Multimedia im Netz

Wintersemester 2013/14

Übung 03 (Hauptfach)

HTML5

- Mit HTML5 wurden neue Features eingeführt (nur Beispiele):
  - Neue Elemente:
    - **<canvas></canvas>**
    - <audio></audio>
    - <video></video>
    - …
  - Formular Features (Beispiele):
    - Platzhalter
    - Validierung
    - …
  - Drag and Drop

HTML5: Dokumentaufbau

```
<!DOCTYPE html>
<html lang="de">
<head>
   <meta charset="UTF-8" />
   <title>HTML5 Grundgeruest</</title>
</head>

<body>
</body>
</html>
```

HTML5: Canvas

- Das <canvas> Element ist ein Container der im HTML Code eingebettet ist:

```
<canvas width="400" height="400"
                    style="border:1px solid #000000;">
    Browser does not support the canvas tag.
</canvas>
```

- In HTML5 wird für das <canvas> Element der *Immediate Mode* genutzt

HTML5: Kontext

- Das Zeichnen selbst erfolgt über JavaScript. Dazu wird der Kontext des Canvas benötigt: **`getContext();`**

- Der Kontext ist ein Objekt mit eigenen Eigenschaften und Methoden, die man verwenden kann um innerhalb des Canvas-Element zu zeichnen.

- Es gibt zwei Kontexte:
  - 2D
  - 3D (webgl)

JavaScript

- JavaScript ist eine eigene Programmiersprache
- Code wird vom Webbrowser interpretiert
- Code kann in HTML integriert sein

```
<script>
<!-
    Hier kommt das Skript
-->
</script>
```

- Oder in einer eigenen Datei liegen

```
<script src="myScript.js"></script>
```

DOM (Document Object Model)

- Mit DOM kann jedes Element und dessen Inhalt in einem HTML (und XML) Dokument referenziert werden
- Die Elemente, ihr Inhalt und ihre Struktur kann modifiziert werden
  - `document`: Inhalt der im Browserfenster angezeigt wird
  - `getElementById()`: Greift auf HTML-Element mit passender ID zu
  - `getElementByTagName()`: Greift auf Tags anhand ihres Namen zu
  - `Knoten.firstChild` : Liefert den ersten Kindknoten
  - `Knoten.nodeValue`: Setzt/Liefert den Wert eines Knotens

- http://de.selfhtml.org/javascript/index.htm

DOM und JavaScript

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8"/>
    <title>HTML 5</title>
</head>
<body>
    <canvas id="canvas" width="400" height="400"
            style="border:1px solid #c3c3c3;">
        Your browser does not support the HTML5 canvas tag.
    </canvas>

<script>
    var canvas=document.getElementById("canvas");
</script>
</body>
</html>
```

Auf den Kontext Zugreifen

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8"/>
    <title>HTML 5</title>
</head>
<body>
    <canvas id="canvas" width="400" height="400"
            style="border:1px solid #c3c3c3;">
        Your browser does not support the HTML5 canvas tag.
    </canvas>

<script>
    var canvas=document.getElementById("canvas");
    var context = canvas.getContext("2d");
</script>
</body>
</html>
```

JavaScript and Canvas

- Farben und Stile festlegen
  - fillStyle
  - strokeStyle
- Rechtecke zeichnen
  - rect();
  - fillRect();
  - strokeRect();
- Bilder auf den Canvas zeichnen
  - drawImage()
- Weitere Funktionen
  http://www.w3schools.com/tags/ref_canvas.asp

Ein Rechteck Zeichnen

```
…
<script>
    var canvas=document.getElementById("canvas");
    var context = canvas.getContext("2d");

    context.fillStyle="#00ff00";
    context.fillRect(0,0, 150, 100);
</script>
</body>
</html>
```

Exkurs: Objektorientiertes JavaScript (I)

- Der „normale“ Programmierstil in JavaScript bringt einige Nachteile mit sich:
  - Sprache, die auf globalen Variablen basiert
  - Variablen können unbeabsichtigt überschrieben werden
  - Einbinden mehrerer JS-Dateien kann zu Konflikten führen
  - Verlust der Übersichtlichkeit
- Idee: Man fasst zusammengehörige Attribute und Methode in einem Objekt zusammen

Exkurs: Objektorientiertes JavaScript (II)

- Es existieren verschiedene Möglichkeiten, ein Objekt in JavaScript zu erzeugen:
  - Konstruktor-Funktionen
  - Literale Notation

- Welche Variante sollte man verwenden?
  - Situationsabhängig
  - Konstruktor-Funktionen:
    - Wenn man mehrere Instanzen eines Objekts benötigt
  - Literale Notation:
    - Wenn man nur ein Objekt im gesamten Skript benötigt
    - Name Spacing

Beispiel: Konstruktor Funktion (I)

```
function Rabbit(){
    this.adjective = "fat";
    this.whatAmI = function(){
        alert("I am a " + this.adjective + " Rabbit!");
    }
};

var fatRabbit = new Rabbit(); fatRabbit.whatAmI();
```

- Eigenschaften sind Variablen
- Methoden sind Funktionen

Beispiel: Konstruktor Funktion (II)

```
function Rabbit(adjective){
    this.adjective = adjective;
    this.whatAmI = function(){
        alert("I am a " + this.adjective + " Rabbit!");
    }
};

var fatRabbit = new Rabbit("fat");
fatRabbit.whatAmI();

var whiteRabbit = new Rabbit("white");
whiteRabbit.whatAmI();
```

Beispiel: Literale Notation

```
var rabbit = {
    adjective : 'fat',
    whatAmI : function(){
        alert("I am a " + this.adjective + " Rabbit!");
    }
};

rabbit.whatAmI();

rabbit.whatAmI(); rabbit.adjective = "black";
rabbit.whatAmI();
```

Übungsblatt 3

- **Thema: Schiebepuzzle mit JavaScript und HTML5**
- Bearbeitungszeit: 2 Wochen
- Abgabe: 27.11.2013 23:00 Uhr

- Übungen für das HF am Montag (18.11.) und Mittwoch (20.11.) dienen als Programmierberatung (mit Voranmeldung)

Danke! Fragen?