

# Chat App selber programmieren - Start Programmieren

Nun starten wir mit dem Projekt, selbst eine Chat-App zu programmieren. Die fertige Version wird folgendes können:

- Nachrichten versenden und empfangen in einem universellen Chat-Raum (Jeder kriegt alle Nachrichten)
- Die Chat-Apps aller Gruppen können am selben Chat teilnehmen (Trennung Front- und Backend)
- Benutzernamen festlegen für das Versenden von Nachrichten

Mögliche Erweiterungen sind:

- Login Funktion (z.B. "Login mit Google")
- Direktnachrichten
- Verschiedene Chat-Räume

Zum Start wird normalerweise erst einmal das HTML programmiert und mit "Testdaten" gearbeitet. Testdaten sind normalerweise im JavaScript "hardcoded", also nicht dynamisch. In Ihrer Vorlage sind bereits einige Testnachrichten drin:

```
let messages = [  
  {  
    author: "Ding",  
    message: "Hallo zusammen"  
  },  
  {  
    author: "Dong",  
    message: "Hallo zurück"  
  }  
]
```

Zu einem späteren Zeitpunkt werden wir diese Daten durch die echten Daten vom Server ersetzen.

Für den Moment reicht es aber aus, um eine erste Version der App zu basteln.

Die App soll für die erste Version folgendes machen:

- Die Testdaten aus der Vorlage in einer Liste in der App anzeigen
- Eingegebene Nachrichten in die Liste der Testdaten hinzufügen und danach auch im HTML anzeigen

Vieles davon ist sehr ähnlich zur Funktionalität der Rezept-Kommentarspalte, die wir schon programmiert haben.

In der letzten Stunde haben wir bereits eine erste Version unserer Benutzeroberfläche (englisch: User interface, kurz: UI) erstellt. Heute beginnen wir nun mit der Programmierung unserer Chat-app.

## Arbeitsweise

Mit den Github-Repositories sind wir bestens vorbereitet, um gemeinsam in Teams an einem Projekt zu arbeiten. In echten Projekten gibt es zwei grundlegende Methoden, um gemeinsam zu Programmieren

Im **Pair-Programming** arbeitet man gemeinsam an einem Laptop. Eine Person übernimmt das schreiben, es denken aber beide mit und tragen zum Code bei. So hat man ein eingebautes "4-Augen Prinzip" und kann Fehler besser erkennen. Sie kennen diese Arbeitsweise bereits vom letzten Projekt.

Die andere Möglichkeit ist es, das jede:r am eigenen Laptop arbeitet. Danach werden die Änderungen über Github zusammengeführt. Dies ist etwas schwieriger zu koordinieren, da man gut besprechen muss, wer an welchem Teil arbeitet.

Wie sie im Team arbeiten möchten ist Ihnen überlassen. Für den Workflow mit Git finden Sie im Teams ein Dokument mit Beispielen und den wichtigsten Kommandos.

## Auftrag 8.05.23

Wie bereits erwähnt ist das Ziel für die heutige Stunde folgendes:

- Die Testdaten aus der Vorlage in einer Liste in der App anzeigen
- Eingegebene Nachrichten in die Liste der Testdaten hinzufügen und danach auch im HTML anzeigen
- Falls Zeit: Erstes Styling der App

Die Kernelemente vom JavaScript, die sie dafür benötigen kennen wir bereits von der Rezept Webseite. Hier aber noch mal eine kurze Repetition:

**addComment** → Durch Knopfdruck "Absenden" ausgelöst

```
function addComment() {  
  // Inhalt der Eingabefelder holen  
  text = document.getElementById("comment").value  
  username = document.getElementById("username").value  
  // Check ob eines von beiden leer ist  
  if (username == "" || text == "") {  
    return  
  }  
  // Neuer Kommentar in die Kommentarliste speichern  
  comments.push({  
    text: text,  
    username: username  
  })  
  // updateComments() auslösen  
  updateComments()  
  // Eingaben Löschen  
  document.getElementById("comment").value = ""  
}
```

**updateComments** → Von `addComment` ausgelöst

```
function updateComments() {  
  // Behälter der Kommentar liste vom HTML holen  
  let kommentarListe = document.getElementById("kommentare")  
  // Inhalt der Kommentarliste entfernen  
  kommentarListe.innerHTML = ""  
  // Inhalt der neuen Kommentarliste erstellen  
  kommentareHTML = ""  
  comments.forEach(comment => {  
    kommentareHTML += `<li class="comment">${comment.text} - ${  
comment.username}</li>`  
  });  
  // Neuer Inhalt speichern  
  kommentarListe.innerHTML += kommentareHTML  
}
```

<!-- Eine JS Funktion an einen HTML-Button knüpfen -->

<button onclick="addComment()">Absenden</button>