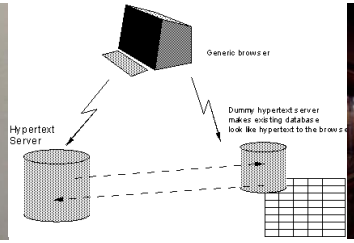
JOURNALISTISCHES SEMINAR

JG|U



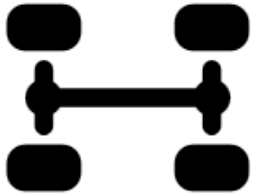
Interaktive Medien: DIY

Prof. Thomas Weibel, lic. phil. hist., Journalist BR

1. Javascript
2. Variablen
3. Javascript und HTML
4. Bedingte Anweisungen
5. Schleifen
6. Kommentare
7. Hausaufgabe

Javascript

1. HTML: Das Chassis



2. CSS: Die Karosserie

3. Javascript: Der Motor

Website, Web-App

- Quellcode — universell
- Compiler — universell (Webbrowser), kostenlos
- Software — universell (HTML)
- Kauf, Installation — kostenlos

- HTML 5 — Struktur und Inhalt
- CSS 3 — Formatierung und Layout
- Javascript — Programmierung

Definition

»Javascript ist eine Skriptsprache, deren Aufgabe es ist, Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu erzeugen und so die Möglichkeiten von HTML und CSS zu erweitern.«

Aufbau

- Javascript erweitert die Funktionen einer Website.
- Mittels »Funktionen«, »Objekten«, »Methoden« und »Eigenschaften« und »Werten« werden Routinen beschrieben. Der Browser übersetzt diese Routinen dann in Funktionalitäten. Beispiel:

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

- Auch Javascript-Dateien bestehen aus reinem Text ohne Formatierung.

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
    Funktion
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
```

Objektdeklaration (Variable)

```
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
```

Objekt

```
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
```

Methode

```
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
                                          Wert
  meinTitel.innerText="Hallo Welt!";
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
```

Objekt

```
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
```

Eigenschaft

```
}
```

Grundelemente

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Wert

Variablen

Variablen sind flexible Objekte. In Variablen können verschiedene Typen von Information gespeichert werden.

Typ	Beispiele
Zahl	`12, 0, -7, 1.23`
String	`"Hallo Welt!", "Desoxyribonukleinsäure"`
Wahrheitswert (Boolean)	`true, false`
Objekt	HTML-, Zeit-, Audioobjekt
Array	Bündel von Variablen
null	
undefined	

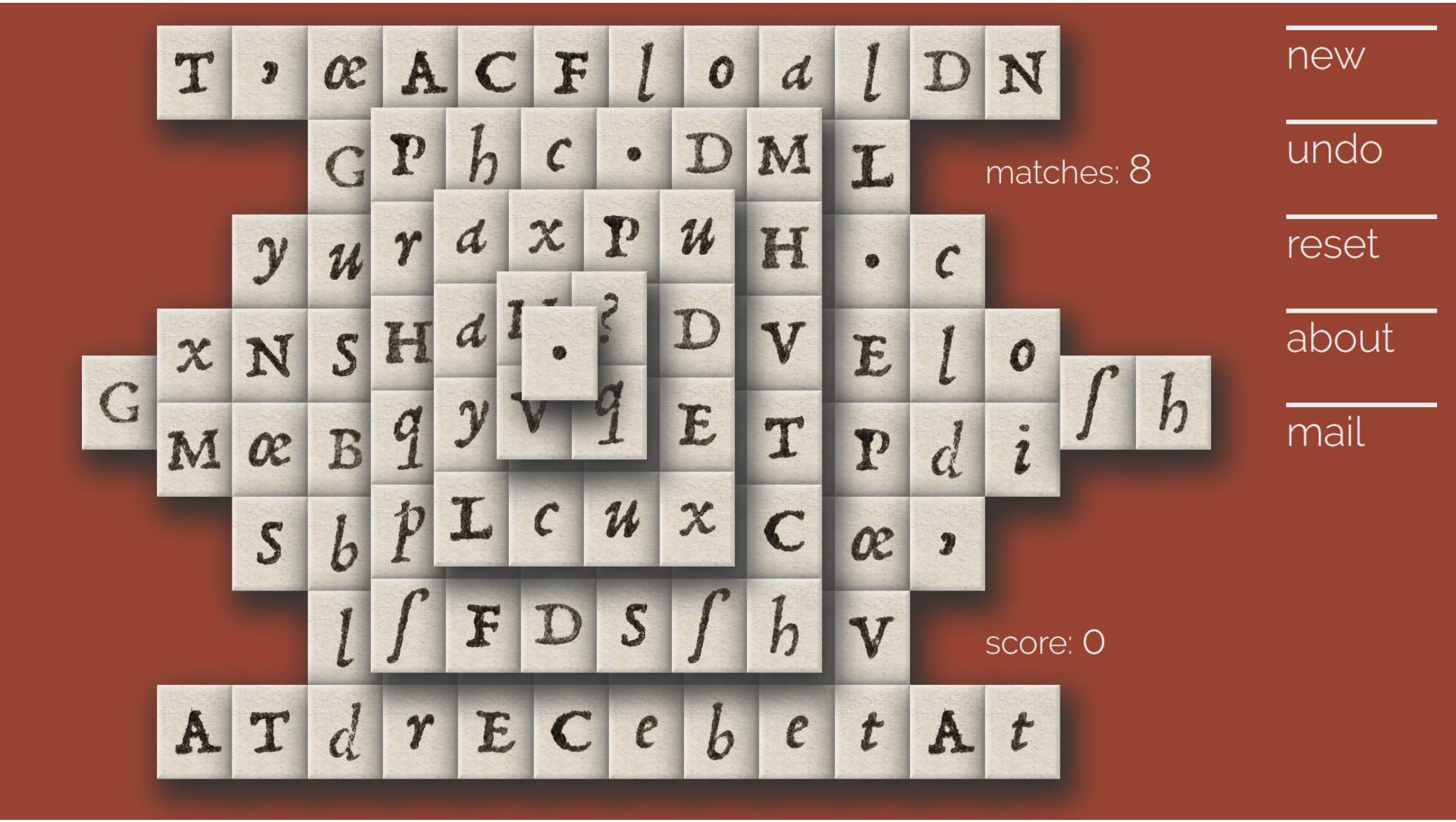
```
function shuffle() {

    // generate random number array
    rand=[];
    for (k=0; k<144; k++) {
        random=parseInt(Math.random()*(k+1));
        rand[k]=rand[random];
        rand[random]=k;
    }
}

function check(a) {
    playable(a); // check if playable

    // check animation
    if (cond) { // check selection rules

        // mark if one tile selected
        if (store.length<2) {
            el=document.getElementById(a);
            store.push(el);
```


new
undo
reset
about
mail
matches: 8
score: 0

Aufbau

- Alle Anweisungen (»Befehle«) werden durch ein Semikolon **;** abgeschlossen.
- Alle Funktionen* werden zwischen geschweifte Klammern **{ }** gesetzt.

  *sowie Schleifen und bedingte Anweisungen, sofern sie mehr als eine Anweisung enthalten

  Mac:

  { = ALT + 8

  } = ALT + 9

  Windows:

  { = ALT GR + ä

  } = ALT GR + $

Aufbau

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Aufbau

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Aufbau

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Aufbau

Geschweifte Klammern

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Aufbau

keine Leerzeichen

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Aufbau

Semikolons

```
function hallo() {
  var meinTitel=document.getElementById("titel");
  meinTitel.innerText="Hallo Welt!";
}
```

Zahl

- In einer Variablen lässt sich eine natürliche Zahl speichern. Dezimalstellen werden mittels eines Punkts . (*nicht* eines Kommas) abgetrennt.

```
var a;
a=17.2;
```

oder

```
var a=17.2;
```

String

- In einer Variablen lässt sich eine Buchstabenfolge speichern. Sog. »Strings« werden zwischen Hochkommas `""` gesetzt.

```
var b;
b="rot";
```

oder

```
var b="rot";
```

Wahrheitswert

- In einer Variablen lässt sich ein Wahrheitswert speichern. »Boolean« sind `true` oder `false`. Wahrheitswerte werden nicht zwischen Hochkommas `""` gesetzt.

```
var b;
b=true;
```

oder

```
var b=true;
```

Objekt

- In einer Variablen lässt sich ein Objekt speichern. Objekte können in Javascript alles sein – häufige Objekte sind das Browserfenster, das aktuelle HTML-Dokument, HTML-Tags, das Zeit-, Audio- oder Videoobjekt.

```
var b;
b=new Audio("musik.mp3");
```

oder

```
var b=new Audio("musik.mp3");
```

Array

- In einer Variablen lässt sich auch ein Array, ein ganzes Bündel von Variablen jeden Typs, speichern. Die Einzelvariablen werden durch Kommas , abgetrennt.

```
var b;
b=new Array("rot", "gelb", "blau");
```

oder

```
var b=[3, -7, 13.2];
```

Array

HTML-Attribut	Anmerkung	Befehl, Ausgabe
`var b=[3, -7, 13.2];`	Anzahl der Variablen	`alert(b.length);` 3
	Ausgabe der dritten Variablen (Indexziffer 2)	`alert(b[2]);` 13.2
	Ausgabe des gesamten Arrays	`alert(b);` 3, -7, 13.2

Array: Methoden

HTML-Attribut	Anmerkung	Befehl, Ausgabe

`var b=[3, -7, 13.2];`
`var a=[11,5.4]`
 	
`b.push(6.6);`
	Variable am Ende anfügen	`alert(b)`
		
		3, -7, 13.2, 6.6

`b.slice(0,1);`	1 Variable an Indexziffer 0 löschen	-7, 13.2
`b.pop();`	Letzte Variable löschen	3, -7
`b.shift();`	Erste Variable löschen	-7, 13.2
`b.indexOf(4.4);`	Suche nach Variablen 3, Ausgabe der Indexziffer (Wert -1 für »nicht gefunden«)	-1
`b.concat(a);`	Array a an Array b anfügen	3, -7, 13.2, 11, 5.4
`b.reverse();`	Reihenfolge der Variablen umkehren	5.4, 11, 13.2, -7, 3

Variablen übergeben

- Anstelle einer formellen Deklaration können Variablen auch im Funktionsaufruf übergeben werden.

```
<button onclick="fenster('Hallo!')">Klick
mich!</button>

function fenster(a) {
  alert(a);
}
```

Globale und lokale Variablen

Die Deklaration von Variablen kann unterschiedliche Geltungsbereiche (»scope«) haben. Eine Variable kann für das gesamte Skript (»global«) oder nur für eine einzelne Funktion (»lokal«) deklariert werden.

- Grundsatz: Variablen nur dann global deklarieren, wenn sie über mehrere Funktionen hinweg benötigt werden.

Globale und lokale Variablen

HTML	Anmerkung	Ergebnis

`var a="Hallo!";`
 	
`function fenster() {`
`  alert(a);`
`}`
	Globale Variable	`Hallo!`

`function fenster() {`
`  var a="Hallo!";`
`  alert(a);`
`}`
	Lokale Variable	`Hallo!`

`function fenster() {`
`  var a="Hallo!";`
`}`
 	
`alert(a);`
	Lokale Variable	*`a is undefined`*

Verknüpfung mit HTML

Javascript-Funktionen lassen sich auf drei Arten mit HTML-Dokumenten verknüpfen.

1. inline (nicht empfohlen)
2. mithilfe des `<script>`-Tags (nur bei kleinen Projekten)
3. mithilfe eines externen Skripts (empfohlen)

Verknüpfung mit HTML

Javascript-Funktionen lassen sich auf drei Arten mit HTML-Dokumenten verknüpfen.

1. inline (nicht empfohlen)
```
<button onclick="alert('Hallo!')"></button>
```
2. mithilfe des `<script>`-Tags (nur bei kleinen Projekten)
```
<head>
  <script>
    alert("Hallo!");
  </script>
</head>
```
3. mithilfe eines externen Skripts (empfohlen)

Verknüpfung mit HTML

HTML	JS

`<doctype html>`
`<html>`
`  <head>`
`  <title>Websitename</title>`
`  <script src="skript.js" />`
`  </script>`
`  </head>`
`  <body>`
`  </body>`
`</html>`
	`alert("Hallo!");`
	
	oder
	`function fenster() {`
	`  alert("Hallo!");`
	`}`

Eventhandler

- Mit einem Eventhandler wird gesteuert, was geschehen soll, wenn ein bestimmtes Ereignis eintritt. Auslösende Events können unterschiedlich sein, werden aber oft durch eine Userinteraktion ausgelöst.

```
<button onclick="fenster()">Klick mich!</button>
```

oder

```
<div onmouseover="fenster()">Klick mich!</div>
```

oder

```
<body onload="fenster()">…</body>
```

Eventhandler

HTML-Attribut	Aktion
`onclick`	nach Klicken
`onload`	nach vollständigem Laden
`onmouseover`	beim Überfahren mit dem Mauszeiger
`onmouseout`	beim Verlassen des Mauszeigers
`onkeydown`	beim Drücken einer Taste
`onchange`	beim Verändern eines HTML-Elements
`ontouchstart`	beim Berühren des Elements
`ontouchend`	beim Loslassen des Elements
`ontouchmove`	beim Bewegen des Fingers
`ontouchcancel`	beim Abbrechen der Berührung

Häufige Fehler

JS	Fehler
`function fenster () { }`	Leerschlag zwischen Funktionsname und Klammern
`alert("Hallo!")`	Fehlendes Semikolon am Ende der Befehlszeile
`alert(a);`	Fehlende Variablendeklaration

Bedingte Anweisungen

- Mittels `if-else`-Anweisungen lässt sich die Ausführung von Code an Bedingungen knüpfen. Der Term »ist gleich« wird mit zwei Gleichheitszeichen == notiert; Werte *null* oder *Boolean* mit drei Gleichheitszeichen ===. Mit `else if` lässt sich der Bedingungssatz erweitern.

```
if (a=="Hallo") {
  alert(a);
}

else {
  alert("Welt");
}
```

Verknüpfung von Bedingungen

- Bedingungen können mittels der Operatoren und `&&` bzw. oder `||` miteinander verknüpft werden.

```
if (a=="Hallo" || a=="Welt") {
  alert(a);
}
```

oder

```
if (a>0 && a<=10) {
  alert(a);
}
```

Schleifen

- Mittels Schleifen lassen sich wiederkehrende Operationen rationalisieren. Ihre häufigste Form ist die `for`-Schleife. Der Zähler `i` beginnt dabei normalerweise bei Null; die Kurznotation `i++` steht dabei für `i=i+1`.

```
for (i=0; i<3; i++) {
  alert("i ist gleich "+i);
}
```

oder

```
for (i=0; i<3; i++) {
  alert("Fenster Nr. "+(i+1));
}
```

Schleifen

JS	Anmerkung

`for (i=0; i<3; i++) {`
`  …`
`}`
	Schleife wird exakt dreimal ausgeführt

`var i=0;`
`while (i>=0 && i<3) {`
`  …`
`}`
	Schleife wird beliebig oft ausgeführt, sofern i grösser als null und kleiner als 3 ist (Prüfung *vor* Beginn der Ausführung).

`var i=0;`
`do {`
`  …`
`} while (i>=0 && i<3);`
	Schleife wird beliebig oft ausgeführt, sofern i grösser als null und kleiner als 3 ist (Prüfung *nach* Beginn der Ausführung).

Kommentare

JS	Anmerkung

`function fenster() {`
`  // alert("Hallo!");`
`}`
	Auskommentieren einer Befehlszeile

`function fenster() {`
`  alert("Hallo!"); // ein Popup`
`}`
	Auskommentieren einer Anmerkung

`/* function fenster() {`
`  alert("Hallo!");`
`} */`
	Auskommentieren eines ganzen Abschnitts

Nachschlagewerke

- w3schools.com (englisch)
- de.selfhtml.org (deutsch)

Hausaufgabe: developer.mozilla.org

- Javascript, Kapitel »Javascript Basics«

Hausaufgabe (alternativ): Lektüre

- Peter Bühler, Patrick Schlaich, Dominik Sinner: HTML 5 und CSS 3
- Peter Bühler, Patrick Schlaich, Dominik Sinner: Webdesign