gruner.org

Offene Fragen

1. Unterschied *Link* und *Style* Tag

- Link = HTML Syntax
  wird/kannl von anderen Programmen ausgewertet werden.

```
<link rel="stylesheet" type="text/css"
  href="./style/default/main.css" media="all" />
```

- Style @import = CSS-Syntax

```
<style type="text/css" media="all">@import
  "./style/default/main.css";</style>
```

Grundsätzlich: beide Möglichkeiten führen zum selben Ergebnis!

Offene Fragen

2. Pseudo Klassen bei **a**-Tag

- `a:active`
  Link der aktuell selektiert wird
- `a:hover`
  Link über den Mouse fährt
- `a:visited`
  Link, der schon angewählt wurde

6. JavaScript

- Leistungsmerkmale
  - Zugriff auf Bestandteile des gelad. Dokumentes
  - Dynamische Änderung des gelad. Dokumentes
  - Steuerung externer Komponenten
  - Reaktion auf Benutzereingaben
  - Fenstermanagement

6. JavaScript

- Spracheigenschaften
  - Scriptsprache
  - Objektorientierte Sprache
  - nutzt eine vertraute Syntax (C, Java, Perl)

6.1 Anwendungsformen

- Anwendungsformen
  JavaScript-Code ist Bestandteil eines HTML-Dokumentes

- JavaScript Contexte
  - Script Tag
  - Attributwert eines Event Handlers
  - Wert von Attributen die gewöhnliche URLs enthalten

Beispiel 1 .. 3

6.1 Anwendungsformen

- Eventhandler
  - Eventhandler mit HTML4.0 Standard
  - Events = Attribute von HTML-Tags
    - z.B.: `<a onClick="...">Link</a>`
- **JavaScript URLs**
  - Definition: `javascript:`***`JavaScript-code`***

Beispiel 4

6.2 Sprachgrundlagen

- Datentypen
  - JavaScript ist nicht typisiert
  - intern(zahl, boolean, string + array,object)

- Operatoren
  - Arithmetische Operationen (+, -, *, /, %)
  - Zuweisungsoperationen (=, +=,+= ,...)
  - Vergleichsoperatoren (>, >=, <, <=, ==, !=)
  - Stringkonkatenation (+)
  - Logische Operatoren (&&, ||, !)
  - Bit-Operationen (&, |,^, ~,<< ,>> ,>>>)

6.2 Sprachgrundlagen

- Variablen
  - Kann über Lebenszeit verschiedene Typen enthalten
  - Muss nicht explizit definiert werden.
  - Lesen einer nicht definierten Variable
    => Laufzeitfehler
  - Definition ohne Wertzuweisung
    => undefined
  - Explizite Definition mit **`var`** Ausdruck

6.2 Sprachgrundlagen

- Kontrollstrukturen
  - Verzweigungen

```
if(boolscher Ausdruck)
   Anweisung
[else
   Anweisung]

switch(skalarausdruck){
 case Literal 1: Anweisungen
 [default: Anweisungen
}
```

6.2 Sprachgrundlagen

- Kontrollstrukturen (Schleifen)
  - `while(Boolscher Ausdruck)`
    `Anweisung`
  - `do`
    `Anweisung`
    `while(Boolscher Ausdruck);`
  - **`for(Ausdruck1;boolscher Ausdruck;Ausdruck3)`**
    `Anweisung`
  - `for(arrayelement in array)`
    `Anweisung`

6.2 Sprachgrundlagen

- Funktionen

```
function Funktionsname(parameter)
{
    var result = parameter+1;
    return result;
}
```

WICHTIG: „declare before use“

6.2 Sprachgrundlagen

- Einbinden von JavaScript in HTML
  - Als embedded JavaScript direkt im Script Tag

```
<script>
   function goodByeWpr()
     {....}
</script>
```

  - Als externe Datei

```
<script Language =“JavaScript“
src=“myjs.js“ />
```

6.2 Sprachgrundlagen

- Objekte
  - Knoten des DOM werden als Objekte bereitgestellt
  - Zugriff auf Attribute und Methoden mit .
    - document.writeln(„Hallo JavaScript“);
    - var version = navigator.appVersion;
    - var vname = document.form1.name.value;
  - Definition und Erzeugen eigener Objekte
    - var wuerfel = new Wuerfel();
  - Sprach-Objekte
    - Date, Math, String, Function, Array

6.2 Sprachgrundlagen

Eigene Objekte

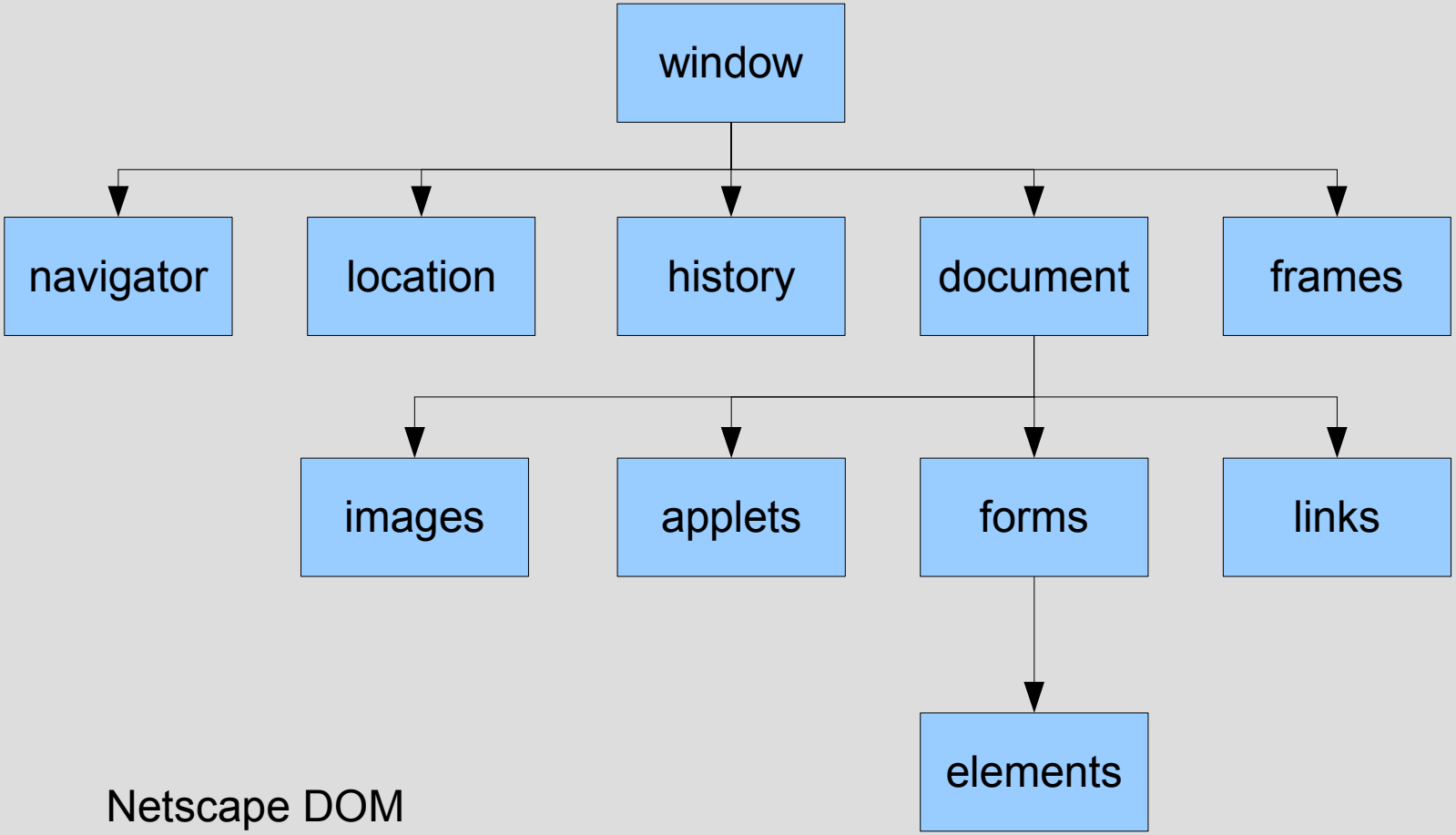
```
function Farbe (R, G, B) {
   this.R = R;
   this.G = G;
   this.B = B;
   this.hex = "#";
}

myFarbe = new Farbe(10,10,10);
```

6.2 Sprachgrundlagen

- Arrays
  - var myArray = new Array(„eins“, „zwei“, „drei“);
  - var element = myArray[0]
- Assoziativ Arrays
  - var myArray = new Array();
  - myArray[„vorname“] = „Hubert“;
  - myArray[„nachname“] = „Ka“;

6.3 JavaScript Client Objekte



Netscape DOM

6.3.1 window-Objekt

- Attribute
  - `closed` = gibt an, ob window noch offen ist
  - `defaultStatus`=Standardtext in der Statuszeile
  - `status` = transienter Text in der Statuszeile
  - `name`
  - `opener` = Opener vom aktuellen window-Objekt
  - `length` = Anzahl Frames in window
- Framebasierte Attribute
  - self = synonym für aktuelle fenster
  - `top` = oberstes window-Object in einer Hierarchie
  - `parent` = direkt übergeordnetes Fenster

6.3.1 window-Objekt

- Methoden
  - open(„ulr“,“name“,[properties...]);
  - close()
  - focus()

  - alert(„Message“);
  - confirm(„Message“); (OK und Cancel Button)
  - prompt(„text“,“default“); Eingaben vom Benutzer

6.3.7 document-Objekt

- Attribute
  - bgColor
  - ...
  - referrer (URL von dem das Dokument geladen wurde)
- Methoden
  - open([mime-type])
  - close()
  - write(text)
  - writeln(text)

6.3.7 form-Objekt

- Attribute
  - Action = URL
  - Elements = Array mit Formular Elementen
  - Length = Anzahl Elemente in elements
  - Method = HTTP-Method zur Datenübertragung
  - Name = Name des Formulares
- Methoden
  - submit()

6.4 Events

- OnBlur
- OnChange
- OnClick
- OnError
- OnFocus
- OnLoad
- OnMouseOver
- OnMouseOut
- OnReset
- OnSubmit
- OnSelect
- OnUnload

6.7 DOM-Nutzung

- Moderne JavaScript-Features benötigen DOM-Level1 oder DOM-Level2
  Modifikation der Dokumentenstruktur
  Objekt *Node*
  - Attribute
    `nodeName, nodeType, parentNode, childNodes, firstChild, lastChild`
  - Methoden
    `hasChildNodes(), appendChild(), removeChild(), getAttribute(), setAttribute()`

6.7 DOM-Nutzung

- Zugriff auf Dokumentenstruktur über `document`-Objekt
  - getElementById()
  - getElementsByName()
  - getElementByTagName()