

## Framework

### Was ist ein Framework?

Was ist ein Framework? Das Framework ist insbesondere in der Software-Entwicklung ein fester Begriff. Es steht für ein Grundgerüst, das dem Programmierer bestimmte vorgefertigte Module und Funktionen bietet. Diese kann er durch einfache Befehle in sein Programm einbinden. Der Vorteil ist, dass wiederkehrende Aktionen, etwa das Öffnen und Schließen von Dateien, nicht von jedem Entwickler neu erstellt werden müssen. Je nach Ausführung des Frameworks, ist es allerdings unter Umständen erforderlich, eine entsprechende Laufzeitumgebung auf dem PC des Nutzers zu installieren.

### Entstehung und konkrete Anwendung von Frameworks

Die Idee, Softwareprogrammierung durch Frameworks zu erleichtern, ist schon mehrere Jahrzehnte alt. Eines der bekanntesten und ältesten Frameworks ist das [“.NET Framework“](#) von Microsoft (gesprochen: Dotnet-Framework). Dieses wurde bereits 2002 als Bestandteil der Entwicklungsumgebung “Visual Studio” veröffentlicht.

Frameworks können in unterschiedlichen Bereichen zur Anwendung kommen. Häufige Einsatzgebiete sind etwa:

- Allgemeine Anwendungsentwicklung
- Webentwicklung
- [Content Management Systeme \(CMS\)](#)
- Entwicklung grafischer Oberflächen
- Größere Eigenentwicklungen von Unternehmen
- Testumgebungen

Entwicklungsumgebungen sorgen für eine Vereinheitlichung und Modularisierung von Software. Für die eingebundenen Funktionen bieten sie genormte Schnittstellen, die sogenannten Application Programming Interfaces (APIs). Der Zugriff auf eine solches API ist genau festgelegt. Hierdurch wird der Code übersichtlicher und leichter zu warten. Es ist beispielsweise möglich, ein [Update](#) für ein Modul einzuspielen, ohne die darauf zugreifenden Programme verändern zu müssen.

Da gerade in großen Softwareprojekten eine Vielzahl an Entwicklern zusammenarbeiten, vereinfacht die

Modularisierung zudem eine Aufteilung in verschiedene Arbeitspakete. Ansonsten müssten sich alle Programmierer permanent absprechen, wer welche allgemeinen Funktionen fertigt, um diese nicht mehrfach parallel für identische Zwecke im Code zu haben. Zudem hätte jeder Programmierer eine umfassende Dokumentation für von ihm gefertigte Module zu erstellen, damit andere Entwickler damit arbeiten könnten.

## Wie funktioniert ein Framework?

Eine Entwicklungs- oder Laufzeitumgebung stellt sozusagen eine weitere Schicht zwischen dem Betriebssystem und der darauf ausgeführten Software dar. Sie fasst komplexere Befehle auf niedriger Ebene in einfache Anweisungen zusammen. Man spricht hierbei auch von einer Abstraktion. Insbesondere im Zusammenhang mit objektorientierter Programmierung spielen Laufzeitumgebungen eine große Rolle.

Beispielhaft könnte ein Framework etwa zur einfachen Anzeige grafischer Elemente bestimmt sein. Für die Darstellung eines Fensters mit dem "OK" -Button und dem "Abbrechen"-Button wäre dort das Objekt "Beispielframework.MessageBox" vorgesehen. Der Programmierer hätte nun die Möglichkeit, dieses Objekt zu verwenden. Dazu würde er es initialisieren und bekäme ein Objekt der Klasse "MessageBox" zurück. Dieses hätte automatisch bereits mehrere Funktionen implementiert. So könnte der Programmierer den anzuzeigenden Text übergeben und bekäme abhängig von der Nutzereingabe einen Wert zurückgemeldet. Üblich wäre die Zahl "1" für die Betätigung des Buttons "OK" und die Zahl "0" für "Abbrechen". Im Anschluss wäre es dem Entwickler möglich, mit diesem Wert weiterzuarbeiten.

Ein fiktiver Beispielcode würde wie folgt aussehen:

```
Abfrage = Beispielframework.MessageBox("Wollen Sie fortfahren?", "OK", "Abbrechen")
```

```
Auswahl = Abfrage.Show()
```

```
if Auswahl == 1:
```

```
    Fiktive Anweisung
```

```
else:
```

```
    break
```

In wenigen Programmzeilen kann auf diese Weise mit dem Beispielframework eine bereits recht komplexe Aktion umgesetzt werden.

## Welche bekannten Frameworks gibt es?

Inzwischen ist eine Vielzahl an Frameworks auf dem Markt erhältlich. Diese verfolgen teilweise ganz unterschiedliche Anwendungszwecke.

## **.NET-Framework**

Aufgrund der hohen Verbreitung von Microsofts Betriebssystem Windows, ist auch das bereits erwähnte .Net-Framework sehr häufig genutzt. Es unterstützt eine Vielzahl an Programmiersprachen, darunter Visual Basic, C# und C++/CLI. Entwickler können damit insbesondere Apps entwickeln, etwa für die Konsole oder mit grafischer Oberfläche. Aber auch für eigene Dienste lässt .NET sich einsetzen. Die erforderliche Laufzeitumgebung stellt Microsoft in einer Version für Entwickler und einer Variante für Anwender zur Verfügung. Letztere ist erforderlich, um mit .NET entwickelte Programme ausführen zu können.

## **Django**

Bei [Django](#) handelt es sich um ein Webframework, welches in der Programmiersprache Python umgesetzt wurde. Entsprechend lassen sich damit vielseitige und dynamische [Webseiten](#) erstellen. Ein besonderer Fokus wird dabei auf die Sicherheit gelegt. Es bietet unter anderem Funktionen, um auf SQL-Datenbanken zuzugreifen.

## **Bootstrap**

[Bootstrap](#) ist dafür gedacht, Nutzeroberflächen von Webseiten (Frontends) zu gestalten. Es verwendet dabei [Templates](#), die mittels [HTML](#) und [CSS](#) umgesetzt werden. Mit Bootstrap kann ein Webentwickler einfach Elemente wie zum Beispiel Sidebars (Randfenster), Navigationsbars oder Anmeldefenster realisieren.

## **Angular.js**

Die Umgebung [Angular.js](#) basiert auf [JavaScript](#). Entsprechend werden damit erstellte Programme auch wie JavaScript in Webseiten eingebunden. Hierdurch lassen sich auf relativ einfache Weise vorgefertigte dynamische Inhalte in ansonsten statische Strukturen integrieren, etwa Formulareingaben.

## **Qt**

Ein sehr vielseitiges Framework ist das für den Aufbau grafischer Oberflächen verwendete [Qt](#). Eine Besonderheit ist seine Universalität. So lässt es sich auf verschiedenen Betriebssystemplattformen verwenden und unterstützt diverse Programmiersprachen, zum Beispiel C++, Python und JavaScript. Mit Qt sind umfangreiche Benutzeroberflächen umsetzbar. Zur Verfügung stehen dafür Elemente wie Checkboxes, Scrollbalken, Eingabefelder und Tabs, wie man sie von [Browsern](#) kennt.

## Vor- und Nachteile

Die Nutzung von Frameworks hat sich sowohl in der Anwendungsentwicklung als auch in der Webentwicklung etabliert. Bei kritischer Betrachtung gibt es dennoch ein paar Argumente, die den zweifellos vorhandenen Vorteilen gegenüberzustellen sind.

### Vorteile

Flexibilität  
Bessere Erweiterbarkeit  
Sicherheit, bei Verwendung getesteter Module

Vereinfachung für Entwickler

### Nachteile

Abhängigkeit von Drittanbietern  
Festlegung auf einen Anbieter notwendig  
Sicherheitslücken betreffen eine Vielzahl an Programmen  
Individualität geht verloren

## Entwicklung und Ausblick

Hinter den aufgeführten Frameworks und etlichen weiteren stehen große Unternehmen oder langjährig existierende Organisationen. Daher ist davon auszugehen, dass eine Fortentwicklung in den meisten Fällen auf Jahre garantiert ist. Insofern ist für solche Frameworks auch zu erwarten, dass regelmäßige Updates und Funktionserweiterungen zur Verfügung gestellt werden.