

Getting started with Alpine.js

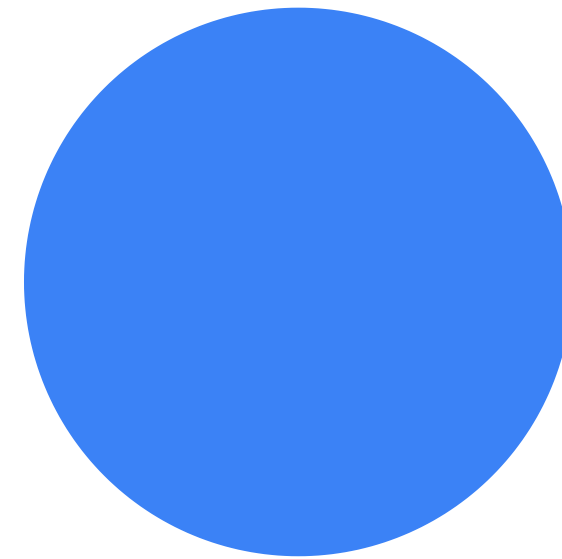
Workshop für das Modul Weaving the Web

Hallo zusammen

TIM LOGES

BACHELOR MEDIENINFORMATIK AN DER TH KÖLN

AKTUELL IM MASTER MEDIENINFORMATIK



Was ist Alpine.js?

- Frontend JS-Framework für (einfache) Logik
- Logik lebt im Markup (Geschmacksache)
- Reaktive, ohne die "Kosten" der großen Brüder wie Vue & React

Wann greife ich zu Alpine.js

Erweitern von bereits bestehenden Seiten

Ein- und Ausblenden von DOM-Elementen

Styling basierend auf JS-Logic (Class binding)

Two-Way binding

Bonus: Transitions

Und wann nicht

Große Projekte mit viel Logik (Auslagern unnötig kompliziert)

SPA (kein Konzept für Routing)

Kein virtual Dom (Performance)

~~Dokumentation nur README~~

CDN

oder

Bundler

Script Tag

import 'alpinejs'

Alpine.js initialisiert sich selbst*

*ab V3 muss Alpine.js initialisiert werden

14 Directives

x-data

x-init

x-show

x-text

x-html

x-bind

x-on

x-ref

x-model

x-if

x-for

x-transition

x-spread

x-cloak

6 Magic Properties

\$el

\$refs

\$event

\$dispatch

\$nextTick

\$watch

Directive ``x-data``

- Neuer "scope"
- Daten nur innerhalb der Kinder verfügbar
- Objekt!

BENUTZUNG

```
<div x-data="{ name: 'wert' }"> ... </div>
```

Alle anderen Directives brauchen mindestens ein ``x-data``. Möchte man ein Directive nutzen, hat aber keine Daten, kann ``x-data`` auch mit einem leeren Object bzw. sogar komplett ohne gesetzt werden.

Directive ``x-init``

- wird nach Initialisierung ausgeführt
- Ausdruck (expression) oder Funktion möglich

BENUTZUNG

```
<div x-data="{ ... }" x-init="[expression]"></div>
```

```
<div  
  x-data="{ name: 'wert' }"  
  x-init="name = andererWert">  
</div>
```

```
<div  
  x-data="{ name: 'wert' }"  
  x-init="setTimeout(function(){ alert(name); }, 3000);">  
</div>
```

Ab V3 kann ``x-init`` innerhalb von ``x-data`` deklariert werden und wird automatisch ausgeführt.

Directive `x-show`

- Blendet Elemente ein und aus
- Setzt `display: none` entsprechend der expression
- Bonus: Transitions

BENUTZUNG

```
<div x-show="[expression]"></div>
```

```
<div x-data="{ open: false }">  
  <button x-on:click="open = true">Zeige dich</button>  
  <p x-show="open">  
    Ich bin anfangst nicht sichtbar  
  </p>  
</div>
```

Je nach Browser & Hardware kann es vorkommen das Alpine.js länger zum initialisieren braucht als der Browser zum Rendern. Daher ist es sinnvoll Elemente initial auszublenden um flackern zu verhindern.

```
style="display: none"
```

Directive `x-show` Transitions

- Einfache Animationen
- Kann individualisiert werden

BENUTZUNG

```
<div x-show.transition="[expression]"></div>
```

```
<div x-data="{ open: false }">  
  <button x-on:click="open = true">Zeige dich</button>  
  <p x-show.transition="open">  
    Ich bin anfangst nicht sichtbar  
  </p>  
</div>
```

```
<div  
  x-show.transition.scale.opacity.delay.50ms  
>
```

Ab V3 ist `x-show.transition` nur noch über Directive `x-transition` nutzbar!

Directive `x-text`

- bindet `innerText` von DOM Elementen

BENUTZUNG

```
<span x-text="[expression]"></span>
```

```
<div x-data="{ msg: 'Hallo' }">  
  <span x-text="msg + ' zusammen'">  
    <!-- Hallo zusammen -->  
  </span>  
</div>
```

Directive `x-html`

- entfernt in V.3
- bindet `innerHTML` von DOM Elementen

BENUTZUNG

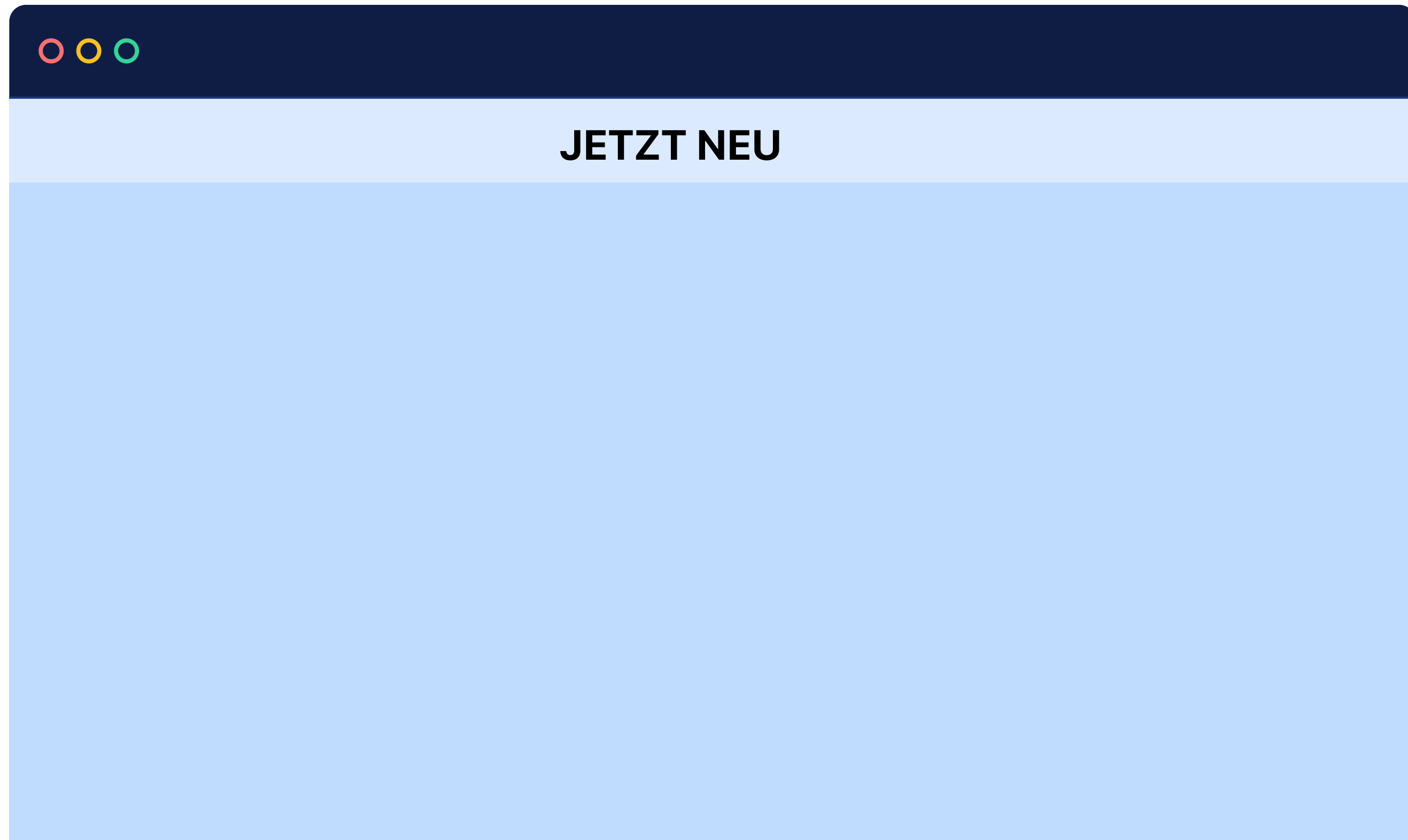
```
<span x-html="[expression]"></span>
```

Übung Nr. 1

Site-Banner mit wechselndem Text

Übung Nr. 1

Site-Banner mit wechselndem Text



Übung Nr. 1

Site-Banner mit wechselndem Text

- ``x-data`` neuer scope
 - ``x-init`` wird ausgeführt nach der Initialisierung
 - ``x-show`` ein- und ausblenden von Elementen
 - ``x-text`` bindet ``innerText``
-
1. ``aufgabe-1.html`` in ``aufgaben/aufgabe-1`` öffnen
 2. p-tag ein- bzw. ausblenden
 3. Bonus: Transitions

Dauer 20 Minuten

Directive `x-bind:TYPE`

- Setzt HTML Attribute
- Für `value` Attribute als auch für `boolean` Attribute

BENUTZUNG

```
<div x-bind:[attribute]="[expression]"></div>
```

```
<div x-data="{ isDisabled : true }">  
  
  <button x-bind:disabled="isDisabled">Versuch es</button>  
</div>  
// isDisabled = true  
<button disabled="disabled">Versuch es</button>  
  
// isDisabled = false  
<button>Versuch es</button>
```

Shorthand: `:TYPE`

Directive `x-bind:class`

- Anderer Aufbau als `x-bind:TYPE` !
- Übergabeparameter muss Objekt sein
- Objekt Aufbau:
name: die Klasse die hinzugefügt wird
wert: boolean der angibt ob die Klasse hinzugefügt wird

BENUTZUNG

```
<div x-bind:class="{ 'className': variable }"></div>
```

⚡ Mehrere Klassen ⚡

```
<div x-bind:class="{ 'className className2': variable }"></div>
```

⚡ Verschiedene Klassen bei unterschiedlichen expression ⚡

```
<div x-bind:class="{ 'text-red': warning, 'text-green': !warning }"></div>
```

Shorthand: `:class`

Directive `x-on:EVENT`

- Event-Listener
- Kann mit jedem verfügbaren Event genutzt werden

BENUTZUNG

```
<button x-on:[event]="[expression]"></button>
```

```
<button x-on:click="isOpen = true"></button>
```

Shorthand: `@event`

Directive `x-on:EVENT` Helfer (Ausschnitt)

- `.away`: triggert Event nur wenn es ausserhalb ausgeführt wird
- `.prevent`: verhindert native Funktionalität
- `.once`: Event wird nur einmal ausgeführt
- `.debounce`: Event wird nur alle Xms ausgeführt
- `.window`: Event wird an Window-Object statt an DOM-Element registriert

```
<div x-on:click.away="isOpen = false"></div>
```

```
<button type="submit" x-on:click.prevent="dontSubmitForm()">
```

```
<div x-on:hover.once="fetchData()"></div>
```

```
<input x-on:input.debounce.750ms="fetchSomething()">
```

```
<div x-on:resize.window="resized = true"></div>
```

Directive `x-ref`

- stellt DOM-Elemente für die Nutzung mit `$refs` zur Verfügung

BENUTZUNG

```
<div x-ref="[name]"></div>
<button
  x-on:click="$refs.[name].innerText = 'Moin'"
></button>
```

```
<div x-data>
  <input x-ref="input" />
  <button x-on:click="$refs.input.value = '' ">
    Input leeren
  </button>
</div>
```

Übung Nr. 2

Password Validator

Übung Nr. 2

Password Validator



Übung Nr. 2

Password Validator

- ``x-bind`` bindet Attribute oder Klassen
 - ``x-on`` Event-Listener
 - ``x-ref`` referenziert und stellt DOM-Element zur Verfügung
 - ``$refs`` zugriff auf ``x-ref``
-
1. ``aufgabe-2.html`` in ``aufgaben/aufgabe-2`` öffnen
 2. Passwort Hinweise je nach Zustand Grün oder Rot einfärben
 3. Verhindern das Button "Account erstellen" gedrückt werden kann, sollten nicht alle Regeln eingehalten sein
 4. BONUS: Passwort zeigen/verstecken

Dauer 20 Minuten + 10 Minuten Pause

Directive `x-model`

- Two-way binding für:

- `<input type="text">`
- `<textarea>`
- `<input type="checkbox">`
- `<input type="radio">`
- `<select>`

BENUTZUNG

```
<div x-data="{ Nachricht: '' }">
  <input type="text" x-model="Nachricht">

  <span x-text="Nachricht">
</div>
```

```
<div x-data="{ Nachricht: '' }">
  <input type="text" x-model="Nachricht">

  <button x-on:click="Nachricht = ''">Reset</button>
</div>
```

Magic Prop ``$dispatch``

- Event Kommunikation
- Event wird nach "oben" weitergegeben
- ``.window`` Modifier
 - Für Kommunikation zwischen Komponenten

BENUTZUNG

```
<div x-on:custom-event="console.log($event.detail.foo)">  
  <button x-on:click="$dispatch('custom-event', { foo: 'foo' })">Klick</button>  
</div>
```

```
<div  
  x-data="{ title: 'Hello' }"  
  x-on:set-title.window="title = $event.detail"  
>  
  <h1 x-text="title"></h1>  
</div>  
  
<div x-data>  
  <button x-on:click="$dispatch('set-title', 'Hello World')">Klick</button>  
</div>
```

Alpine.js v3

Neue Dokumentation

Globales State Management

Bessere wiederverwendbarkeit

Wie kann ich mit **Alpine.js** weitermachen?

- Dokumentation
- Code with Hugo
- Alpine.js Examples
- Awesome Alpine.js
- Alpine.js Magic Helper

Übung Nr. 3

Notification System mit Events

Übung Nr. 3

Notification System mit Events



Übung Nr. 3

Notification System mit Events

- ``x-model`` Two-Way binding
 - ``$dispatch`` Events zur Kommunikation
-
1. ``aufgabe-3.html`` in ``aufgaben/aufgabe-3`` öffnen
 2. Event dispatchen wenn Button gedrückt wird
 3. Event Listener an Notification hängen
 4. Je nach Event zugehöriges Icon anzeigen und Texte für Heading und Message setzen
 5. Notification schließen wenn "close Button" gedrückt wird
 6. BONUS: Transitions

Dauer 30 - 40 Minuten