

The background of the book cover is a photograph of a fisherman in silhouette, standing on a boat and holding a large, circular net. The net is spread out, creating a complex, grid-like pattern that fills the upper half of the image. The lighting is warm, suggesting a sunset or sunrise, with orange and yellow hues reflecting on the water and the net. The fisherman is positioned in the lower right, looking towards the net.

**Autor des
Bestsellers**
*„HTML5
Handbuch“*

Clemens Gull

BigData

mit JavaScript visualisieren

D3.js für die Darstellung
großer Datenmengen einsetzen

Clemens Gull

BigData
mit JavaScript visualisieren

Clemens Gull

BigData

mit JavaScript visualisieren

D3.js für die Darstellung
großer Datenmengen einsetzen

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2014 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Programmleitung: Dr. Markus Stäuble

Herausgeber: Ulrich Dorn

Satz: DTP-Satz A. Kugge, München

art & design: www.ideehoch2.de

Druck: C.H. Beck, Nördlingen

Printed in Germany

ISBN 978-3-645-60347-8

Vorwort

Aus Gründen der einfacheren Lesbarkeit wird im gesamten Buch auf die geschlechts-spezifische Differenzierung wie beispielsweise Anwender/-innen verzichtet. Entsprechende Begriffe gelten in meinem Sinne und im Sinne der Gleichbehandlung grundsätzlich für beide Geschlechter.

Worum geht es in diesem Buch?

Dieses Buch ist kein klassisches »Lernbuch«, wie wir es dutzendfach kennen. Aber es ist auch weder eine Referenz noch ein Kompendium für einen Datenbestand oder eine bestimmte Programmiersprache. Es ist eine praktische Einführung in das Thema Open Data und die Visualisierung dieser Daten mit der Bibliothek `3D.js` im World Wide Web. Das Buch ist in drei Teile gegliedert.

- Am Anfang steht eine Einführung in die Bibliothek `D3.js`.
- Der zweite Teil befasst sich mit der Visualisierung von Daten und der Interaktion mit diesen.
- Der dritte Teil erklärt den Begriff »Open Data« und beschäftigt sich mit einer kompletten Anwendung, die wir mit dem Wissen aus den ersten Abschnitten erstellen.

Ein Glossar am Ende des Buches bietet noch mal einfache Erläuterungen zu den verschiedenen Fachbegriffe. Im Index sind die wichtigsten Begriffe zusammengefasst, damit Sie sie schnell nachschlagen können.

Was wird für dieses Buch benötigt?

Natürlich sollten Sie einen Computer nutzen können, der ein fehlerfrei installiertes Betriebssystem hat. Ob Sie Microsoft Windows, Mac OS X oder eine Unix-Variante einsetzen, ist für die Arbeit mit diesem Buch nicht von Belang. Aber denken Sie bitte daran: Sie installieren Software, die das System erweitert. Daher wäre es gut, wenn keine wichtigen Daten auf dem System gespeichert wären, falls etwas gravierend schiefgehen sollte. Zusätzlich benötigen Sie eine funktionierende Internetverbindung. Sie wird für die Installation der einzelnen Softwareteile verwendet. Aber sie kommt auch im letzten Teil dieses Buches zum Einsatz, um die Applikation entwickeln und testen zu können.

Download der Listings zum Buch

Besuchen Sie unsere Website unter <http://www.buch.cd> und geben Sie dort die letzten fünf Ziffern der ISBN dieses Buches samt Bindestrich ein, um alle Beispielcodes und sonstigen Ressourcen zu diesem Buch herunterzuladen. Die verfügbaren Dateien werden nach der erfolgreichen Anmeldung angezeigt.

Berichtigungen, Änderungen, Verbesserungen

Obwohl alle Beteiligten mit größter Sorgfalt vorgehen, um die Richtigkeit der Inhalte sicherzustellen, passieren Fehler. Sollten Sie einen Fehler in diesem Buch entdecken, ob im Text oder im Quellcode, bin ich sehr dankbar, wenn ich eine Mitteilung erhalte. So können Sie anderen Lesern Ärger ersparen und mithelfen, die nachfolgende Version des Buches zu verbessern. Wenn Sie also irgendeinen Druckfehler finden, teilen Sie ihn mir bitte per E-Mail an buch@guru-20.info mit. Ich werde alle Berichtigungen, Änderungen und Verbesserungen auf meinem Blog <http://www.guru-20.info> veröffentlichen.

Gute Voraussetzungen

Für den erfolgreichen Einsatz dieses Buches benötigen Sie zumindest Grundkenntnisse im Erstellen von Webseiten. Sie werden einfacher und schneller Lösungen erarbeiten können, wenn Sie HTML und CSS schon verwendet haben und auch einsetzen können. Es hilft ebenfalls, wenn Sie bereits Grundkenntnisse im Programmieren – idealerweise mit JavaScript – mitbringen. Aber eine absolute Voraussetzung ist das nicht. Es werden immer wieder Fachbegriffe im Text auftauchen.

Danke für die Unterstützung durch das Team des Franzis Verlags.

Ich würde mich über Reaktionen und Anregungen sehr freuen. Sie erreichen mich unter der Adresse gull@guru-20.info.

Ihr

Clemens Gull

Für den Menschen, der
mich immer wieder
ungeheuerlich motiviert!

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Datendarstellung im Web mit D3 | 13 |
| 1.1 | Eigenheiten und Syntaxregeln | 14 |
| 1.1.1 | Auswahl von Elementen im DOM | 14 |
| 1.1.2 | Dynamische Eigenschaften | 15 |
| 1.1.3 | Enter und Exit | 17 |
| 1.1.4 | Transitions | 18 |
| 1.2 | Plug-ins von D3.js | 18 |
| 1.3 | Das Document Object Model | 19 |
| 1.3.1 | Beziehungen von Knoten | 20 |
| 1.3.2 | Arten von Knoten | 21 |
| 1.3.3 | Verarbeitung von Knoten | 21 |
| 2 | Visualisierung von Daten | 23 |
| 2.1 | Vorbereitung für die Übungen | 23 |
| 2.2 | Erstellen eines Balkendiagramms | 26 |
| 2.2.1 | Die Daten für das Diagramm | 26 |
| 2.2.2 | Das Balkendiagramm automatisch erstellen | 28 |
| 2.2.3 | Was haben wir gelernt? | 33 |
| 2.3 | Erstellen eines Kreisdiagramms | 40 |
| 2.3.1 | Die Daten für das Diagramm | 42 |
| 2.3.2 | Das Kreisdiagramm automatisch erstellen | 42 |
| 2.3.3 | Was haben wir gelernt? | 46 |
| 2.4 | Animierte Diagramme | 52 |
| 2.4.1 | Kreisdiagramm für Zugriffszahlen von zwei Halbjahren | 52 |
| 2.4.2 | Die Daten für das Diagramm | 54 |
| 2.4.3 | Das Kreisdiagramm erstellen | 54 |
| 2.4.4 | Die Daten für das Diagramm einlesen | 55 |
| 2.4.5 | Die Animation für das Diagramm einfügen | 58 |
| 2.4.6 | Mit fehlenden Daten arbeiten | 59 |
| 2.4.7 | Die Segmente mit einem Titel versehen | 62 |
| 2.4.8 | Was haben wir gelernt? | 63 |
| 2.5 | Exkurs: Open Data | 71 |
| 3 | Der Métro Information Tracker | 73 |
| 3.1 | Grundlegende Software | 73 |
| 3.1.1 | Lokale Installation unter Mac OS X | 73 |

| | | |
|-------|--|-----|
| 3.1.2 | Lokale Installation unter Windows | 83 |
| 3.1.3 | Test der lokalen Installation | 88 |
| 3.1.4 | Die Entwicklungsumgebung | 91 |
| 3.2 | Was ist der MIT? | 102 |
| 3.2.1 | Die Daten der Pariser Métro | 103 |
| 3.3 | Darstellung als Force-Directed-Graph | 106 |
| 3.3.1 | Vorbereitung des Projekts | 107 |
| 3.3.2 | Das Stylesheet erstellen | 108 |
| 3.3.3 | Die Grundstruktur für die HTML-Datei des MIT | 112 |
| 3.3.4 | Einlesen der Daten und Erstellen einer Datenstruktur | 114 |
| 3.3.5 | Anzeigen des Graphen mit den vorhandenen Daten | 117 |
| 3.3.6 | Die Datenpunkte anklickbar machen | 132 |
| 3.4 | Anpassen der Graph-Darstellung | 139 |
| 3.4.1 | Die Gewichtung der Datenpunkte dynamisch ändern | 139 |
| 3.4.2 | Die Knoten durchsuchbar machen | 150 |
| 3.5 | Force-Directed-Graph als Baum darstellen | 154 |
| 3.5.1 | Anpassen des Stylesheets | 155 |
| 3.5.2 | Anpassen der Startseite | 156 |
| 3.5.3 | Erstellen der Anwendung | 158 |
| 3.5.4 | Die JSON-Daten einlesen und verarbeiten | 161 |
| 3.5.5 | Die Datenstruktur als Baumdiagramm darstellen | 170 |
| 3.5.6 | Die Anwendung initialisieren | 185 |
| 3.5.7 | Einen Knoten im Diagramm suchen | 186 |
| 3.5.8 | Zusätzliche Funktionen zur Unterstützung der Anwendung | 188 |
| 3.5.9 | Was haben wir gelernt? | 192 |
| 3.6 | Grafiken für die Knoten verwenden | 193 |
| 3.6.1 | Scalable Vector Graphics im Überblick | 194 |
| 3.6.2 | Den Métro Information Tracker mit Grafiken erweitern | 198 |
| 3.7 | Anpassung am Userinterface vornehmen | 200 |
| 3.7.1 | Die CSS-Datei verändern | 201 |
| 3.7.2 | Eine neue Symbolleiste in die HTML-Datei einfügen | 202 |
| 3.7.3 | Die Funktion zum Ein-/Ausblenden der Knoten anpassen | 203 |
| 3.8 | Métrolinien geografisch darstellen und nach Fahrgastzahlen gewichten | 204 |
| 3.8.1 | Die Daten für Version 3 des MIT | 205 |
| 3.8.2 | HTML-Datei für die Version 3 des MIT erstellen | 207 |
| 3.8.3 | Die Stylesheets des MIT von Version 3 | 210 |
| 3.8.4 | Das JavaScript für den MIT von Version 3 | 218 |
| 3.8.5 | Was haben wir gelernt? | 233 |
| 3.9 | Die Métrolinie um ein Widget und einen Tooltipp ergänzen | 236 |

| | | |
|----------|-----------------------------------|------------|
| A | Anhang | 251 |
| A.1 | Probleme und Lösungen | 251 |
| A.1.1 | MAMP startet nicht | 251 |
| A.2 | Glossar | 255 |
| A.3 | Verzeichnis der Listings | 260 |
| | Stichwortverzeichnis | 265 |



Visualisierung von Daten

Meiner Meinung nach hat lernen etwas mit machen zu tun. Daher ist es am besten, wenn wir sofort ins kalte Wasser springen und versuchen, Daten zu visualisieren und die Bibliothek `d3.js` einzusetzen. Natürlich werden wir nicht mit dem schwierigsten beginnen, sondern uns langsam vorarbeiten.

Für das Arbeiten mit Webseiten benötigen wir eine Entwicklungsumgebung. Welche Sie dazu einsetzen, bleibt vollständig Ihnen überlassen. Ich verwende gern Aptana Studio³ oder auch Eclipse⁴. Denn diese beiden Produkte sind einerseits kostenlos im Web verfügbar und andererseits auf den beiden großen Betriebssystemen Microsoft Windows und Mac OS X analog zu bedienen.

2.1 Vorbereitung für die Übungen

Bevor Sie mit dem Programmieren beginnen, benötigen Sie noch eine Kleinigkeit.

Natürlich können wir mit einer Internetverbindung immer auf die aktuelle Bibliothek von `d3.js` zurückgreifen, es ist für Sie im Moment jedoch einfach praktischer, alle Teile der Anwendung lokal auf Ihrem Computer zu haben. Dadurch sind Sie erstens

³ Online verfügbar unter <http://www.aptana.com>

⁴ Online verfügbar unter <http://www.eclipse.org>

schneller, da Sie keine Internetverbindung nutzen müssen, und zweitens sind Sie unabhängig, denn wenn die Netzverbindung nicht funktioniert, können Sie trotzdem weiterarbeiten. Daher sollten Sie sich jetzt die Bibliothek unter der Webadresse <http://d3js.org> herunterladen.

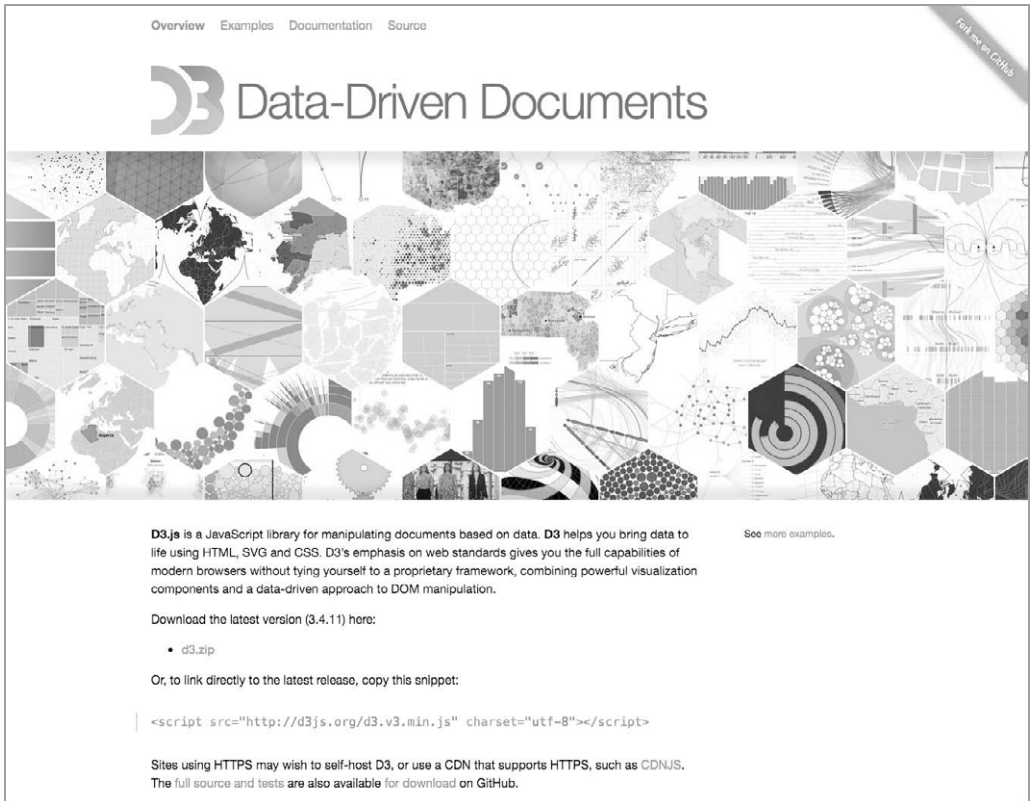


Bild 2.1: Download der Bibliothek unter <http://d3js.org>.

Auch ist jetzt der Zeitpunkt gekommen, eine Verzeichnisstruktur auf der Festplatte zu erstellen, damit Sie alle einzelnen Übungen einfach und schnell anlegen und durchführen können. Wie in der folgenden Abbildung zu sehen ist, benötigen wir vier Unterverzeichnisse:

```
/css
/data
/images
/js
```

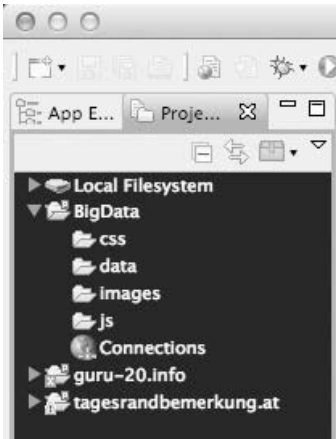


Bild 2.2: Die Verzeichnisstruktur für die Übungen zu D3.

Nachdem Sie die Bibliothek heruntergeladen haben, müssen Sie das Archiv entpacken und den Inhalt in das Verzeichnis `/js` kopieren. Damit sollte die Struktur des Projekts wie in der folgenden Abbildung aussehen.

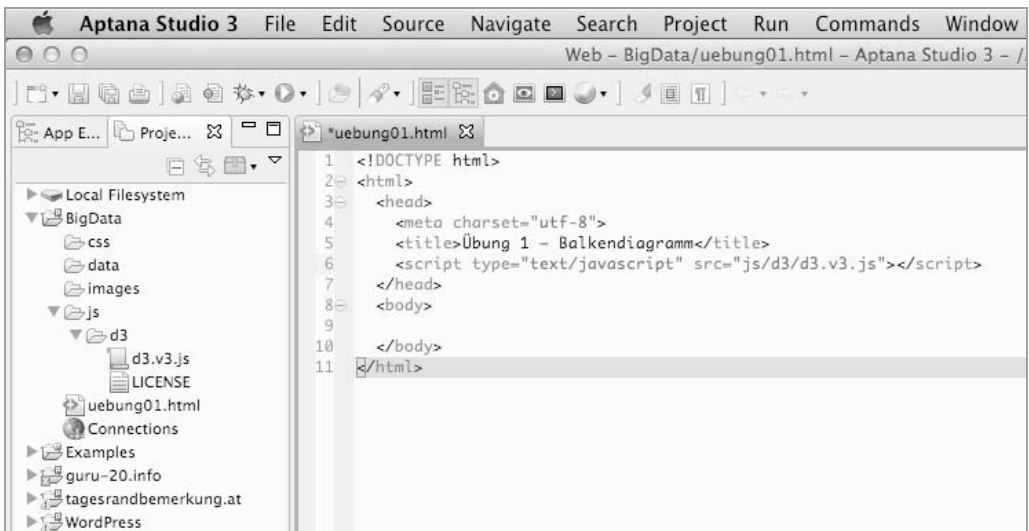


Bild 2.3: Grundstruktur des HTML-Dokuments und des Übungsprojekts.

Nun können wir mit der ersten Übung beginnen.



Der Métro Information Tracker

Unsere große Applikation, die wir jetzt erarbeitet, ist der *Métro Information Tracker* oder kurz MIT. Diese Applikation wird mit lokalen Daten und Daten aus der Open-Data-Initiative Informationen zu den Métrostationen von Paris darstellen. Dazu werden Sie die verschiedenen Diagrammarten von D3 kennenlernen und diese auch erstellen. Mit der Interaktion des Benutzers können dann weitere Informationen abgerufen werden.

3.1 Grundlegende Software

Bevor wir mit der Applikation und besonders mit der Programmierung derselben beginnen können, muss der Computer mit grundlegender Software darauf vorbereitet werden.

3.1.1 Lokale Installation unter Mac OS X

Prinzipiell hat dieses Betriebssystem die notwendigen Komponenten bereits integriert, und sie müssten nur aktiviert werden. Da dies jedoch einiges an Fachwissen über Webserver und Konfiguration voraussetzt, wird ein fertiges Paket – zusätzlich zu den vorhandenen Komponenten – installiert.

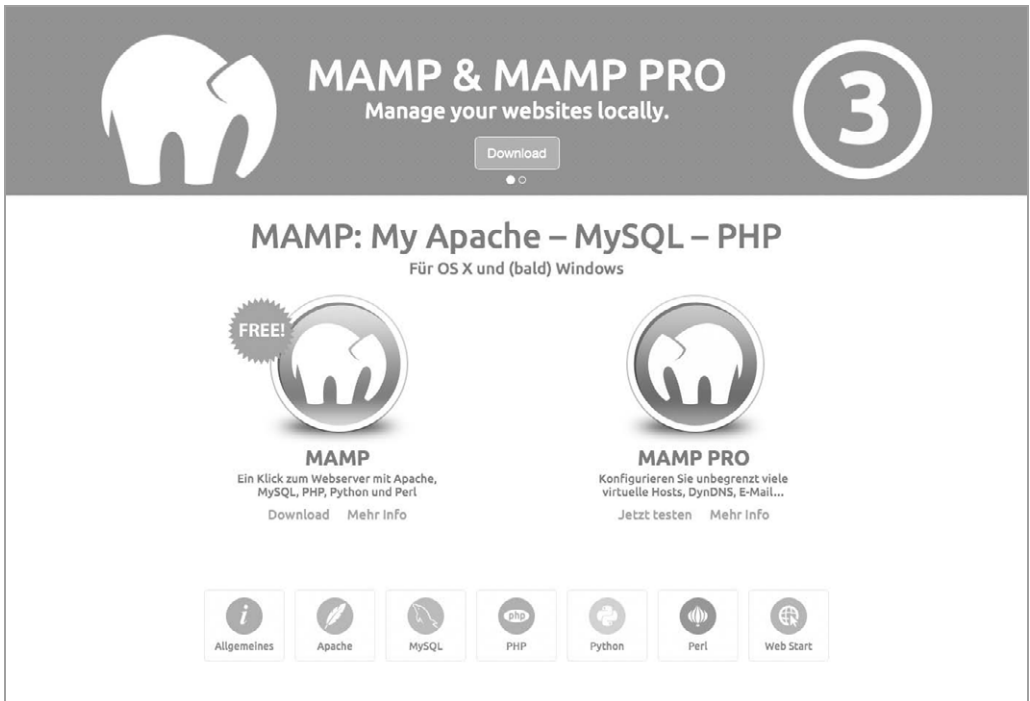


Bild 3.1: Für Mac OS X und bald auch Windows – der Download von MAMP & MAMP PRO 3.0.5 unter <http://www.mamp.info/de/downloads/>.

Für die Installation notwendige Software

Die einfachste Methode für die Installation ist das bereits fertig zusammengestellte Paket MAMP. Die Abkürzung setzt sich aus den Anfangsbuchstaben der enthaltenen einzelnen Anwendungen zusammen.

- M – für das Betriebssystem Mac OS X
- A – für den Webserver Apache
- M – für das Datenbanksystem MySQL
- P – für PHP

Für die Systeme Mac OS X 10.6 und höher ist das Paket MAMP & MAMP PRO 3.0.5 unter <http://www.mamp.info/de/downloads/> zu finden.

Sie erhalten beide Dateien auch auf der Webseite zum Buch unter <http://www.buch.cd>, wie auf Seite 6 unter »Download der Listings zum Buch« beschrieben.



Anhang

A.1 Probleme und Lösungen

A.1.1 MAMP startet nicht

Wenn nach der Installation von MAMP auf Mac OS X Probleme auftreten, ist meistens eine der beiden folgenden Lösungen hilfreich, um trotzdem damit arbeiten zu können.

Der Webserver startet nicht

Häufig ist das der Fall, wenn bereits ein Webserver auf Mac OS X installiert ist. Auf den ersten Blick können Sie dies nicht erkennen, aber nach wenigen Klicks ist es offensichtlich. Öffnen Sie dazu über das Apfel-Menü die *Systemeinstellungen*.

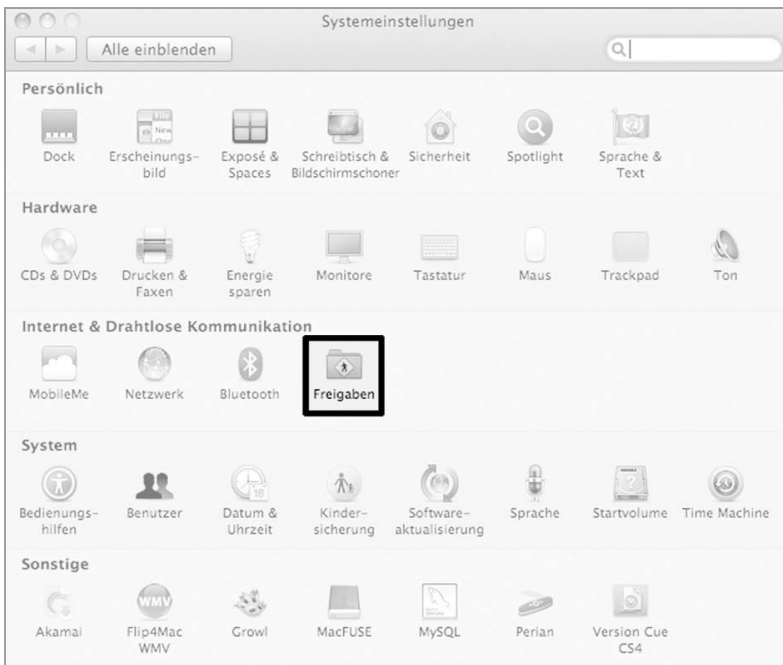


Bild A.1: Systemeinstellungen von Mac OS X.

Öffnen Sie nun den Bereich *Freigaben*. Dort sehen Sie alle Freigaben des Systems auf Ihrem Monitor.



Bild A.2: Freigaben von Mac OS X.

Wenn in der Liste der Freigaben beim Eintrag *Webfreigabe* ein Häkchen gesetzt ist, dann ist der interne Webserver von Mac OS X aktiv. Entfernen Sie das Häkchen mit einem Klick. Sobald die Systemeinstellungen geschlossen sind, können Sie MAMP wie auf Seite 81 beschrieben starten.

Der Datenbankserver startet nicht

Wenn in der Steuerungszentrale von MAMP die Ampel bei *MySQL* nicht auf Grün springt, ist wahrscheinlich bereits ein Datenbankserver auf Ihrem System aktiv.

Um das zu überprüfen, öffnen Sie im Apfel-Menü die *Systemeinstellungen*. Überprüfen Sie im Bereich *Sonstige* am unteren Rand des Fensters, ob Sie ein Symbol *MySQL* sehen. Ist das der Fall, klicken Sie einmal darauf, um das Befehlsfenster des Datenbankservers zu öffnen.

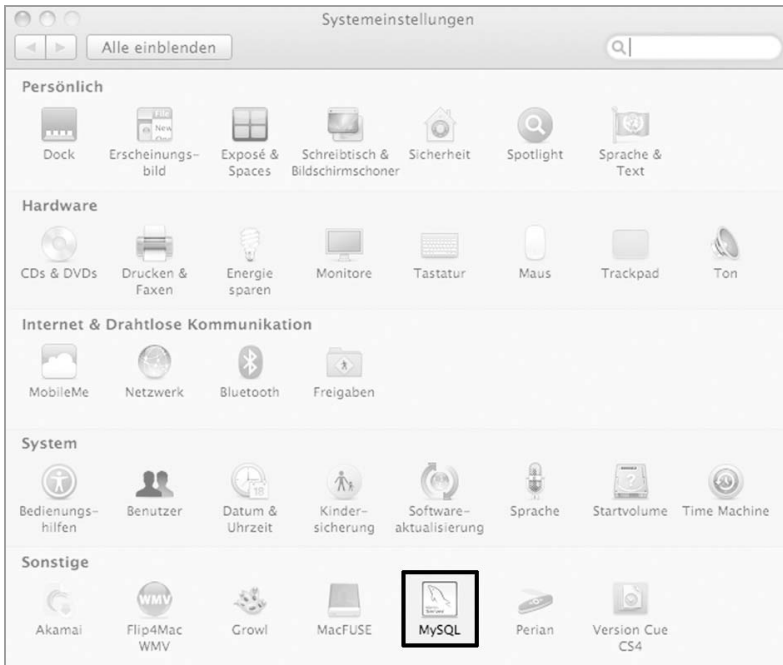


Bild A.3: Systemeinstellungen mit installiertem MySQL.

Im Befehlsfenster wird der Status des Datenbankservers angezeigt. Normalerweise sollte er auf *running* stehen. Mit dem Button *Stop MySQL Server* rechts vom Status können Sie den aktiven Datenbankserver anhalten.



Bild A.4: Status des MySQL-Servers.

Nachdem Sie den bereits installierten Datenbankserver angehalten haben, können Sie MAMP wie auf Seite 81 beschrieben in Betrieb nehmen.

A.2 Glossar

AJAX – Asynchronous JavaScript And XML

Ist ein Konzept zur asynchronen Datenübertragung zwischen einem Webbrowser und einem Webserver. Dadurch können weitere Anfragen über das Protokoll HTTP durchgeführt werden, ohne die gesamte Webseite neu laden zu müssen. Nach der Antwort können die Daten in die bestehende Webseite integriert werden. Dadurch entsteht für den Anwender der Eindruck einer dauerhaften Verbindung zwischen Webbrowser und Server.

API – Application Programming Interface

Ist eine Schnittstelle, die es dem Programmierer erlaubt, bestimmte Funktionen eines Betriebssystems oder einer anderen Anwendung zu nutzen, ohne genauere Kenntnisse über die Grundlagen der Software zu haben. Bei einem Betriebssystem sind dies etwa fehlende Kenntnisse über die Hardware, bei einer Software wäre es beispielsweise fehlendes Wissen über die Datenbankstruktur.

Array

Dies ist in der Informatik ein Datenfeld. Es enthält gleichartige Daten in einer definierten Struktur. Diese ergibt sich bei einem »normalen« Array aus den Indexzahlen, einer fortlaufenden Nummerierung. Bei einem assoziativen Array werden die Zahlen durch Begriffe ersetzt. Dadurch ist es für den Programmierer oft einfacher, diese doch komplexen Strukturen zu verarbeiten.

ASCII – American Standard Code for Information Interchange

Ist eine 7-Bit-Zeichencodierung und dient als Grundlage für spätere, auf mehr Bits basierenden Codierungen für Zeichensätze in der Informatik. Sie definiert 128 Zeichen, bestehend aus 32 nicht druckbaren sowie 96 druckbaren Zeichen. Letztere sind beginnend mit dem Leerzeichen, den Zahlen und den Buchstaben festgelegt worden. Diese Zeichen entsprechen weitgehend denen einer Tastatur für die englische Sprache. Die nicht druckbaren Steuerzeichen enthalten Ausgabezeichen wie Zeilenvorschub, Tabulator, Ende der Übertragung und Ähnliches. Durch die Internationalisierung der Informatik reicht dieser Zeichenvorrat aber nicht mehr aus und wurde infolgedessen durch weitere Definitionen (ANSI und UTF-8) ersetzt.

CSS – Cascading Style Sheet

Sind verschachtelte Gestaltungsvorlagen von strukturierten Dokumenten. Stylesheets werden vor allem zusammen mit HTML und XML eingesetzt. CSS gilt heute als der Standard für Webseiten.

Mit CSS wird die visuelle Darstellung (Farben, Layout, Typografie etc.) der Elemente einer HTML-Datei festgelegt. Mit CSS ist es möglich, für verschiedene Ausgabeme-

dien (Bildschirm, Papier, Sprache und noch viele andere) unterschiedliche Darstellungen vorzugeben.

CSV – Comma Separated Values

Ist ein sehr häufig verwendetes Datenaustauschformat. Bei diesem Format steht ein Datensatz in je einer Zeile. Die einzelnen Felder werden innerhalb der Zeile durch Kommata getrennt. Texte werden in Anführungsstrichen eingeschlossen, Zahlen hingegen nicht. Daher bildet auch das Komma kein Problem, da Zahlen mit Dezimalpunkt geschrieben werden.

DOM – Document Object Model

Ist eine Spezifikation des W3C einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Sie wird vom »World Wide Web Consortium« (W3C) definiert. Das DOM wird in HTML-Dokumenten im Wesentlichen benutzt, um es mit verschiedenen Programmiersprachen – hauptsächlich JavaScript – zu manipulieren und dem Anwender Interaktionen mit einer Webseite zu ermöglichen.

Event

Ist ein Ereignis, das in einem Computerprogramm auftritt. Meistens wird es durch den Anwender ausgelöst, beispielsweise durch einen Klick auf ein Element. Es kann aber auch von einem Programm selbst ausgelöst werden, zum Beispiel wenn ein Fehler im Ablauf auftritt. Auf diese Events reagiert das Programm.

FTP – File Transfer Protocol

Ist ein von den RFC 959 (<http://tools.ietf.org/html/rfc959>) und 1985 (<http://tools.ietf.org/html/rfc1985>) festgelegtes Netzwerkprotokoll. Es dient der Übertragung von Dateien in einem IP-Netzwerk. Für dieses Protokoll werden zwei unterschiedliche Verbindungen benötigt – ein Steuerkanal, über den die Befehle zwischen den beiden Geräten übertragen werden, und ein Datenkanal, über den die Dateien gesendet werden.

GET-Anfrage

Ist eine der Methoden, um Daten im World Wide Web zu übertragen. In diesem Fall werden die Schlüssel-Wert-Paare durch das Zeichen ? in der URL übergeben. Oft wird diese Vorgehensweise gewählt, um eine Liste von Parametern zu übertragen, die der Server bei der Bearbeitung einer Anfrage berücksichtigen soll. Häufig besteht diese Liste aus mehreren Paaren, die durch das &-Zeichen voneinander getrennt sind.

HTML – HyperText Markup Language

Ist eine textbasierte Auszeichnungssprache zur Strukturierung von digitalen Inhalten wie Texten, Bildern und Hyperlinks in elektronischen Dokumenten. Diese Dokumente sind die Grundlage des World Wide Web und können von Browsern optisch ansprechend dargestellt werden. Neben den Inhalten enthält HTML zusätzliche Angaben in

Form von Metainformationen, die auch über die im Text verwendete Sprache, den Autor, die Schlüsselwörter oder auch den Kurzzinhalt Auskunft geben.

HTML wird vom World Wide Web Consortium W3C weiterentwickelt, wobei offiziell die Version 4.01 freigegeben ist. HTML 5 befindet sich offiziell noch in der Entwicklung, wird jedoch bereits von fast allen Webbrowsern unterstützt.

HTTP – Hypertext Transfer Protocol

Dient der Übertragung von Daten in IP-Netzwerken. Heute wird das Protokoll hauptsächlich dazu eingesetzt, Webseiten im WWW zu übertragen. Dieses Protokoll ist zustandslos, daher ist auch keine dauerhafte Verbindung zum Webserver vorgesehen. Diese dauerhafte Verbindung muss dem Anwender mit sogenannten Sitzungsprotokollen vorgespielt werden.

Spezifiziert wurde das Protokoll in den RFC 1946 (<http://tools.ietf.org/html/rfc1946>) und 2616 (<http://tools.ietf.org/html/rfc2616>).

JSON – JavaScript Object Notation

Ist ein kompaktes Datenformat in für Mensch und Maschine einfach lesbarer Textform. Es dient im Web häufig als Grundlage zum Datenaustausch zwischen Server und Anwendungen. Jedes gültige JSON-Dokument sollte auch ein gültiges JavaScript sein und durch den Befehl `eval()` interpretiert werden können. Davon abgesehen ist JSON aber unabhängig von der Programmiersprache. Parser existieren in praktisch allen verbreiteten Sprachen. Der Standard ist im RFC 4627 festgelegt worden und kann unter <http://tools.ietf.org/html/rfc7159> abgerufen werden. Mit den RFC 7158 und 7159 wurde der Standard noch näher erläutert.

MIME-Type – Multipurpose Internet Mail Extensions

MIME ermöglicht es, zwischen Sender und Empfänger Informationen über den Typ der übermittelten Daten auszutauschen und gleichzeitig eine für den verwendeten Übertragungsweg sichere Zeichencodierung festzulegen.

Es sind Erweiterungen des RFC 822 (<http://tools.ietf.org/html/rfc822>) und des RFC 5322 (<http://tools.ietf.org/html/rfc5322>), die das Format von E-Mails definieren. Der Standard sieht nur den ASCII-Code für die Datenübertragung vor, dieser reichte am Anfang für englischsprachige Texte. Das brachte aber im Zuge der Internationalisierung des Netzes Probleme mit sich. MIME schafft daher Kompatibilität für zusätzliche Zeichen wie Umlaute sowie für Multimedia-Daten, etwa bei E-Mail-Anhängen. Definiert wurden sie in den RFC 2045 bis 2049.

Node

Als *Node*, zu Deutsch »Knoten«, werden innerhalb von Graphen die Datenpunkte bezeichnet. Im DOM ist es ein einzelnes Element, das in der Baumstruktur integriert ist. Der Basisknoten ist das `html`-Element, das die beiden Elemente `head` und `body` enthält. Dies sind die ersten drei Knoten, die eine HTML-Datei enthalten soll.

Bei JSON-Daten sind diese Daten genauso strukturiert. Auch hier sind die Informationen in einer Baumstruktur mit Knoten angeordnet.

Parser

Ist ein Programm, das für die Zerlegung und Umwandlung einer beliebigen Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist. Er gibt eine Eingabe in einer gewünschten Form aus und erzeugt zusätzlich Strukturbeschreibungen.

POST-Anfrage

Dies ist neben GET eine weitere Übertragungsmöglichkeit von Daten im World Wide Web. Da sich die Daten nicht in der URL befinden, können per POST große Datenmengen, wie Bilder oder Texte, übertragen werden. Dazu werden die Daten im `body`-Element mitgesendet.

RFC – Request for Comments

Bilden eine Sammlung von Richtlinien für die Kommunikation im Internet. Zunächst sind es noch keine fest definierten Standards, stattdessen werden sie als Vorschlag für eine Variante der Kommunikation veröffentlicht. Daran geknüpft ist die Bitte um Kommentare zum Vorschlag. Daraus leitet sich auch der Name ab. Ist der Vorschlag einmal technisch erfolgreich umgesetzt worden, wird er zum Standard erhoben.

SGML – Standard Generalized Markup Language

Ist eine 1986 unter der Norm ISO 8879 definierte Vorschrift für eine universelle Auszeichnungssprache. Diese Art von Computersprachen definieren Elemente auf Basis ihrer Bedeutung und arbeiten mit Tags und einer Baumstruktur. Davon abgeleitet wurden beispielsweise HTML oder auch XML.

SVG – Scalable Vector Graphics

Ist ein speziell für das World Wide Web brauchbares Dateiformat für Vektorgrafiken. Diese Grafiken lassen sich – anders als Pixel- bzw. Bitmapgrafiken – ohne Verluste in jede beliebige Größe bringen. Alle modernen Browser können SVG-Grafiken darstellen.

Tag

In Auszeichnungssprachen wie SGML, XML oder HTML bezeichnen Tags die in spitzen Klammern eingeschlossenen Kürzel, die Sie brauchen, um Elemente zu kennzeichnen und zu strukturieren. Meist kommen sogenannte öffnende und schließende Tags (mit einem führenden Schrägstrich) zum Einsatz.

Tags können immer nur paarweise auftreten. Auf diese Weise erzeugen sie eine Baumstruktur. In HTML existieren noch historisch bedingte Ausnahmen. Ein leeres Element `<div></div>` kann auch als `<div/>` geschrieben werden.

In HTML bedeutet zum Beispiel die Auszeichnung `<h1>MIT</h1>`, dass der Text eine Überschrift der Ebene 1 ist, also ein sehr wichtiger Text für diese Datei. Er wird auch von den Suchmaschinen als wichtiges Element betrachtet.

Tags können Attribute enthalten, in HTML zum Beispiel `id`, um das Element zu identifizieren, oder `class`, um es zu formatieren.

Tooltip

Ist ein Fenster zu einem Element in der Nähe des Mauszeigers, das zusätzliche Informationen in einer grafischen Benutzeroberfläche anzeigt. Ein weiterer Name in der allgemeinen Sprache der Webdeveloper ist **MouseOver**, da das Fenster erscheint, wenn der Mauszeiger kurze Zeit über einem Element stehen bleibt.

Der Ursprung liegt in den Anfängen der grafischen Oberflächen. Die Tooltips erklärten die Bedienelemente mit ein paar kurzen Worten. Im Web werden sie oft für zusätzliche Informationen benutzt, die in der Darstellung momentan keinen Platz finden.

TSV – Tab Separated Values

Ist ein spezielles Dateiformat. Es verhält sich ähnlich wie eine CSV-Datei, verwendet aber statt des Kommas einen Tabulator als Trennzeichen der Datenfelder.

URI – Uniform Ressource Identifier

Ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. Er wurde in RFC 3986 (<http://tools.ietf.org/html/rfc3986>) und ursprünglich in RFC 1630 definiert. URIs werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, Aufruf von Webservices, aber auch E-Mail-Empfängern) in Netzwerken, vor allem aber im World Wide Web verwendet.

URIs können als Zeichenfolge in Dokumenten verwendet, sie können aber auch von Hand geschrieben werden. Meistens werden sie im allgemeinen Sprachgebrauch Link oder Hyperlink genannt.

URL – Uniform Ressource Locator

URLs identifizieren und lokalisieren Ressourcen wie eine Website über die zu verwendende Zugriffsmethode, beispielsweise das verwendete Protokoll (etwa HTTP oder FTP) und die Ressource in Computernetzwerken. Die letzte gültige Definition wurde in RFC 1738 (<http://tools.ietf.org/html/rfc1738>) festgelegt.

URLs sind eine Unterart der URIs und werden auch oft miteinander verwechselt. Sie unterscheiden sich aber darin, dass URIs keine Zugriffsmethode enthalten.

W3C – World Wide Web Consortium

Ist ein 1994 gegründeter Zusammenschluss aus verschiedenen Organisationen (Firmen, Regierungen und auch Forschungseinrichtungen), der sich mit der Standardisierung und Weiterentwicklung des World Wide Web beschäftigt.

Widget

Der Begriff ist ein Kunstwort aus den englischen Wörtern *window* und *gadget*. Es bezeichnet einen eigenständigen Bereich in einer grafischen Benutzeroberfläche, der dem Anwender zusätzliche Informationen oder Steuerelemente zur Verfügung stellt.

WWW – World Wide Web

Ist ein Teilbereich des Internets, der oft mit diesem gleichgesetzt wird. Wesentliche Bestandteile sind die Darstellung von multimedialen Inhalten und das Vorhandensein von Verknüpfungen, sogenannten Links (Hyperlinks). Seiten für das WWW werden in der Beschreibungssprache HTML erstellt. Für die optische Gestaltung sind CSS-Dateien notwendig. Sollen die Dokumente dem Benutzer auch eine Interaktion ermöglichen, wird meistens JavaScript eingesetzt.

XML – Extended Markup Language

Ist eine von SGML abgeleitete Auszeichnungssprache. Durch die gute Lesbarkeit für Mensch und Maschine wird dieses Format gern für den standardisierten Austausch von Daten verwendet, denn hier können Struktur und Daten des Dokuments sauber getrennt werden. XML ist kein endgültiges Format, sondern kann durch eine eigene Beschreibungsdatei ergänzt werden. Dadurch sind eigene Knoten möglich.

A.3 Verzeichnis der Listings

| | |
|--|----|
| Listing 1: Textfarbe für alle Absätze mit der DOM-API ändern | 14 |
| Listing 2: Textfarbe für alle Absätze mit color ändern | 14 |
| Listing 3: Hintergrundfarbe für das body-Element mit D3 ändern | 15 |
| Listing 4: Textfarbe für alle p-Elemente mit einer Funktion zufällig ändern | 16 |
| Listing 5: Hintergrundfarbe für alle p-Elemente abwechselnd festlegen | 16 |
| Listing 6: Array mit sechs Werten definieren | 16 |
| Listing 7: Schriftgröße für alle p-Elemente dynamisch auf Basis eines Arrays festlegen ... | 16 |
| Listing 8: Elemente der enter-Selection hinzufügen | 17 |
| Listing 9: Aufteilen der Knoten in drei logische Gruppen | 17 |
| Listing 10: Die Hintergrundfarbe eines Dokuments ändern | 18 |
| Listing 11: Plug-in für D3 aktivieren | 19 |
| Listing 12: Eine Tabelle in HTML | 19 |
| Listing 13: Die Grundstruktur des HTML-Dokuments für die erste Übung | 26 |
| Listing 14: Ein numerisches Array in JavaScript darstellen | 26 |

| | |
|--|-----|
| Listing 15: Darstellung eines Balkendiagramms mit CSS | 28 |
| Listing 16: Erstellen eines Balkendiagramms mit JavaScript und D3 | 29 |
| Listing 17: Angepasstes Skript, um die maximalen Werte zu skalieren | 32 |
| Listing 18: Verknüpfte Methoden in einer Zeile | 34 |
| Listing 19: Verknüpfte Methoden mehrzeilig dargestellt | 35 |
| Listing 20: D3-Methoden ohne Verknüpfung darstellen | 36 |
| Listing 21: Die Grundstruktur des HTML-Dokuments für die zweite Übung | 41 |
| Listing 22: Die CSS-Formatierung und die HTML-Elemente für die zweite Übung | 41 |
| Listing 23: Werte für die zweite Übung in der CSV-Datei | 42 |
| Listing 24: Das Skript für die zweite Übung | 42 |
| Listing 25: Basiswerte für das Kreisdiagramm | 43 |
| Listing 26: Das Kreissegment und das Diagramm definieren | 43 |
| Listing 27: Das Kreisdiagramm zeichnen | 44 |
| Listing 28: Die Daten aus der CSV-Datei einlesen und verarbeiten | 45 |
| Listing 29: Die Grundstruktur des HTML-Dokuments für die dritte Übung | 52 |
| Listing 30: Die CSS-Formatierung und die HTML-Elemente für die dritte Übung | 54 |
| Listing 31: Werte für die dritte Übung in der CSV-Datei | 54 |
| Listing 32: Das Skript für die dritte Übung | 54 |
| Listing 33: Basiswerte für das Kreisdiagramm | 55 |
| Listing 34: Die Daten für das Kreisdiagramm aus einer CSV-Datei lesen | 56 |
| Listing 35: Die Daten der CSV-Datei verarbeiten | 56 |
| Listing 36: Die Ergänzung für die Animation | 58 |
| Listing 37: Die ergänzte Funktion change() | 58 |
| Listing 38: Die Callback-Funktion für den Tween der Segmente | 59 |
| Listing 39: Werte mit fehlenden Daten für die dritte Übung | 59 |
| Listing 40: Quellcode der dritten Übung mit geänderter Datenbasis | 60 |
| Listing 41: Für fehlende Daten angepasste Callback-Funktion | 61 |
| Listing 42: Die Segmente mit einem title-Attribut ergänzen | 63 |
| Listing 43: Der vollständige Quellcode für ein animiertes Ringdiagramm | 66 |
| Listing 44: Testen von MAMP auf Mac OS X mit der Datei index.php | 91 |
| Listing 45: Das Stylesheet für den Métro Information Tracker | 112 |
| Listing 46: Die Basis-HTML-Datei für den MIT | 112 |
| Listing 47: Einlesen und Kombinieren von zwei JSON-Dateien | 116 |
| Listing 48: Der Code zum Anzeigen des Force-Directed-Diagramms | 121 |
| Listing 49: Der Code für die Interaktion mit dem Force-Directed-Diagramm | 137 |
| Listing 50: Das angepasste Stylesheet für den Métro Information Tracker | 142 |
| Listing 51: Die angepasste Startseite für die Suche und die geänderte Datenbasis | 142 |
| Listing 52: Der geänderte Quellcode für die Suche und die geänderte Datenbasis | 149 |
| Listing 53: Das erweiterte Stylesheet für den Métro Information Tracker | 156 |
| Listing 54: Die erweiterte HTML-Datei des MIT | 157 |
| Listing 55: Globale Variable für den Métro Information Tracker | 159 |

| | |
|--|-----|
| Listing 56: Einlesen der JSON-Daten in den MIT | 162 |
| Listing 57: Die Funktion setData() zum Erstellen der Datenstruktur | 167 |
| Listing 58: Die Funktion setStationLines() | 169 |
| Listing 59: Darstellen der Daten mit der Funktion svgUpdate() | 172 |
| Listing 60: Bewegen der Knoten für ein ideales Layout | 173 |
| Listing 61: Die Farben für einen Knoten mit color() ermitteln | 173 |
| Listing 62: Alle Kindknoten mit flatten() ermitteln | 174 |
| Listing 63: Knoten einer Ebene mit closeNodes() ausblenden | 175 |
| Listing 64: Einen Knoten anklicken und das Event mit click() verarbeiten | 176 |
| Listing 65: Die Stationsinformation mit showStationInfo() anzeigen | 182 |
| Listing 66: Die Bilder von flickr mit showStationPictures() abrufen | 183 |
| Listing 67: Die Bilder von flickr mit showFlickrPictures() anzeigen | 184 |
| Listing 68: Das Dokument mithilfe von jQuery initialisieren | 186 |
| Listing 69: Die Knoten mit searchNode() durchsuchen | 187 |
| Listing 70: Ein gefundenes Element mit changeElement() hervorheben | 188 |
| Listing 71: Eine JSON-Variable mit getObjects() effektiv durchsuchen | 189 |
| Listing 72: Die URL zur flickr-API mit getFlickrURL() zusammenstellen | 189 |
| Listing 73: Nationale Sonderzeichen aus Texten entfernen | 191 |
| Listing 74: Zahlen mit formatNumber() optisch ansprechend darstellen | 192 |
| Listing 75: Dezimalgrad mit GradKonv() umwandeln | 192 |
| Listing 76: Ein Quadrat und ein Rechteck mit svg erzeugen | 195 |
| Listing 77: Einen Kreis und eine Ellipse mit svg erzeugen | 195 |
| Listing 78: Eine gerade Linie und einen Linienzug mit svg erzeugen | 196 |
| Listing 79: Ein Polygon mit svg erzeugen | 196 |
| Listing 80: Das image-Element in HTML | 197 |
| Listing 81: Die angepasste svgUpdate()-Funktion | 198 |
| Listing 82: Die Funktion getNodeInfo() für das image-Element | 200 |
| Listing 83: Anpassungen der Funktion tick() | 200 |
| Listing 84: Die Überschrift h1 positionieren | 201 |
| Listing 85: Die Symbolleiste für den Inhalt definieren | 201 |
| Listing 86: Das Aussehen der Buttons in der Symbolleiste festlegen | 202 |
| Listing 87: Die neue Symbolleiste in der Datei index2.html | 202 |
| Listing 88: Die Funktion closeNodes() anpassen | 204 |
| Listing 89: Die Buttons für das Anzeigen der Levels aktivieren | 204 |
| Listing 90: Auszug aus der Datei parisPath.js | 206 |
| Listing 91: Auszug des freq-Objekts von ratp2011.json | 207 |
| Listing 92: Die paris.html für die Darstellung des geografischen MIT | 209 |
| Listing 93: Das Layout in der Datei paris.css für den MIT der Version 3 | 215 |
| Listing 94: Die ersten Zeilen der Datei fontello.css | 217 |
| Listing 95: Die Klassen mit den Symbolen in fontello.css | 218 |
| Listing 96: Globale Variable initialisieren und das svg-Element erzeugen | 218 |

| | |
|--|-----|
| Listing 97: Den Viewport für die Anwendung definieren | 219 |
| Listing 98: Den Hintergrund der Applikation festlegen | 219 |
| Listing 99: Die Daten für ein Jahr laden und im Webbrowser darstellen | 221 |
| Listing 100: Die Daten der Jahre in die Applikation laden | 224 |
| Listing 101: Den Métro Information Tracker mit jQuery initialisieren | 225 |
| Listing 102: Eine Station auswählen und die Informationen mit statSelect anzeigen | 226 |
| Listing 103: Das Balkendiagramm mit statBarChart() anzeigen | 227 |
| Listing 104: Die Informationen einer Station mit statDeselect() ausblenden | 228 |
| Listing 105: Die Informationen einer Linie mit lineSelect() anzeigen | 229 |
| Listing 106: Die Funktion lineDeselect() zum Ausblenden von Informationen einer Linie .. | 230 |
| Listing 107: Die Hilfsfunktionen für den Métro Information Tracker | 232 |
| Listing 108: Die Métrostationen als Ringe erzeugen | 232 |
| Listing 109: Widget und Tooltip in die HTML-Datei einfügen | 237 |
| Listing 110: Ergänzungen im Stylesheet paris.css für das Widget und den Tooltip | 239 |
| Listing 111: Zusätzliche globale Variable in der Anwendung | 240 |
| Listing 112: Das click-Event für das Ein-/Ausblenden des Widgets | 241 |
| Listing 113: Den Tooltip positionieren und mit Informationen ergänzen | 242 |
| Listing 114: Den Tooltip ausblenden | 243 |
| Listing 115: Das aktuelle Jahr mit getActiveYear() ermitteln | 243 |
| Listing 116: Mit setTrafficLine() das Fahrgastaufkommen berechnen | 245 |
| Listing 117: Das Liniendiagramm mit setLineChart() erstellen | 247 |

Stichwortverzeichnis

Symbole

_children 175, 176

A

Abkühlung 125, 131

accessor 49, 51, 132, 235

Achse 248

 x 245, 248

 y 245, 248

active 202

Administratorkennwort 80

AJAX 167, 255

Alphawert 125

Anfrage

 GET 49, 117, 256, 258

 POST 258

Animation 58, 152

 Abfolge 70

 neu 71

 Tween 71

Anonyme Funktion 31, 36, 45, 48, 50, 67,
 68, 69, 71, 139, 174

ANSI 255

Anziehungskraft 122, 125, 128

Apache 74, 83, 87, 91

API 21, 154, 163, 167, 168, 169, 182, 191,
 255

 flickr 189, 190

 Key 159, 161

apiKey 161

append() 35, 37, 193, 196

Aptana 75, 84, 91, 92, 109

 Project Explorer 100, 102

Projekt 99

Projekt neu anlegen 100

Workspace 99

arc() 44, 47, 232

 innerRadius() 47, 232

 outerRadius() 47

areaPAR 222

Array 26, 30, 31, 37, 38, 39, 40, 42, 48,
 50, 51, 66, 71, 114, 116, 122, 125, 131, 132,
 137, 164, 169, 174, 175, 188, 192, 205,
 210, 234, 235, 255

 assoziativ 224, 244, 255

 dataFreq 224

 global 224

 mitIndex 167

 numerisch 30

 Objekt 48

 pictures 183

 sortieren 48

ASCII 255, 257

Assoziatives Array 224, 244, 255

Asynchron 49, 50, 68, 116, 117

attr() 31, 36, 152, 196

Attribut

 _children 175, 176

 children 164

 class 31, 187, 259

 id 113, 150, 151, 259

 info 174

 name 164

 owner 168

 photo 168

 picCount 174, 176

- preserveAspectRatio 218
- stationInfo 176
- title 63, 168
- traffic 163
- trafficPos 163
- transform 200
- viewBox 218

- Attributknoten 21, 38

- attrTween() 71, 153

- axis() 248

- axis(), ticks() 248

- axis, tickFormat() 248

B

- background 219

- Balkendiagramm 26, 224, 225, 226, 227

- Barnes-Hut 131

- barStation 210

- Basiselemente 194

- Baumdiagramm 154, 164, 170, 173, 175, 192, 197

- Baumstruktur 20, 104

- behavior

- zoom() 234

- Benutzerinterface 114

- Benutzerkontensteuerung 84, 85

- Benutzerverwaltung 79

- Bibliothek, jQuery 113, 137, 138

- Blockelement 27

- body 30, 35, 37, 38, 43, 55, 113, 257, 258

- Breitengrad 137

- Browser

- Chrome 105

- Firefox 105

- btnLevel 201

C

- call() 219, 233

- Callback-Funktion 49, 51, 56, 58, 61, 68, 71, 116, 117, 122, 123, 124, 137, 138, 139, 149, 152, 153, 162, 167, 172, 173, 174,

- 199, 200, 222, 223, 224, 229, 233, 234, 235, 236, 248

- category10() 66

- category20() 66

- category20b() 66

- category20c() 66

- CDN 113, 210

- Chaining 30, 34, 35, 233

- change 68

- changeElement() 151, 187, 188

- charge() 122, 128

- checked 69

- children 164, 192

- circle 194, 196

- class 31, 187

- classed() 236

- clearName 150

- click 138, 241

- Event 224

- click() 176, 177

- closeNodes() 169, 175, 176, 203

- color 217

- color() 173

- container 222

- context 167

- contLines 224

- contMap 210, 218

- contParis 222

- Control Panel 82, 86, 87

- contStations 223

- css

- active 202

- btnLevel 201

- color 217

- font-size 217

- important 38

- Klasse 69

- linkData 224

- toolbarContent 201

- CSS 13, 31, 38, 41, 52, 255, 260

- Klasse 224

- css/ 210, 216
- CSS-Datei 140, 217, 237
- CSS-Klasse 36, 55, 199, 202, 217, 236, 243
- CSV 42, 45, 46, 49, 54, 56, 59, 256
- CSV einlesen 49
- csv() 45, 49, 51, 63, 149
- CSV-Daten 51, 116, 140
- CSV-Format 259

D

- d 139
- d3
 - ascending 48
 - csv() 49, 63, 149
 - descending 48
 - event() 178
 - interpolate() 71
 - json() 117, 162, 222, 224
 - map() 222, 233
 - max() 235
 - median() 234
 - min() 235
- D3, Plug-in 162
- Darstellung
 - Balkendiagramm 26
 - Baumdiagramm 154
 - Force-Directed-Graph 106, 139, 154
 - Kreisdiagramm 40
- data 233
- data() 31, 37, 67, 223
- data/ 206
- data/img/ 198
- dataFreq 224
- dataRATP 162
- Dataset 67
- Datei
 - CSS 113, 140, 217, 237
 - DMG 77, 91
 - HTML 113, 142, 184, 194, 207, 210, 215, 255, 256, 257

- index.html 112, 114, 156
- index2.html 202
- JS 143
- lightbox.css 157
- lightbox.min.js 157
- main.js 114, 117
- main2.js 158
- metroparisbase.json 108, 114
- metroparisfull.json 162, 163
- paris.html 209, 218, 237
- paris.js 218, 239
- parisPath.js 205
- ratp2011.json 206
- ratp2012.json 206
- ratp2013.json 206
- screen.css 157
- style.css 108, 155, 201
- XML 194, 256
- ZIP 216

Daten

- Array 26, 42
- aus Datei lesen 49
- CSV 42, 45, 46, 49, 51, 54, 56, 59, 116, 140
- HTML 49
- JSON 49, 69, 114, 116, 122, 132, 137, 159, 161, 258
- Text 49
- TSV 49
- verarbeiten 49
- XML 49

Datenbankserver 88, 253

- Status 254

Datenpunkt 131, 132

Datentyp

- object 151
- string 151

datum() 67

defaultPrevented 178

delay() 70, 151, 153

Designtime 50

Desktop 86
 Diagramm, Force-Directed 125
 Diagrammfläche 127
 Dienst 87
 distance() 122
 Distanz 131
 div 28, 30, 31, 35, 36, 37, 38, 55, 113, 158,
 197, 202, 210, 218, 236, 239, 258
 div-Element 184
 DMG-Datei 77, 91
 Dock 83
 Document Object Model 19
 Dokument
 HTML 19, 21, 26, 27, 30, 40, 52, 59, 68,
 91
 XML 19
 Dokumentknoten 21
 DOM 13, 19, 21, 30, 35, 37, 39, 67, 68, 69,
 71, 193, 194, 256, 257
 domain() 32, 40
 drawArc() 223, 232
 drawLine() 224, 233
 duration() 70, 149, 151

E

each() 68
 Eigenschaft 38
 checked 69
 clearName 150
 idxLine 149
 idxStation 149
 selected 69
 value 69
 elem 151
 Element
 animieren 70
 Block 27
 body 30, 35, 37, 38, 43, 55, 113, 257,
 258
 circle 194, 196

 div 28, 30, 31, 35, 36, 37, 38, 55, 113,
 158, 184, 197, 202, 210, 218, 236, 239,
 258
 form 54, 69
 g 197, 219, 222, 223, 224, 248
 h1 91, 201, 259
 head 113, 257
 html 257
 image 197, 198
 img 26
 input 56, 68, 69
 li 182
 p 38, 39
 path 222, 223, 224, 248
 rect 219, 248
 script 43, 55, 59, 113
 span 201, 202
 style 30, 31
 svg 187, 196, 197, 219, 222, 233
 table 19, 21
 td 21
 text 196, 197, 199
 this 67, 68, 69, 71
 title 173, 196, 197, 199
 tr 20
 Elementknoten 21
 Elternknoten 20, 173
 empty() 37
 endAngle() 47
 enter() 31, 37, 172, 198
 enter-Selection 17, 37, 38, 63
 Entwicklungsumgebung 91
 Ereignis 51, 68, 177, 225, 256
 Ereignisverfolgung 138
 eval() 257
 event
 defaultPrevented 178
 Event 177, 256
 change 68
 click 138, 178, 224, 241
 Handling 178

- Listener 68, 219, 234
- mousedown 178
- mousemove 248
- mouseout 223, 224
- mouseover 138, 223, 224, 248
- mouseup 178
- submit 138
- event() 178
- Event-Listener 138, 173, 223, 224
- Events 51, 68
- exit() 172
- exit-Selection 17, 38

F

- Farbe, RGB 71
- Finder 81, 91
- Firewall 87
- flatten() 172, 174
- flickr 154, 160, 167, 182, 183, 203
 - API 167, 168, 189
 - API-Key 159, 161
 - place_id 190
 - woe_id 190
- font/ 216
- Fontello 215
- fontello.css 215, 216, 218
- font-size 217
- force() 125
 - alpha() 131
 - charge() 126
 - distance() 131
 - friction() 129
 - gravity() 128
 - links() 132
 - nodes() 131
 - size() 126
 - start() 132
 - theta() 131
- Force-Directed-Graph 106, 125, 139, 154
- forEach() 45, 50, 174
- form 54, 69

Format

- CSV 259
- JSON 104, 122, 167
- formatNumber() 149, 191
- for-Schleife 123, 167
- Freigaben 252
- Freigaben, Web 253
- freq 206, 222
- FTP 256, 259
- Funktion
 - accessor 132, 235
 - anonym 31, 36, 45, 48, 50, 67, 68, 69, 71, 139, 174
 - Callback 49, 56, 58, 61, 68, 71, 116, 117, 122, 123, 124, 137, 138, 139, 149, 152, 153, 162, 167, 172, 173, 174, 199, 200, 222, 223, 224, 229, 233, 234, 235, 236, 248
 - changeElement() 151, 187, 188
 - click() 176, 177
 - closeNodes() 169, 175, 176, 203
 - color() 173
 - drawArc() 223, 232
 - drawLine() 224, 233
 - flatten() 172, 174
 - formatNumber() 149, 191
 - genPie() 223, 233
 - getActiveYear() 243
 - getFlickrURL() 167, 189
 - getNodeInfo() 200
 - getObjects() 188
 - getRadius() 149
 - getTraffic() 150
 - GradKonv() 137, 192
 - key 38
 - line() 233
 - lineDeselect() 230
 - lineSelect() 228
 - loadYear() 221, 225
 - loadYears() 224, 225
 - mousemove() 248

- nodeClick() 149, 151
- recurse() 174, 175
- searchNode() 186
- setData() 162, 164
- setLineChart() 245
- setNodeRadius() 149
- setStationInfos() 140, 150
- setStationLines() 169
- showFlickrPictures() 183, 184
- showStationPictures 182
- sortStations() 223
- statBarChart() 227
- statDeselect() 228, 243
- stationClearName() 151
- statSelect() 225, 241
- success 167
- svgUpdate() 169, 170, 175, 176, 198
- tick() 124, 173, 200
- transform() 232
- verschachtelt 174

G

- g 197, 219, 222, 223, 224, 248
- genPie() 223, 233
- geo
 - mercator() 234
- Geoid 222
- Geokoordinaten 205
- geoPrj 222
- Geschwisterknoten 21, 175
- GET 49, 117, 256, 258
- getActiveYear() 243
- getFlickrURL() 167, 189
- getNodeInfo() 200
- getObjects() 188
- getRadius() 149
- getTraffic() 150
- Globale Variable 116, 161, 162, 189, 218, 219, 222, 239
- Globales Array 224
- Google

- CDN 113

- Fonts 113

- Google Chrome 105

- GradKonv() 137, 192

- Grafik

- Pixel 194

- SVG 122, 187, 194, 206, 248

- Vektor 194

- Grafik zeichnen 46

- gravity() 122

H

- h1 91, 201, 259

- head 113, 257

- htdocs 88, 90, 99

- html 257

- HTML 13, 19, 21, 26, 27, 30, 34, 40, 49,

- 52, 59, 68, 91, 256, 258, 260

- 4.01 257

- 5 257

- HTML-Datei 113, 142, 184, 194, 207, 210,

- 215, 255, 256, 257

- HTML-Element 137

- HTTP 257, 259

- Hyperlink 259

I

- i 139

- id 150, 151

- ID

- barStation 210

- contMap 210, 218

- infoBoxLine 239

- infoBoxStation 239

- infos 210

- id-Attribut 113

- idxLine 137, 149

- idxStation 137, 149

- if-Anweisung 122

- image 197, 198

- img 26

- index.html 112, 114, 156
- index.php 91
- index2.html 202
- info 174
- infoBoxLine 239
- infoBoxStation 239
- infos 210
- infosLine 210
- infosStation 210
- innerRadius() 47, 232
- input 56, 68, 69
- insert() 37, 193, 196
- Installation
 - lokal 88
 - Webserver 80, 84
- Installationsverzeichnis 80, 85
- intDel 151
- intDur 151
- interpolate() 71, 233
- Interpolieren 71
- intRadiusEnd 151
- intRadiusStart 151
- ISO-8859-15 105

J

- JavaScript 260
- JavaScript, jQuery 113
- jQuery 113, 137, 138, 149, 157, 162, 167, 184, 185, 188, 222, 224, 241, 242
 - ajax() 167
- js/ 205, 218
- JS-Datei 143
- JSON 49, 69, 257
 - Aufruf 162
 - Datei 122
 - Daten 116, 122, 132, 137, 159, 161, 174, 258
 - Daten einlesen 161
 - Format 104, 114, 122, 167
 - Struktur 137, 149, 162, 172, 184, 187, 188, 189

- Viewer 104, 105
- json() 117, 162, 222, 224
- jsonCalls 167, 169
- jsonTraffic 149

K

- Kante 107, 122, 123, 124, 125, 129, 132, 172
- Kennwort, Administrator 80
- key 188
- key-Funktion 38
- Key-Value-Pair 49
- Kindknoten 20, 175, 192
- Kindknoten() 176
- Klasse 55, 69, 224
 - CSS 199, 202, 217, 236, 243
 - infosLine 210
 - infosStation 210
- Knoten 20, 104, 106, 116, 122, 123, 124, 125, 126, 129, 131, 132, 149, 150, 151, 169, 172, 173, 174, 175, 186, 187, 197, 203, 257, 260
 - Attribut 21, 38
 - Breitengrad 106
 - children 164
 - Dokument 21
 - Element 21
 - Eltern 20
 - Geschwister 21, 175
 - id 106
 - info 114
 - Kind 20, 114, 175, 176
 - Längengrad 106
 - lat 106
 - lon 106
 - name 106
 - pos 137
 - ref 106
 - stations 114
 - Text 21, 37
 - Wurzel 20
- Koeffizient, Reibung 130

Koordinaten 132
 Koordinatensystem 232, 233
 Kreisbogen 47
 Kreisdiagramm 40
 Kressegment 47, 232
 Kuchendiagramm 232, 233

L

Längengrad 137
 lat 137
 layout
 force() 125
 alpha() 131
 charge() 126
 distance() 131
 friction() 129
 gravity() 128
 links() 132
 nodes() 131
 size() 126
 start () 132
 theta() 131
 pie() 44, 48
 sort() 48
 value() 48
 layout()
 tree() 172, 192
 li-Element 182
 Lightbox 157, 184
 lightbox.css 157
 lightbox.min.js 157
 line() 233, 235, 248
 interpolate() 233
 linear
 domain() 40
 range() 40
 linear() 32, 39
 lineDeselect() 230
 lines 224
 lineSelect() 228
 Linien 244

Liniendiagramm 236, 239, 245, 248
 Link 259
 linkData 224
 Listener 68
 Listener, Event 138, 223, 224
 Lizenzbedingungen 76, 93
 loadYear() 221, 225
 loadYears() 224, 225
 localhost 89, 91
 log() 223, 235
 Lokale Variable 198
 lon 137

M

main.js 114, 117
 main2.js 158
 MAMP 74, 82, 90, 251
 Control Panel 82
 MAMP Pro 82
 map() 222, 233
 Maps 222
 Math.max() 235
 Math.min() 235
 max() 235
 Median 222
 median() 234
 Mercator 222
 mercator() 234
 Mercator, Projektion 222
 Messwerte 39
 Metainformationen 257
 Methode
 append() 196
 attr() 196
 insert() 196
 Methoden verknüpfen 34
 metroparisbase.json 108, 114
 metroparisfull.json 162, 163
 Microsoft Excel 139
 MIME-Type 49, 117, 257
 min() 235

- MIT 73
 - Array
 - arrcolor 123
 - arrLinks 122
 - arrNodes 123
 - metroStations 122
 - mitIndex 167
 - pictures 183
 - ArraysdataFreq 224
 - Attribut
 - _children 175, 176
 - children 164
 - group 123
 - id 150, 151
 - idx 115, 122
 - info 174
 - lines 122
 - name 164
 - picCount 174, 176
 - stationInfo 176
 - target 115, 122
 - traffic 163
 - trafficPos 163
 - value 123
 - css
 - active 202
 - btnLevel 201
 - linkData 224
 - toolbarContent 201
 - Datei
 - index.html 112, 142, 156
 - index2.html 202
 - lightbox.min.js 157
 - main.js 114, 117, 143
 - main2.js 158
 - metroparisbase.json 108, 114
 - metroparisfull.json 162, 163
 - paris.html 209, 218, 237
 - paris.js 218, 239
 - parisPath.js 205
 - ratp2011.json 206
 - ratp2012.json 206
 - ratp2013.json 206
 - style.css 108, 140, 155
 - Dateistyle.css 201
 - Eigenschaft
 - clearName 150
 - group 123
 - idxLine 149
 - idxStation 149
 - lines 123
 - Funktion
 - changeElement() 151, 187, 188
 - click() 176, 177
 - closeNodes() 169, 175, 176
 - color() 173
 - drawArc() 223, 232
 - drawLine() 224, 233
 - flatten() 172, 174
 - formatNumber() 149, 191
 - genPie() 223, 233
 - getActiveYear() 243
 - getFlickrURL() 167, 189
 - getNodeInfo() 200
 - getObjects() 188
 - getRadius() 149
 - getTraffic() 150
 - GradKonv() 137, 192
 - line() 233
 - lineDeselect() 230
 - lineSelect() 228
 - loadYear() 221, 225
 - loadYears() 224, 225
 - mousemove() 248
 - nodeClick 137
 - nodeClick() 149, 151
 - recurse() 174, 175
 - searchNode() 186
 - setData() 162, 164
 - setLineChart() 245
 - setNodeRadius() 149
 - setStationInfos() 116, 140, 150

- setStationLines() 169
- showFlickrPictures() 183, 184
- showStationPictures 182
- sortStations() 223
- statBarChart() 227
- statDeselect() 228, 243
- stationClearName() 151
- statSelect() 225, 241
- svgUpdate() 169, 170, 175, 176, 198
- tick() 124, 173, 200
- transform() 232
- ID
 - barStation 210
 - contMap 210, 218
 - infoBoxLine 239
 - infoBoxStation 239
 - infos 210
- JSON-Struktur 137
- Klasse
 - infosLine 210
 - infosStation 210
- Knoten
 - children 164
 - info 114
 - pos 137
 - stations 114
 - title 124
- Layout 109
- Objekt
 - freq 206
 - station 227
- Parameter
 - data 233
 - prj 232, 233
 - year 221
- Positionsdaten 116
- Projekt 109
- Schlüssel
 - rank 207
 - traffic 207
- Skript 143
- Startseite 142, 156
- Stylesheet 140
 - fontello.css 215, 216, 218
 - paris.css 210
- svg
 - preserveAspectRatio 218
 - viewBox 218
- Variable
 - apiKey 161
 - areaPAR 222
 - background 219
 - container 222
 - contLines 224
 - contParis 222
 - contStations 223
 - dataRATP 162
 - drawArc 232
 - drawLine 233
 - elem 151
 - freq 222
 - genPie 233
 - geoPrj 222
 - graph 123
 - idxLine 137
 - idxStation 137
 - intDel 151
 - intDur 151
 - intRadiusEnd 151
 - intRadiusStart 151
 - jsonCalls 167, 169
 - jsonTraffic 149
 - lat 137
 - line 233
 - lineDeselect 230
 - lines 224
 - lineSelect 228
 - lon 137
 - nodeEnter 198
 - objStation 137
 - pie 232
 - root 162, 168, 169, 172, 189

- statDeselect 228
- statSelect 225
- transform 232
- viewport 219
- Verzeichnis
 - /css 157, 210
 - /data 162, 206
 - /img 157
 - /js 157, 158, 205, 218
 - css 107, 216
 - data 107, 108
 - data/img 107
 - data/img/ 198
 - font 216
 - js 107, 108
- MIT Datei
 - lightbox.css 157
 - screen.css 157
- MIT-Funktion
 - closeNodes() 203
- mitIndex 167
- mousemove() 248
- mouseout 223, 224
- mouseover 138, 223, 224
- Mozilla Firefox 105
- MySQL 74, 78, 84, 87, 253

N

- name 164
- Netz, vermascht 107
- node 34
- Node 14, 105, 150, 151, 172, 174, 176, 187, 257
- nodeClick() 137, 149, 151
- nodeEnter 198
- nodes 20, 106
- null 38, 48, 51, 67, 68

O

- obj 188
- object 151

Objekt

- Array 48
- freq 206
- station 227
- objStation 137
- on() 68, 139, 173, 223, 224
- Open Data 49, 71, 73, 139, 162, 204
 - Métro 102
 - Open Street Map 103
 - Portale 72
- Open Street Map 114, 116, 132
- Operator
 - Punkt 34
 - style() 196
- ordinal
 - range() 43
- ordinal() 46
 - range() 46
- outerRadius() 47

P

- p 38, 39
- Parameter
 - d 139
 - data 233
 - distance 132
 - fixed 132
 - i 139
 - key 188
 - obj 188
 - prj 232, 233
 - source 132
 - strength 132
 - target 132
 - val 188
 - weight 132
 - year 221
- paris.css 210
- paris.html 209, 218, 237
- paris.js 218, 239
- parisPath.js 205

- Parser 258
- path 222, 223, 224, 248
- Path 194
- Performance 88
- Perl 84
- PHP 74, 78, 84, 90
- PHP Project 101
- phpinfo 90
- picCount 174, 176
- pictures 183
- pie 232
- pie() 44, 48
 - sort() 48
 - value() 48
- place_id 190
- Plug-in 18, 162
- Portale, Open Data 72
- pos 137
- POST 258
- preserveAspectRatio 218
- Primitives 15
- prj 232, 233
- Project Explorer 100, 102, 109
- Projekt, Aptana 99
- Projektion
 - Mercator 222, 234
- property() 69
- Punktoperator 34

Q

- Quellen, Open Data 72
- Queue 162
- Quickinfo 259

R

- Radiant 48
- Radius 172
- range() 32, 40, 43, 46
- rank 207
- RATP 162
- ratp2011.json 206

- ratp2012.json 206
- ratp2013.json 206
- rect 219, 248
- recurse() 174, 175
- Reibung 129, 130
- Rekursion 174
- Rekursiv 68
- remove() 193
- Request
 - GET 256
 - POST 258
- RFC 258
 - 1630 259
 - 1738 259
 - 1946 257
 - 1985 256
 - 2045 257
 - 2046 257
 - 2047 257
 - 2048 257
 - 2049 257
 - 2616 257
 - 3986 259
 - 4180 49
 - 4627 257
 - 5322 257
 - 7158 257
 - 7159 257
 - 822 257
 - 959 256
- RGB 71
- root 162, 168, 169, 172, 189
- Runtime 50

S

- scale
 - category10() 66
 - category20() 66
 - category20b() 66
 - category20c() 66
 - linear

- domain() 40
- range() 40
- linear() 32, 39
- ordinal() 46
- scale() 46, 219, 234
 - log() 223, 235
 - ordinal() 46
 - range() 46
- scaleExtent() 219, 234
- Schleife 45
- Schlüssel
 - rank 207
 - traffic 207
- Schlüssel-Wert-Paar 49, 256
- Schwerkraft 125, 128, 129
- screen.css 157
- script 43, 55, 59, 113
- searchNode() 186
- select() 30, 33, 35
- selectAll() 14, 34
- selected 69
- selection
 - append() 37
 - attr() 36
 - call() 219
 - classed() 236
 - data() 37, 67, 223
 - datum() 67
 - each() 68
 - enter() 37
 - on() 68, 139, 223, 224
 - property() 69
 - remove() 193
 - scale 39
 - sort() 223, 229
 - style() 38
 - text() 39
 - transition() 69, 149, 151, 153
 - delay() 153
 - duration() 149, 153
- Selection 14, 16, 35, 36, 152
- enter 17, 37, 38, 63
- enter() 31
- exit 38
- update 38
- selection()
 - call() 233
 - enter() 172
 - exit() 172
 - forEach() 174
 - style() 151
- Semikolon 35
- Server starten 83
- set
 - forEach() 50
- setData() 162, 164
- setLineChart() 245
- setNodeRadius() 149
- setStationInfos() 140, 150
- setStationLines() 169
- SGML 258, 260
- showFlickrPictures() 183, 184
- showStationPictures 182
- siblings 21, 192
- Sizzle 15
- Skalierung 39
- sort() 223, 229
- sortStations() 223
- span 201, 202
- start() 125
- startAngle() 47
- Startmenü 86, 95
- Startseite 156
- statBarChart() 227
- statDeselect() 228, 243
- station 227
- stationClearName() 151
- stationInfo 176
- statSelect() 225, 241
- Stil, CSS 38
- string 151
- String 151

Struktur

JSON 149, 162, 172, 184, 187, 188, 189

style 30, 31

style() 31, 38, 46, 151, 152, 196

style.css 108, 155, 201

Stylesheet 113, 114, 155, 210, 215

fontello.css 215, 216, 218

paris.css 210

styleTween() 153

submit 138

svg 187, 196, 197, 219, 222, 233

arc() 44, 47, 232

axis() 248

Basiselemente 194

Ellipse 194, 195

gerade Linie 194, 196

Kreis 194, 195

line() 235, 248

Linienzug 194, 196

Polygon 194, 196

preserveAspectRatio 218

Recheckt 194

Rechteck 195

transform 219

translate() 232

viewBox 218

SVG 13, 258

SVG-Grafik 122, 187, 194, 206, 248

svgUpdate() 169, 170, 175, 176, 198

T

table 19, 21

Tag 258

Taskleiste 88

td 21

text 196, 197, 199

text() 31, 35, 39

TextEdit 75

Textknoten 21, 37

TextWrangler 75

Theta 125

this 36, 67, 68, 69, 71

tick() 124, 173, 200

tickFormat() 248

ticks() 248

title 63, 173, 196, 197, 199

toolbarContent 201

Tooltipp 236, 241, 243, 244, 259

tr 20

traffic 163

trafficPos 163

trafic 207

transform 200, 219

transform() 232

Transition 152

transition() 69, 149, 151, 153

attrTween() 71

delay() 70, 151, 153

duration() 70, 151, 153

Transitions 18

translate() 232

tree() 172, 192

tsv 259

TSV 49

Tupel 49

Tween 58, 71

U

Übergänge 18

undefined 50

Unterknoten 105

update-Auswahl 38

URI 259

URL 49, 159, 167, 189, 190, 256, 258, 259

UTF-8 105, 255

V

val 188

value 69

Variable

apiKey 161

areaPAR 222

background 219
container 222
contLines 224
contParis 222
contStations 223
dataRATP 162
drawArc 232
drawLine 233
elem 151
freq 222
genPie 233
geoPrj 222
global 116, 161, 189, 218, 219, 222, 239
idxLine 137
idxStation 137
intDel 151
intDur 151
intRadiusEnd 151
intRadiusStart 151
jsonCalls 167, 169
jsonTraffic 149
lat 137
line 233
lineDeselect 230
lines 224
lineSelect 228
lokal 198
lon 137
nodeEnter 198
objStation 137
pie 232
root 162, 168, 169, 172, 189
statDeselect 228
statSelect 225
transform 232
viewport 219
Vermaschtes Netz 107
Verschachtelte Funktionen 174
Verzeichnis
 /css 107, 157, 210, 216
 /data 107, 108, 162, 206

 /data/img 107
 /data/img/ 198
 /font 216
 /htdocs 88, 90, 99
 /img 157
 /Installation 80, 95
 /js 107, 108, 114, 157, 158, 205, 218
 /Programme 91
 /xampp 88
Verzögerung 70
viewBox 218
Viewer, JSON 104, 105
Viewport 219

W

W3C 19, 20, 256, 257, 260
Warteschlange 162
Webfreigabe 253
Webserver 78, 85, 251
 Dienst 87
 installieren 80, 84
 starten 81, 87
Wert
 null 38, 48, 51, 67, 68
 undefined 50
Widget 236, 240, 245, 248, 260
Windows 83
Winkelmaß 48
woe_id 190
WordPad 84
Workspace 99
Wurzelknoten 20
WWW 257, 260

X

x-Achse 245, 248
XAMPP 83, 84
XML 19, 49, 255, 258, 260
XML-Datei 194, 256

Y

y-Achse 245, 248

Yahoo! 159

year 221

Z

Zeichensatz

ISO-8859-15 105

UTF-8 105

ZIP-Datei 216

Zoom 234

zoom() 234

scale() 219, 234

scaleExtent() 219, 234

Zugriffsrechte 79

BigData

mit JavaScript visualisieren

Willkommen im Datenzeitalter. Ob Smartphone, DSL-Router oder auch beim Einkauf um die Ecke: Daten fallen überall in großer Menge an und werden gespeichert. Um diese großen Datenmengen effizient und nutzbringend einsetzen zu können, ist eine geeignete Darstellungsform notwendig. Mit D3.js steht eine JavaScript-Bibliothek bereit, um große Datenmengen in unterschiedlichsten Diagrammtypen darstellen zu können. Werden Sie damit Herr der Daten.

Erste Diagramme mit D3.js

Bevor Sie eigene Diagramme mit D3.js erstellen können, müssen Sie sich mit der entsprechenden Syntax auseinandersetzen. Gull erklärt Ihnen die wichtigsten Regeln im ersten Kapitel. Danach geht es direkt ans Eingemachte: Balkendiagramme, Kreisdiagramme und animierte Diagramme. Erleben Sie die großen Möglichkeiten von D3 anhand von eigenen Diagrammen.

Komplette D3-Anwendung umsetzen

Nach einem theoretischen und praktischen Einstieg in den ersten beiden Kapiteln widmet sich das Kapitel 3 einer großen Anwendung, dem Métro Information Tracker. Hierbei nutzen Sie öffentlich zugängliche Daten (Open Data), um mit unterschiedlichen Diagrammen Informationen zur Métro in Paris darzustellen. Hierbei setzen Sie sich mit allen Themen auseinander, die die moderne Webentwicklung zu bieten hat: HTML, CSS, JSON und SVG. In einem Force-Directed-Graph stellen Sie die Daten dar und bereichern diesen mit SVG-Grafiken an. Nachdem Sie dieses Projekt selbst umgesetzt haben, sind Sie gewappnet für eigene große D3-Anwendungen. Der Quellcode steht auch online zur Verfügung.

Die Königsdisziplin: Animation

So lernen Ihre Modelle das Laufen! Erfahren Sie, wie Sie eine Armatur – eine Art Skelett – für Ihre Charaktere erstellen und sie mit der 3-D-Grafik verknüpfen. Anschließend können Sie die Animationsphasen selbst erstellen und aufnehmen. Darüber hinaus sehen Sie, wie Sie Licht und Schatten für die Kameraaufnahme gekonnt einsetzen.

Aus dem Inhalt:

- Datendarstellung im Web mit D3
- Eigenheiten und Syntaxregeln
- Plug-ins von D3.js
- Document Object Model
- Visualisierung von Daten
- Erstellen eines Balkendiagramms
- Erstellen eines Kreisdiagramms
- Animierte Diagramme
- Open Data
- Daten als Force-Directed-Graph darstellen
- Stylesheet für Formatierung einsetzen
- Force-Directed-Graph als Baum darstellen
- JSON-Daten einlesen und verarbeiten
- Knoten im Diagramm suchen
- Grafiken für die Knoten verwenden
- Scalable Vector Graphics einsetzen

Über den Autor:

Clemens Gull unterrichtet Webprogrammierung und Webdesign an einer technischen Schule in Salzburg.

Sein Weblog *Guru 2.0* (www.guru-20.info) gehört zu den meistgelesenen deutschsprachigen Blogs zum Thema Internetprogrammierung.

