

# Lernjournal

Patrick Günthard

5ia13c Berufsbildungsschule Winterthur

## Contents

<b>1</b>	<b>Wöchentliches Journal</b>	<b>2</b>
1.1	<2016-09-06> . . . . .	2
1.2	Gemacht . . . . .	2
1.3	<2016-09-13> . . . . .	2
1.4	Gemacht . . . . .	2
1.5	Was habe ich gelernt? . . . . .	2
1.6	<2016-09-20> . . . . .	2
1.7	Gemacht . . . . .	2
1.8	<2016-09-27> . . . . .	2
1.9	Gemacht . . . . .	2
1.10	Was Ich heute gelernt habe . . . . .	2
1.11	<2016-10-04> . . . . .	2
1.12	Gemacht . . . . .	2
1.13	Was Ich heute gelernt habe . . . . .	2
1.14	<2016-11-15> . . . . .	3
<b>2</b>	<b>Dokumentierung der Themen</b>	<b>3</b>
2.1	Bilder . . . . .	3
2.1.1	Bildformate . . . . .	3
2.1.2	Downloaddauer einer Datei . . . . .	3
2.1.3	Auflösung und Farbtiefe . . . . .	4
2.1.4	Bildergalerie . . . . .	5
2.2	Juristische Fragen . . . . .	5
2.2.1	Proprietäre Inhalte . . . . .	5
2.2.2	Freie Lizenzen . . . . .	6
2.3	Banner . . . . .	8
2.3.1	Verwendung . . . . .	8
2.3.2	Beispiele . . . . .	8
2.3.3	Implementierung in diesem Projekt . . . . .	8
<b>3</b>	<b>Über dieses Dokument</b>	<b>9</b>

# **1 Wöchentliches Journal**

## **1.1 <2016-09-06>**

### **1.2 Gemacht**

Wir haben angefangen, die verschiedenen Komponenten welche wir verwenden wollen, herunterzuladen und das Projekt aufzusetzen.

## **1.3 <2016-09-13>**

### **1.4 Gemacht**

Ich habe an der Web-Applikation weitergearbeitet. Ich arbeite schon seit einigen Tagen daran gearbeitet. Heute habe ich die Login- und Signup funktion fertig implementiert und getestet. Ich habe auch begonnen, eine logout funktion zu implementieren.

### **1.5 Was habe ich gelernt?**

Ich habe nichts neues hinzugelernt, da ich alle dinge welche ich implementiert habe, schon einmal gemacht habe.

## **1.6 <2016-09-20>**

### **1.7 Gemacht**

Heute habe ich am Backend gearbeitet. Ich habe weiter Rest-Services implementiert.

## **1.8 <2016-09-27>**

### **1.9 Gemacht**

Heute habe ich am Frontend weitergearbeitet. Ich habe auch recherchiert, wie man mit **node js** einen file-upload zum laufen bringen kann.

### **1.10 Was Ich heute gelernt habe**

Ich habe mich mit File Upload mit node js beschäftigt und gelernt, wie dieser funktioniert.

## **1.11 <2016-10-04>**

### **1.12 Gemacht**

Ich habe weiter an der Upload Funktion gearbeitet. Ich habe sowohl am Frontend und am Backend versucht, die nötigen methoden zu implementieren. Ich habe es jedoch nicht fertig geschafft.

### **1.13 Was Ich heute gelernt habe**

Ich habe heute mehr über HTML-Forms und Ajax in AngularJS gelernt.

### 1.14 <2016-11-15>

Ich habe die Video funktion zur Seite hinzugefügt und die Audio Funktion verbessert. Auch habe ich angefangen, eine Dokumentation der Themen zu schreiben.

## 2 Dokumentierung der Themen

### 2.1 Bilder

#### 2.1.1 Bildformate

##### 1. Rastergrafiken

jpg	Bildformat mit verlustbehafteter Kompression. Kommt oft bei Photographien zum Einsatz. (Farbtiefe meist 24 bit, jedoch nicht spezifiziert)
png	Bildformat mit verlustfreien Kompression. Kommt hauptsächlich bei Grafiken zu Einsatz. (Die Farbtiefe ist 24 bit. Zusätzlich gibt es einen 8 bit Alpha-Kanal für Transparenz)
gif	Veraltetes Bildformat mit verlustfreien Kompression. Es können mit GIF animationen gemacht werden. (Die Farbtiefe ist 8 bit und es ist ein 1 bit Alpha-Kanal vorhanden)

Es gibt weitere Formate, welche historisch eine Rolle spielten, heute jedoch kaum mehr eingesetzt werden. Beispielsweise:

- **.bmp** Windows Bitmap
- **.xbm** X11 Bitmap (1 bit)
- **.xpm** X11 Bitmap (8 bit)

##### 2. Vektorgrafiken

Vektorgrafiken werden für Grafiken eingesetzt. Im Gegensatz zu Rastergrafiken werden hier nicht pixel, sondern geometrische Objekte respektive deren Vektoren gespeichert und beim Rendering in eine Rastergrafik umgewandelt. Dies macht sie unendlich skalierbar.

Es gibt mehrere Formate in welchen Vektorgrafiken gespeichert werden können:

- **.ps** PostScript
- **.pdf** Portable Document Format
- **.svg** Scalable Vector Graphic

#### 2.1.2 Downloaddauer einer Datei

Die Downloaddauer einer Datei ist von 2 Faktoren abhängig:

- Grösse der Datei
- Geschwindigkeit des Netzwerkes

Beispiele:

- Bei einer 5 MBit/s Verbindung ist eine 100 MiB grosse Datei in etwa 2.5 Minuten heruntergeladen
- Bei einer 100 MBit/s Verbindung ist eine 800 KiB grosse Datei in etwa einer zehntel Sekunde heruntergeladen

### 2.1.3 Auflösung und Farbtiefe

#### 1. Auflösung

Die Auflösung eines Bildes ist von grosser Wichtigkeit um eine gute ausgabe zu garantieren, sowohl im Druck als auch im Web.

Im Druck wird generell eine höhere Auflösung benötigt, da ein Drucker ein genaueres Resultat produzieren kann als ein Bildschirm.

Mindestwerte:

Druck	300 dpi
Web	144 dpi - 200 dpi

**dpi** bedeutet *dots per inch*, also punkte pro inch. Ebenfalls gibt es den Begriff **ppi** welcher *Pixels per inch* bedeutet und - wie der name schon sagt - bei Bildschirmen und digitalen Inhalten zur Anwendung kommen.

#### 2. Farbtiefe

Die Farbtiefe definiert, wie viele Farben in einem Pixel eines Bildes gespeichert werden können. Die Farbtiefe wird normalerweise in bit-Grösse angegeben. Beispiele:

1 bit	Monochrom (Schwarz Weiss)
8 bit	Pro Farbe 8 bit (255 Abstufungen) also $256^3$ Farben

#### 3. Farbräume

Es gibt in der grafischen Industrie 2 grundlegende Weisen Farben zu speichern:

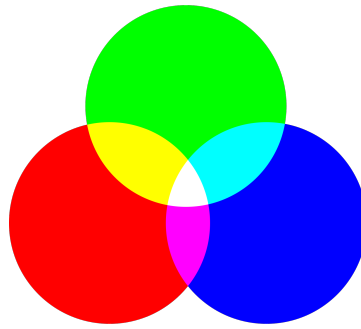
- CYMK für Druck
- RGB(A) für das Web

CYMK steht für die Farben *Cyan, Yellow, Magenta, Key (Schwarz)*, welche normalerweise im Druck verwendet werden.

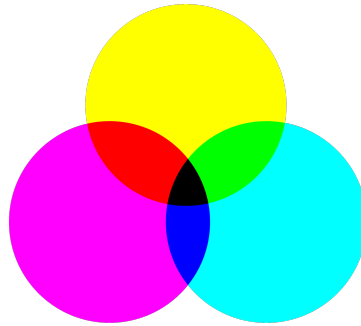
RGB steht für *Red o. Rot, Green o. Grün und Blue o. Blau*. Diese Farben (mit unterschiedlichen tönen) stehen einem Bildschirm zur verfügung, warum Grafiken für das Web normalerweise in RGB gespeichert sind.

Diese beiden Farbdefinitionen decken nicht genau die selben Farben ab, sodass es bei der kovertierung zwischen den beiden Modellen zu Problemen kommen kann (Bspw. kann eine Verfärbung entstehen)

Wie sich die Farbräume verhalten, wird in den folgenden Abbildungen gezeigt:



**RGB**



**CMYK**

#### **2.1.4 Bildergalerie**

Es gibt für das WWW viele Möglichkeiten, Bilder darzustellen, beispielsweise in einer Galerie. Es gibt u.A. PHP Frameworks zur Darstellung von Bildern. Am geläufigsten sind heutzutage (2016) Bildergalerien welche mit JavaScript verwirklicht wurden. Dafür gibt es verschiedene Libraries.

## **2.2 Juristische Fragen**

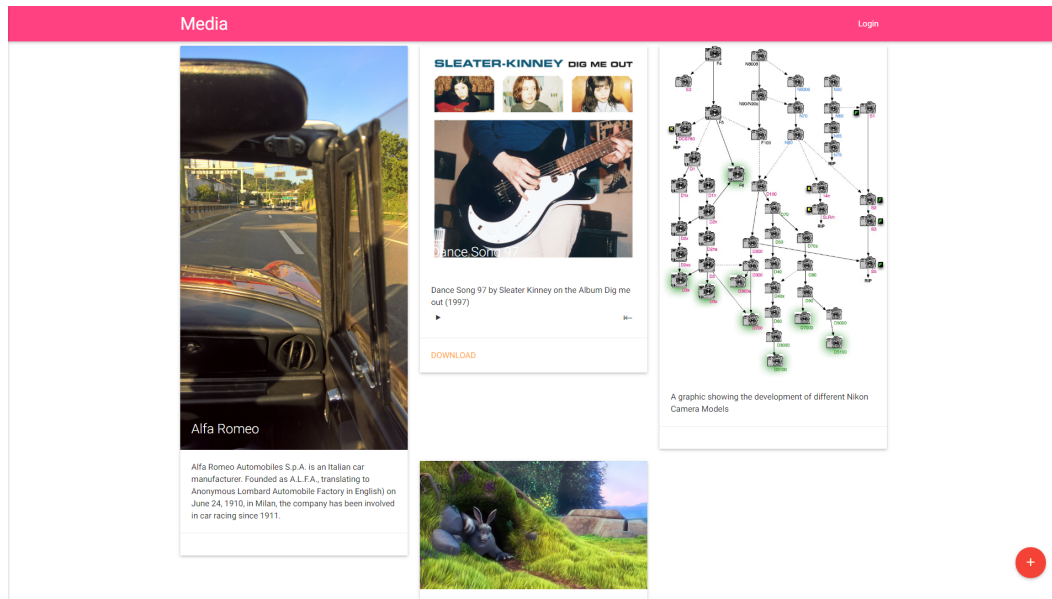
Wenn in Betracht gezogen wird, Inhalte aus dem Internet zu verwenden, muss jeweils die rechtliche Lage der Inhalt geprüft werden um sich zu vergewissern, ob diese überhaupt verwendet werden dürfen.

Es gibt verschiedene Varianten, wie Inhalte im Internet lizenziert werden:

- Proprietär
- Freie Lizenzen
- Public Domain

### **2.2.1 Proprietäre Inhalte**

Proprietäre Inhalte gehören explizit dem o. der AutorIn oder der Firma an welche der o. die AutorIn seine o. ihre Arbeitskraft verkauft. Die Nutzung ohne Rücksprache derer welche die sogn. Urheberrechte besitzen, ist illegal. Oft können die Rechte diese Inhalte selber weiterzuverwenden, gekauft werden.



Screenshot einer Bildergalerie

## 2.2.2 Freie Lizenzen

Es gibt eine Reihe von freien Lizenzen welche der o. die AutorIn explizit erlauben, die Inhalte weiterzuverwenden, jedoch unter bestimmten Bedingungen. Auch sind AutorInnen rechtlich geschützt.

Je nach Lizenz, können bestimmte Bestimmungen getroffen werden:

- Darf der Inhalt verändert werden?
- Darf mit dem Inhalt Profit erwirtschaftet werden?
- Unter welcher Lizenz darf der Inhalt geteilt werden?

Für unterschiedliche Inhalte gibt es verschiedene Lizenzen:

### 1. Bilder, Audio, Video

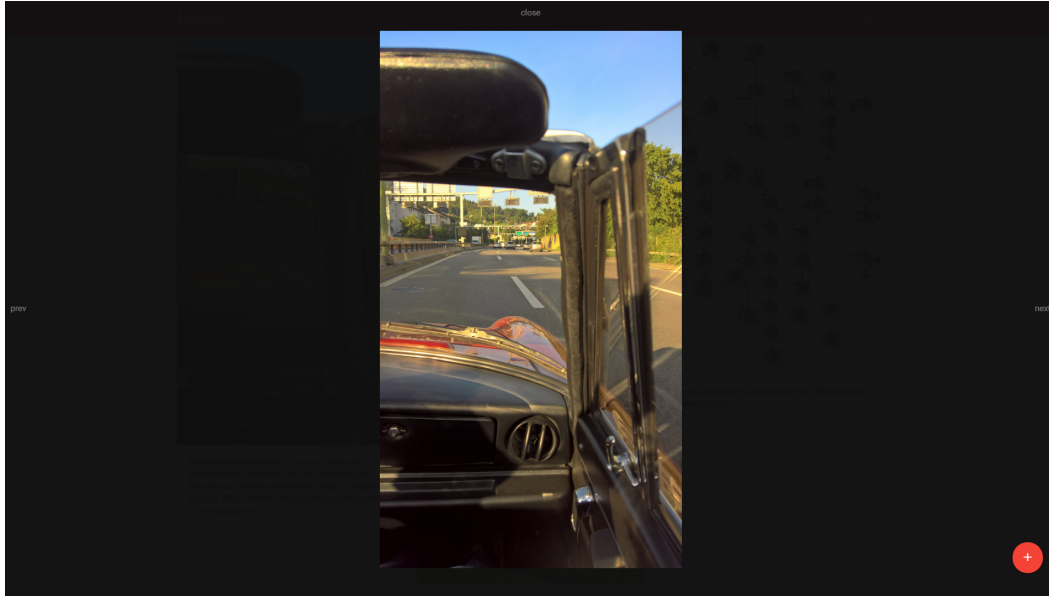
Bei Multimedia Inhalte sind vor allem die Creative Commons Lizenzen <sup>1</sup> verbreitet. Die Creative Common gibt es in verschiedene Varianten welche aus verschiedenen Attributen beliebig zusammengesetzt werden können:

- CC: Kennzeichnung, dass es sich um die Creative Commons Lizenz handelt
- BY: AutorIn muss angegeben werden
- NC: (Non Commercial) Es dürfen keine Profite erwirtschaftet werden
- SA: (Share Alike) Es dürfen veränderung am Inhalt gemacht werden. Der neue Inhalt muss unter den gleichen Bedingungen / Lizenz veröffentlicht werden
- ND: (No Derivates) Es dürfen keine Veränderungen vorgenommen werden

Beispiel: *CC BY NC SA*

---

<sup>1</sup><https://creativecommons.org/>



Screenshot eines geöffneten Bildes in einer Bildergalerie (Die Librarie wurde vom Autor dieses Dokumentes entwickelt)

## 2. Computer Programme & Source Code

Für Computer Programme existieren ebenfalls spezielle Lizenzen welche vor allem die Verfügbarkeit des Source Codes regeln. Die Details sind in der folgenden Tabelle beschrieben (welche natürlich keinen Anspruch auf vollständigkeit hat)

GNU GPL	Die GNU GPL (Gnu's Not Unix General Public License) ist die älteste Lizenz für freie Software und eine der Radikalsten. Sie wurde von einem der Pioniere der Free Software Movement, Richard Stallman geschrieben. Wie bei allen Lizenzen für freie Software muss jede Änderung am Sourcecode der Öffentlichkeit zugänglich gemacht werden. Die GNU GPL hat zusätzlich ein Copyleft, was festlegt, dass alle Veränderungen zwingenderweise unter der GNU GPL oder einer kompatiblen Lizenz veröffentlicht werden muss. (Die Kompatibilität wird von der FSF (Free Software Foundation) festgelegt). <sup>2</sup>
BSD	Die BSD Lizenz wurde ursprünglich für das Betriebssystem Berkeley Software Distribution (UNIX Distribution von der University of California Berkeley) geschrieben. Im Unterschied zur GNU GPL besitzt die BSD Lizenz kein Copyleft, was bedeutet, dass der Code für Proprietäre Software verwendet werden kann. <sup>3</sup>
Apache	Die Apache Lizenz wurde für den Apache Webserver geschrieben. Sie ist sehr ähnlich wie die BSD Lizenz und wird oft für Web Projekte (Beispielsweise JavaScript Libraries). <sup>4</sup>

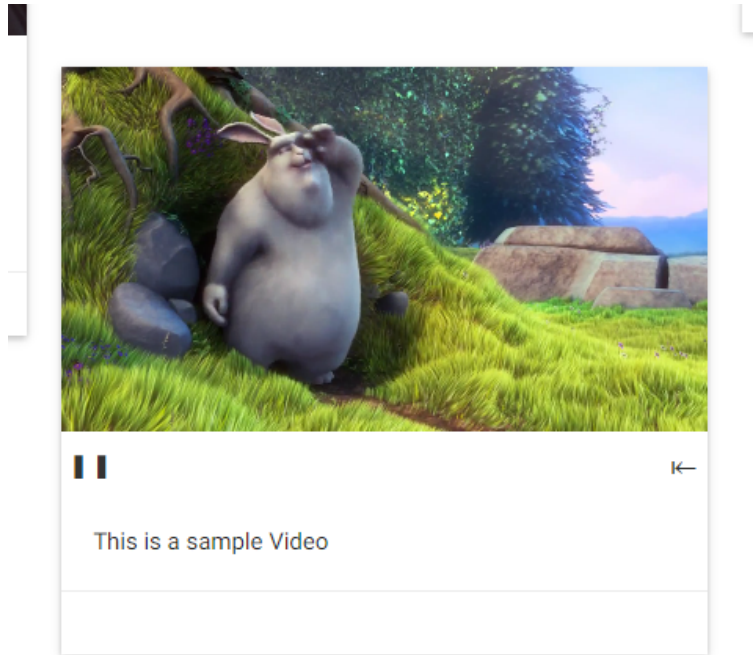
## 3. Dokumente

Es gibt spezielle Lizenzen für Dokumente jeglicher Art. Grundsätzlich können Dokumente unter den Bedingungen der Creative Commons Lizenz veröffentlicht werden. Auch gibt es

<sup>2</sup><http://gnu.org>

<sup>3</sup><https://opensource.org/licenses/BSD-3-Clause>

<sup>4</sup><https://www.apache.org/licenses/LICENSE-2.0>



Screenshot eines in die Website eingebundenen Video

die *GNU Free Documentation License* <sup>2</sup>, welche es erlaubt, Dokumente unter GNU GPL ähnlichen Bedingungen zu veröffentlichen.

## 2.3 Banner

### 2.3.1 Verwendung

Banner koennen verschieden eingesetzt werden, hauptsaechlich jedoch um Inhalt welcher nicht direkt mit dem Inhalt des eigentlichen Dokumentes zusammenhaengt. Vor allem fuer Werbung auf Websites werden Banner verwendet.

### 2.3.2 Beispiele

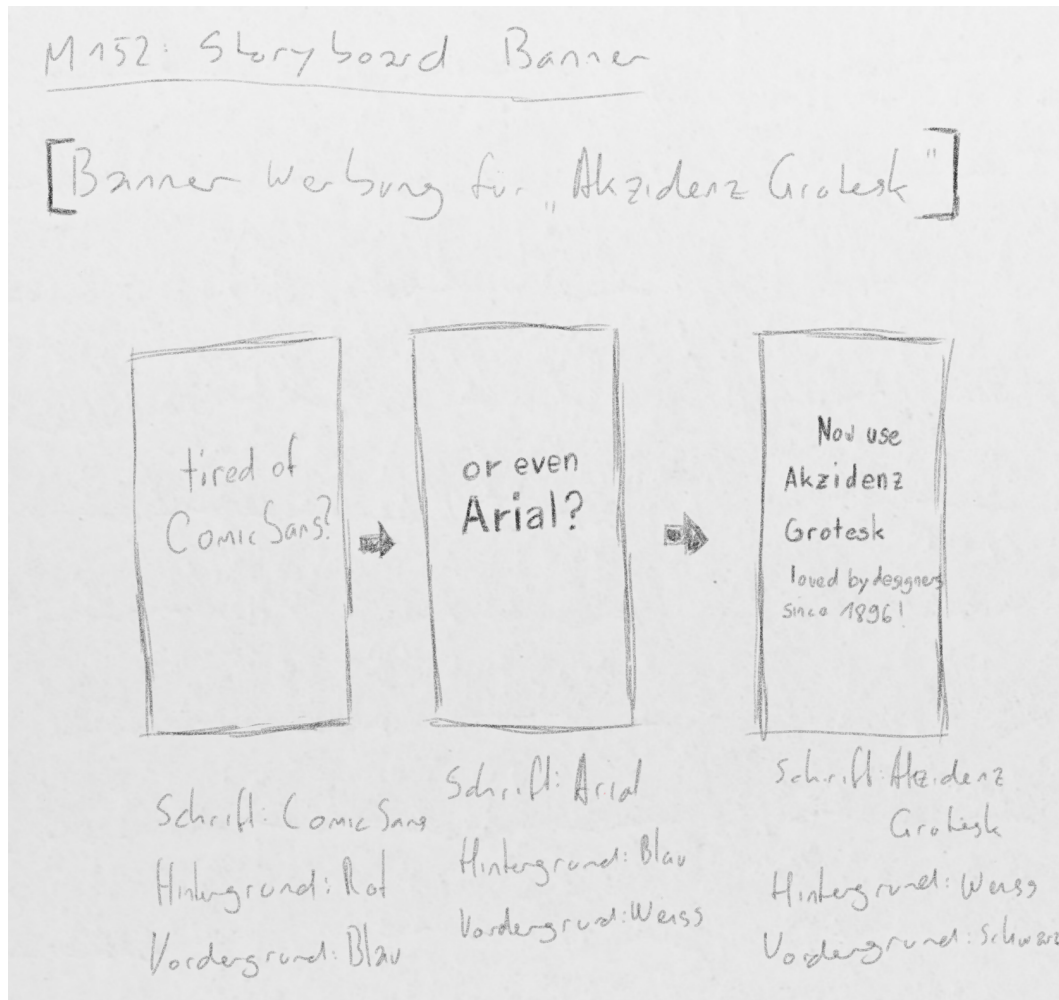
Es gibt verschiedene Anbieter für Online Werbung, der Grösste ist AdSense von Google. AdSense stellt folgende Grössen zur Verfügung:

- Medium Rectangle (300 x 250)
- Large Rectangle (336 x 280)
- Leaderboard (728 x 90)
- Large Skyscraper (300 x 600)
- Banner (468 x 60)

### 2.3.3 Implementierung in diesem Projekt

1. Storyboard Für die Planung eines Banners ist ein Storyboard hilfreich um eine





Storyboard

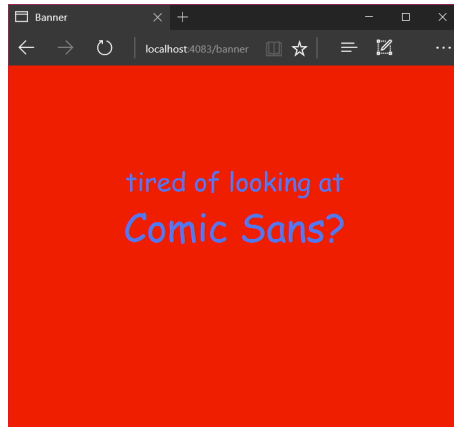
## 2. Implementierung

Die Implementierung der Banner kann mit verschiedenen Methoden erfolgen. Im folgenden Beispiel wurde HTML 5 verwendet:

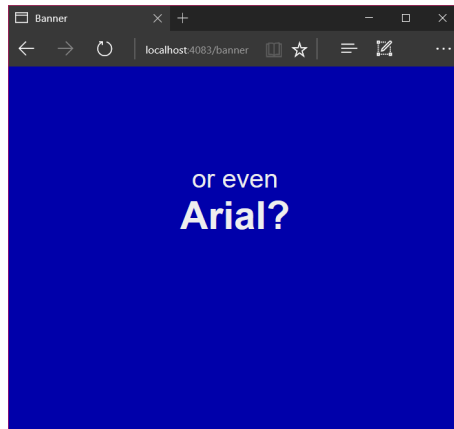
## 3 Über dieses Dokument

(c) 2016 von Patrick Günthard

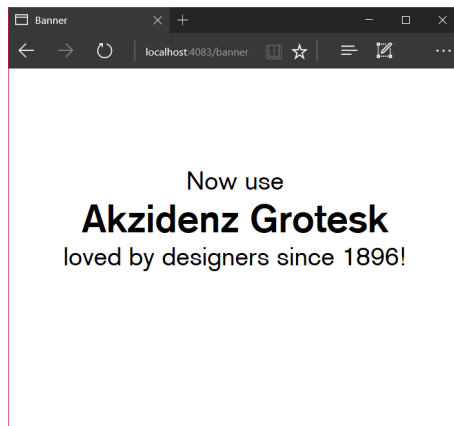
Dieses Dokument ist lizenziert unter der GNU Free Documentation License. Mehr Informationen sind auf <http://gnu.org> zu finden.



Screen 1



Screen 2



Screen 3