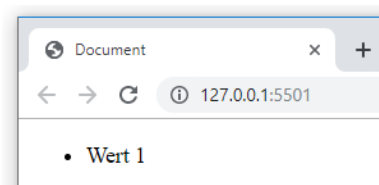


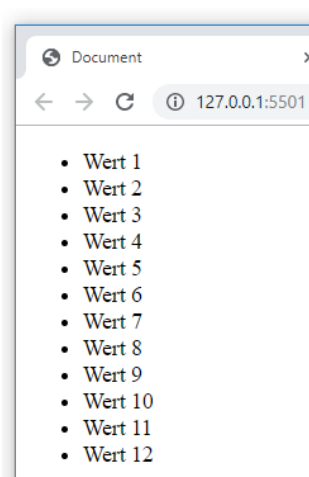
Übungsbeispiele

Verwende das zur Verfügung gestellte Projekt **Start 1**. Ändere die HTML Datei nur, wenn dies explizit erwähnt wird. Schreibe sonst den gesamten Code in die Datei.

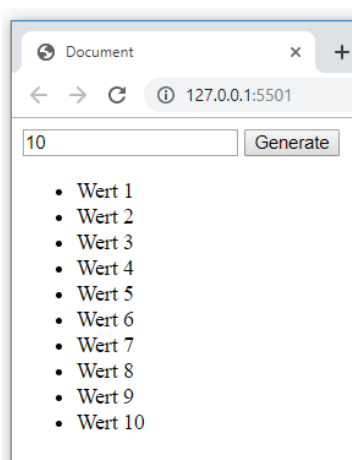
1. Erstelle JavaScript Code in **app.js**, durch den ein Listitem - wie unten stehend - zu der unordered List hinzugefügt wird.



2. Füge mittels einer Schleife 12 Listitems - wie unten stehend - hinzu. Verwende ein Template Literal für den String!



3. Erstelle in der index.html Datei ein Eingabefeld und einen Button. Nun können die Benutzer angeben, wie viele Zeilen sie benötigen. Ändere **app.js** so, damit dies klappt.



4. Füge einen Button **Delete** in der **index.html** hinzu. Jedes Mal wenn der Button gedrückt wird, soll der letzte Wert aus der Liste entfernt werden. Achte darauf, dass nichts gelöscht wird, wenn keine **li** Elemente mehr da sind!

Tipp: Verwende die Methoden **hasChildNodes()** und **removeChild()**, sowie die Eigenschaft **lastChild** der **ul**. Ermittle **lastChild** und übergib es dann an **removeChild**.

5. Es wäre doch toll, wenn man den Button einsparen könnte. Baue das Programm so um, dass, sobald die Benutzer eine Zahl **n** eingegeben haben, die Liste **Wert 1 ... Wert n** in der **ul** dynamisch angezeigt wird. Eine Live View sozusagen! Jeder Tastendruck lässt die Liste neu entstehen. Entferne auch die beiden Buttons!

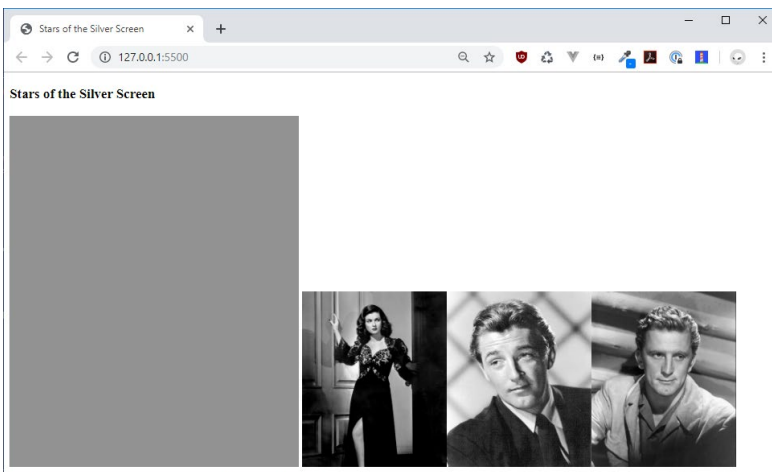
Tipp 1: Verwende den Event **keyup** für das Eingabefeld. Frage dann den Wert des Eingabefeldes ab.

Tipp 2: Lösche die alte Liste bevor du die neue hinzufügst. Siehe Beispiel 4 und mache eine Schleife! Eine nicht empfohlene Art wäre: `innerHTML=""` zu schreiben!

6. Verwende das zur Verfügung gestellte Projekt **Start 2**. Ändere die HTML Datei nur, wenn dies explizit erwähnt wird. Erzeuge die drei Bilder in **app.js** dynamisch. Setze auch die Größe auf 200px via **app.js**.

Tipp 1: Attribute wie **src**, **width** können mit den Methoden **setAttribute/getAttribute** angesprochen werden.

Tipp 2: Verwende eine Funktion, um ein Bild zu erzeugen.



7. Bewegt sich die Maus über das kleine Bild, wird es auf dem großen Bild angezeigt.

Tipp: Hänge noch einen Eventhandler für **mouseover** an das Bild, welches dem grauen Bild die richtige Source zuweist. Ebenso setze für **mouseout** wieder das graue Bild. Verwende den Parameter für den Mausevent um herauszufinden, wo sich gerade der Cursor befindet.

