# Development Fundamentals

## Klonen des Repositories

### Einleitung

Diese Einleitung beschreibt die Schritte, die erforderlich sind, um ein Git-Repository auf deinem lokalen Rechner zu klonen. Das Klonen eines Repositories ermöglicht es dir, die vollständige Historie des Projekts, inklusive aller Dateien, Verzeichnisse und Commit-Historie, auf deinem lokalen System zu haben.

### Voraussetzungen

Git muss auf deinem Rechner installiert sein.

Du benötigst den Repository-URL, um das Repository zu klonen.

### Ausführung

**Schritt 1:** Auf der Seite von GitHub klickt man auf den Grünen Knopf mit der Beschriftung «Code».

**Schritt 2:** Dort kann man dann entweder den generierten HTTPS- oder den SSH-Link kopieren.

**Schritt 3:** Anschliessen muss man CMD oder Git-Bash im gewünschten Ordner öffnen.

**Schritt 4:** Als Befehl schreibt man $ git clone «kopierter Link» und drück Enter.

### Meine Erfahrung

Ich hatte keine Probleme. Das Einzige war, dass ich im Nachhinein den Ablageordner ändern musste, da ich es nicht angegeben habe.

## Einrichtung der Entwicklungsumgebung

### Einleitung

Da Sie nun eine Anwendung haben, können Sie die erforderlichen Docker-Assets erstellen, um Ihre Anwendung zu containerisieren.

Sie können die in Docker Desktop integrierte Funktion Docker Init verwenden, um den Prozess zu optimieren, oder Sie können die Assets manuell erstellen.

## Erstellung der README.md

### Einleitung

Sie ist in der Regel die erste Datei, die Benutzer sehen, wenn sie ein Repository besuchen, da GitHub, GitLab und andere Plattformen sie automatisch auf der Startseite eines Repositories anzeigen.

### Meine Erfahrung

Das README.md wird standardmässig bei der Erstellung erstellt.

Mit Markdown kenne ich mich schon ein bisschen aus und kann mit dem Cheat Sheet kreativ arbeiten.

## Verwendung von Git (Commit, Push)

Git ermöglicht es Entwicklern, den Fortschritt ihrer Projekte nachzuverfolgen und Änderungen zu speichern. Zwei der wichtigsten Befehle, die du bei der Arbeit mit Git verwenden wirst, sind **git commit** und **git push**. Diese Befehle helfen dir, Änderungen zu speichern und sie in ein zentrales Repository hochzuladen.

1. **git add**: Bereite Änderungen für den Commit vor, indem du sie zur Staging Area hinzufügst.
2. **git commit**: Speichere die Änderungen lokal und füge eine Beschreibung hinzu.
3. **git push**: Lade die lokal gespeicherten Änderungen in das Remote-Repository hoch.

Diese Git-Befehle helfen dir dabei, effizient in einem Team zu arbeiten und den Fortschritt deines Projekts zu verwalten.

## Erstellung und Nutzung von Docker-Containern

Nach der Konfiguration, muss man es nur noch ausführen.

1. Man geht wieder in den CMD oder Git-Bash
2. Dort tippt man:

$ docker compose up --build

1. Die Applikation sollte ausgeführt werden.
2. Wenn alles geklappt han kann man auf [localhost:3000](http://localhost:3000/)
3. Zum schliessen gilt

$ docker compose down

1. Und dann wieder zu öffnen

$ docker compose up