

# Programmieren für Journalisten

Steffen Kühne

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

# Über das Seminar

Teilnehmer des Seminars sollen die Grundlagen der Web-Entwicklung mit HTML, CSS und JavaScript lernen. Dabei steht die praktische Vermittlung von Programmierkenntnissen, so wie das Entwickeln von Problemlösungsstrategien im Vordergrund.

- Seminarunterlagen, Beispiel und Aufgaben:  
<https://github.com/stekhn/programming-workshop>
- Präsentation:  
<http://bit.ly/abp-programmieren>

**Fragen, Kritik und Wünsche sind jederzeit gerne gesehen!**

# Programmieren lernen

Es gibt viele Gründe Programmieren zu lernen:

- **Effizienter und produktiver arbeiten**
- **Besser kommunizieren und zusammenarbeiten**
- **Verstehen wie Software funktioniert**
- **Neue Problemlösungsstrategien lernen**
- **Neue Chancen im Job**

Programmieren lernen ist ein Prozess. Dieser Prozess erfordert Zeit und ist im Idealfall nie abgeschlossen.

# Programmierer im Newsroom

Ein kleiner Überblick wo Entwickler im Medienbereich arbeiten:

- **Frontend-Entwickler und UI-Entwickler**
- **App-Entwickler (Mobile Developer)**
- **Backend-Entwickler**
- **Dev-Ops und Administratoren**
- **Data Scientist und Datenjournalisten**

In anderen Bereichen typische Jobs sind außerdem: Desktop-Entwickler, Game Developer, Datenbank-Entwickler, Embedded-Entwickler, Fach-Experten wie Wirtschaftsinformatiker, Bioinformatiker und Geoinformatiker

# Programmiersprachen

Populärer Programmiersprachen nach Anwendungsfall:

- **Frontend-Entwicklung:** JavaScript
- **Backend-Entwicklung:** JavaScript, Python, Ruby, Java, PHP, Scala, Go
- **Desktopanwendung:** C++, C#, Objective-C, Java, Python
- **Mobile Entwicklung:** Swift, Java, C#
- **Spieleentwicklung:** C++, C#
- **Datenanalyse:** R, SAS, Matlab

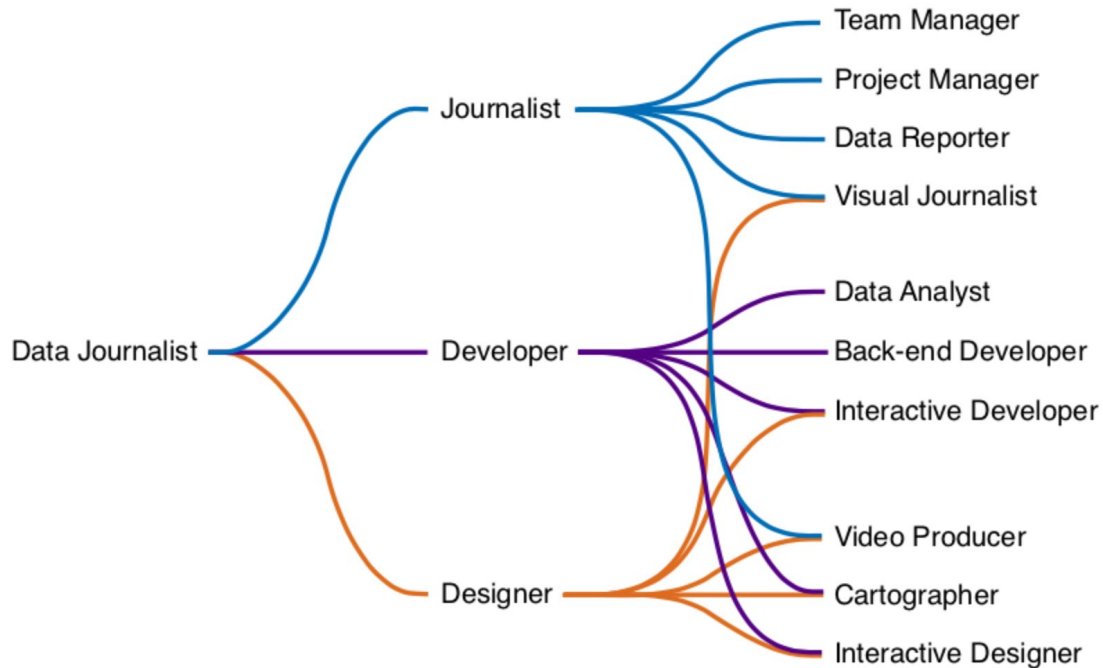
Ein anderes Konzept ist die Kategorisierung nach Paradigma: Es gibt imperative, deklarative, objektorientierte und funktionale Programmiersprachen.

# Programmiersprachen für Journalisten

Für Journalisten sind jedoch vor diese Programmiersprachen relevant:

- **JavaScript:** Entwicklung von interaktiven Webseiten, Grafiken und Anwendungen. Meist in Verbindung mit HTML und CSS. Backend-Entwicklung mit Node.js.
- **Python:** Scrapen, Analysieren und Aufbereiten von Daten verwenden wird. Größere Web-Anwendungen mit Frameworks wie Django.
- **Ruby:** Ähnlich wie Python für verschieden Aufgaben im Bereich Datenverarbeitung geeignet. Größere Web-Anwendungen mit Ruby on Rails.
- **PHP:** Ältere Programmiersprache, oft für CMS-Lösungen wie Wordpress.
- **R:** Statistische Analysen und Erstellen einfacher Diagramme und Karten.

# Rolle von Datenjournalisten



Die Rolle des Datenjournalisten hat sich in den letzten Jahren stark ausdifferenziert – vor allem in großen Medienunternehmen.

# Aufgaben von Datenjournalisten

Datenjournalisten haben vielfältige Aufgaben:

- Scraping, Data-Mining
- Datenanalyse
- Recherche-Werkzeuge
- Longreads und Storytelling-Projekte
- Interaktive Grafiken und Anwendungen
- Web-Analytics



# Neue Aufgaben für Datenjournalisten

Datenjournalismus bleibt ein spannendes Beschäftigungsfeld, da ständig neue Aufgaben hinzukommen.

- Künstliche Intelligenz (Machine Learning) nutzen und erklären
- Dokumentenanalyse (Natural Language Processing)
- Bild- und Videoanalyse (Computer Vision)
- Algorithmen bewerten (Algorithmic Accountability)
- Automatisierung (Automated Journalism)
- Internationale Projekte koordinieren (\*leaks, \*papers)
- Analytics und Business Intelligence

# Datenquellen im Internet

- **Datenbanken:** Verschiedene Datenbanktypen mit unterschiedlichen Anfragesprachen. Beispielsweise MySQL, MongoDB, SQLite, DB2.
- **APIs:** Programmierschnittstellen um an Daten (einer Datenbank) zu kommen. Beispielsweise Facebook, Twitter, Wahlergebnisse.
- **RSS-Feeds:** regelmäßig aktualisierter Datenstrom. Wird oft von Nachrichtenseiten angeboten.
- **Fileserver:** Datensammlungen die entweder über eine Index oder beispielsweise einen FTP-Server zugänglich sind.
- **Webseiten:** Oft werden Daten auch einfach als HTML dargestellt. Häufigster Anwendungsfall für einen Web Scraper.

# Git und Github

**Git** und **Github** werden oft verwechselt:

- **Git** ist eine freie Software zur verteilten Versionsverwaltung von Dateien
- **Git** ermöglicht es nachzuvollziehen, wer wann was an einer Datei geändert hat
- **Git** eignet sich hervorragend für die kollaborative Software-Entwicklung
  
- **Github** ist eine Online-Plattform zum Teilen von (Software-)Projekten
- **Github** baut auf Git auf und ist für Privatpersonen kostenlos
- **Github** hostet einige der bekanntesten Open-Source-Projekte

# Moderne Softwareentwicklung

Wichtige Grundsteine des modernen Softwareentwicklungsprozess sind:

- **Agile Methoden und Prozesse:** Scrum, Kanban, User Stories, MVPs
- **Modulare Systeme:** Viele Microservices mit klar definierten Aufgaben und Standardschnittstellen statt einem großen Softwarepaket (Monolith)
- **Iterative Entwicklung:** Software in kleinen Schritten nutzerzentriert weiterentwickeln statt große Konzepte umsetzen
- **Continuous Delivery:** Neue Features werden bei Fertigstellung ausgerollt
- **Selbstorganisierte Teams:** Flache Hierarchies, flexible Arbeitszeiten, Entscheidungsfreiheit und feste Rituale (Daily, Weekly, Retro)

# Agile & Scrum

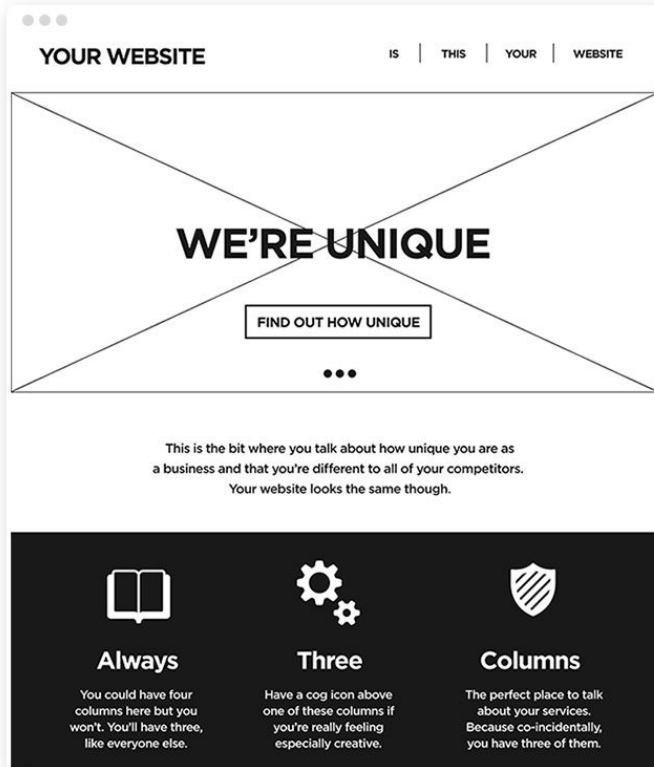
Die Leitsätze der agilen Software-Entwicklung sind:

- **Individuen und Interaktionen** stehen über Prozessen und Werkzeugen
- **Funktionierende Software** steht über einer umfassenden Dokumentation
- **Zusammenarbeit mit dem Kunden** steht über der Vertragsverhandlung
- **Reagieren auf Veränderung** steht über dem Befolgen eines Plans

**Scrum** ist eine Umsetzung dieser Leitsätze und beschreibt einen Software-Entwicklungsprozess in festen (zwei Wochen) Zyklen:

- Planning am Anfang, tägliches Daily, Review und Retro am Ende

# Modernes Webdesign



Hier ein paar Trend:

- Flat- und Material-Design
- Mutige Typografie
- Illustrationen und Icons
- Knallige Leitfarbe
- Farbverläufe
- Scrollytelling
- Mobile First

Bild: [Webdesign Journal](#)

# Webhosting

Um eine Webseite ins Internet zu bekommen, muss man diese auf einen Web-Server hochladen:

- Für die meisten Projekte reichen ein paar hundert MegaByte Web-Space
- Web-Space, inklusive eigener Domain, kann man für wenige Euro im Monat mieten: [Domain Factory](#), [All-Inkl](#), [Hetzner](#), [Webgo](#), [Ionos](#), [Manitu](#), [Host Europe](#) uvm.
- Datei können mit über FTP oder eine Web-Oberfläche hochgeladen werden
- Gute FTP-Programme sind [FileZilla](#) (PC und Mac) und [CyberDuck](#) (Mac)
- Für Wordpress oder andere CMS-Lösung bieten die meisten Web-Hoster mittlerweile eine genannte „Ein-Klick-Installation“ an

# Kommunikation mit Programmieren

Ein paar Tipps, um besser mit Entwicklern zu kommunizieren:

- Herausforderungen geben statt Aufgaben verteilen
- Über das Problem oder die Anforderung, statt über die Lösung sprechen
- Gemeinsamen Wortschatz überprüfen und pflegen
- Nicht vor einen Computer zu setzen, sondern mit Stift und Papier in die Kantine
- Termin vereinbaren und nicht einfach so hereinplatzen, Programmierer sind gerne mal „im Tunnel“ und wollen nicht gestört werden
- Entwickler enger an den Newsroom (und dessen Ziele) anbinden
- Entwickler verstehen oftmals auch nicht, was Journalisten eigentlich machen



# Erfolg messen

Mit Webanalyse-Tools kann man nicht nur den Erfolg eines Projekts messen, sondern auch lernen, wie die Benutzer mit der Seite interagieren (Aufenthaltszeit, Klicks, Interaktionen, Absprungrate).

Die verbreitetsten Webanalyse-Tools sind:

- Google Analytics: [https://www.google.com/intl/de\\_de/analytics/](https://www.google.com/intl/de_de/analytics/)
- Piwik: <https://piwik.org/>

Google Analytics ist kann datenschutzrechtlich bedenklich sein. Piwik hingegen kann man auf den eigenen Servern installieren, was deutlich besser ist.

# Probleme lösen

Programmieren kann oftmals frustrierend sein, vor allem wenn man ein scheinbar unlösbares Problem hat. Dabei hilft:

- Pause machen oder einfach mal drüber schlafen
- Das Problem einem Kollegen (oder eine Badeente) Schritt für Schritt erklären
- Problematischen Code-Abschnitt isolieren, zum Beispiel in eine neue Datei kopieren und schauen, ob das Problem immernoch auftritt
- Versuchen das Problem in kleinere Teilprobleme herunterzubrechen
- Bereits existierende Lösungen eingehend studieren und vergleichen
- Problem, nach eingehender Internetrecherche, bei [Stack Overflow](#) einstellen

# Weiterlernen

Wer seine Fähigkeiten im Bereich Web-Entwicklung weiter ausbauen oder Kursinhalte wiederholen möchte wird hier fündig. Die meisten Online-Kurse sind kostenlos:

- Knight Lab: [Digital Journalism, Bit by Bit](#)
- Khan Academy: [Computer Programming](#)
- MDN: [Learn Web Development](#)
- FreeCodeCamp: [Learn to Code](#)
- Codecademy: [Web Development](#)
- Udemy: [Front-End Development Foundations](#)

# Vielen Dank!

[s.kuehne@onlinehome.de](mailto:s.kuehne@onlinehome.de)

[stekhn.de](http://stekhn.de)

[twitter.com/stekhn](https://twitter.com/stekhn)

[linkedin.com/in/stekhn/](https://linkedin.com/in/stekhn/)

[xing.com/profile/Steffen Kuehne11](https://xing.com/profile/Steffen_Kuehne11)