AUSWAHLKOMPONENTE

IP5 Final Presentation

Ramona Marti, Lea Burki

fhnw.ch

Hallo

Wir sinds...

Lea Burki

lea.burki@students.fhnw.ch

Ramona Marti

ramona.marti@students.fhnw.ch

Aufgabenstellung

Problem

- Begrenzte Möglichkeiten mit `select & datalist`
- Schwer umzugestalten
- Schlechtes Intraktionsdesign
- Bibliotheken mit vielen Abhängigkeiten

Ziel

- Erstellen einer Auswahlkomponente
- Ansprechendes Design
- Wiederverwendbare Komponenten
- Effiziente und benutzerfreundliche Bedienung

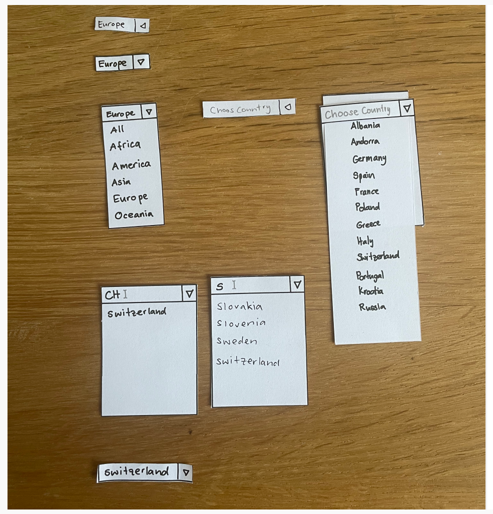
LIVE DEMO

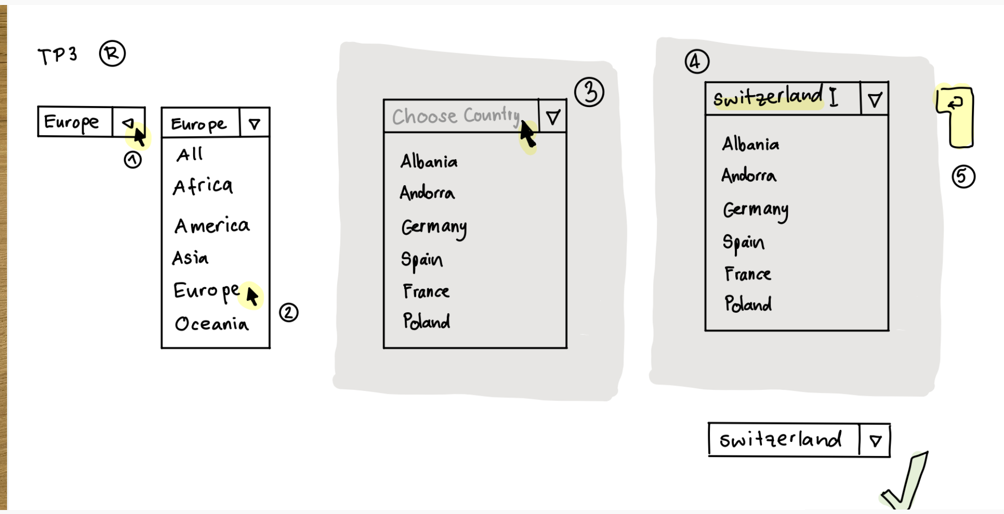
K o m p o n e n t e

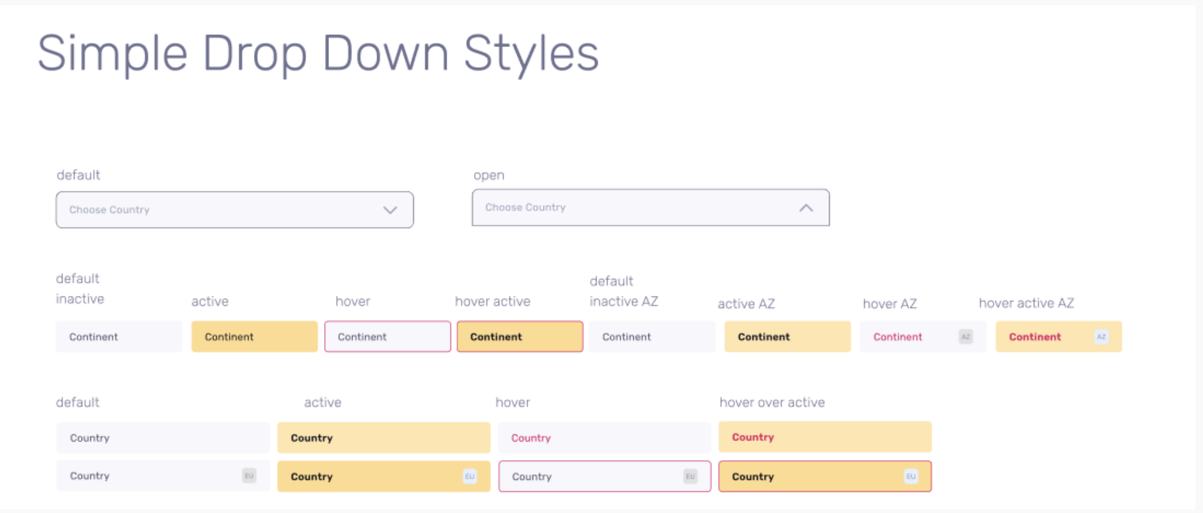
Ausgangslage

- Codebasis des Kolibri Toolkit
    - SimpleInput Komponente
    - Debounce Funnktion
- Kolibri Designsystem
- Gemeinsames Verständnis

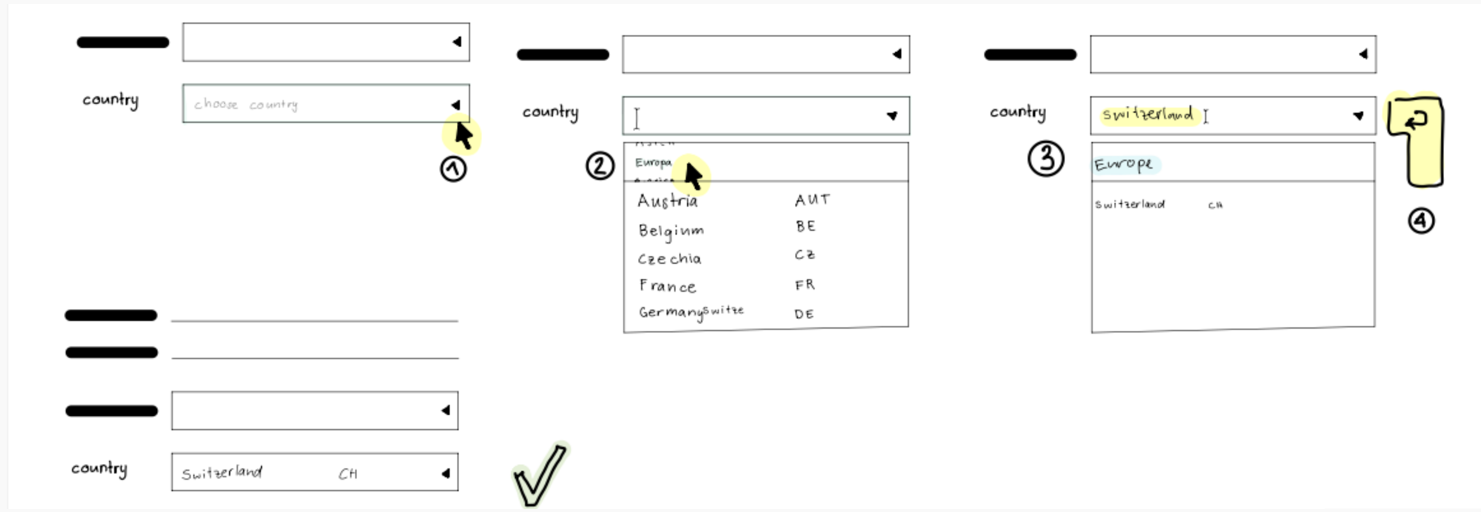
Kolibri – The Web UI Toolkit

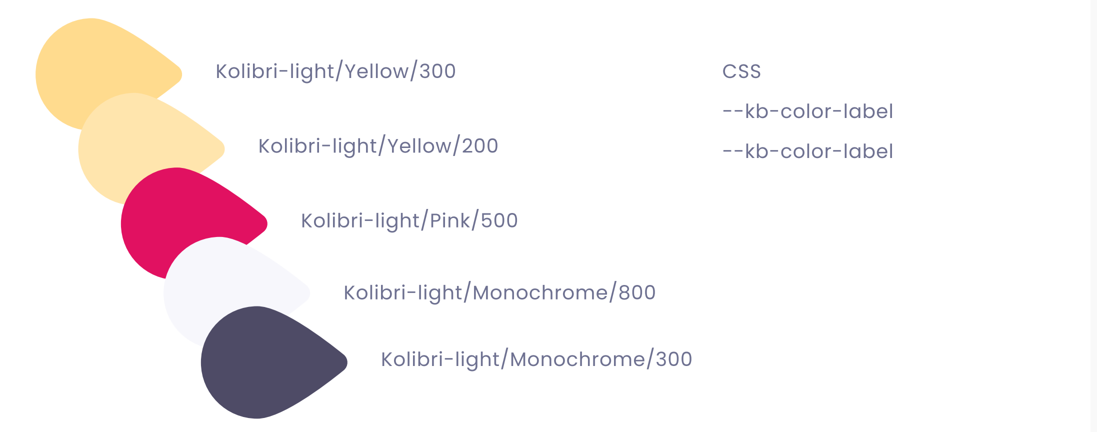


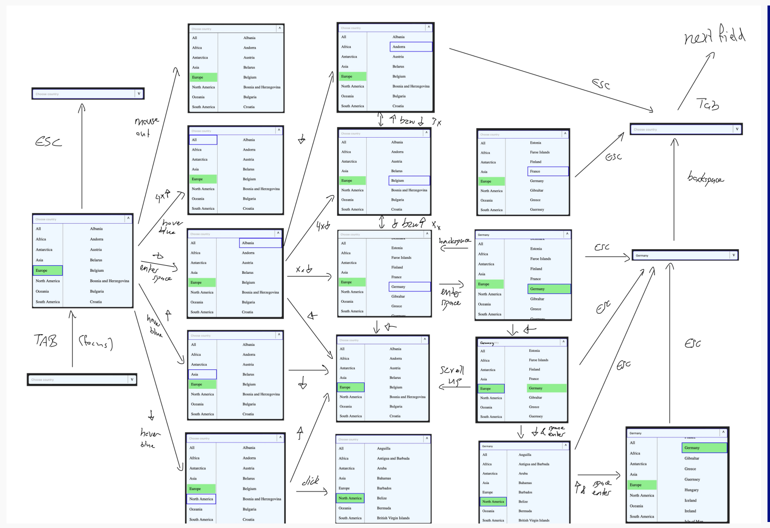


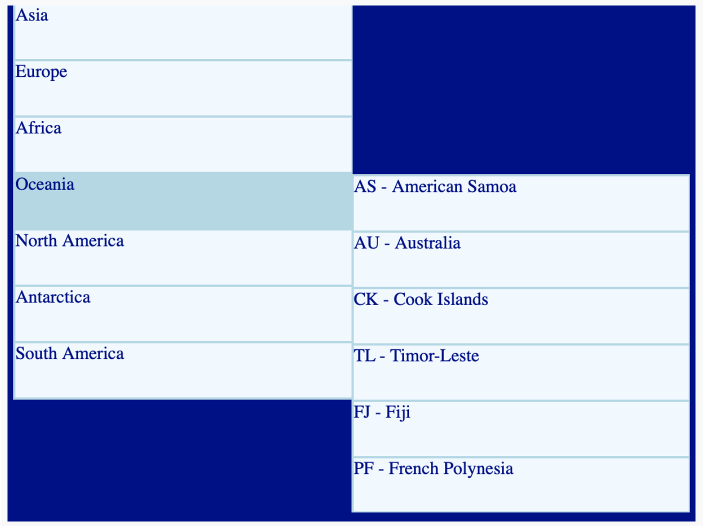


Simple Drop Down Styles



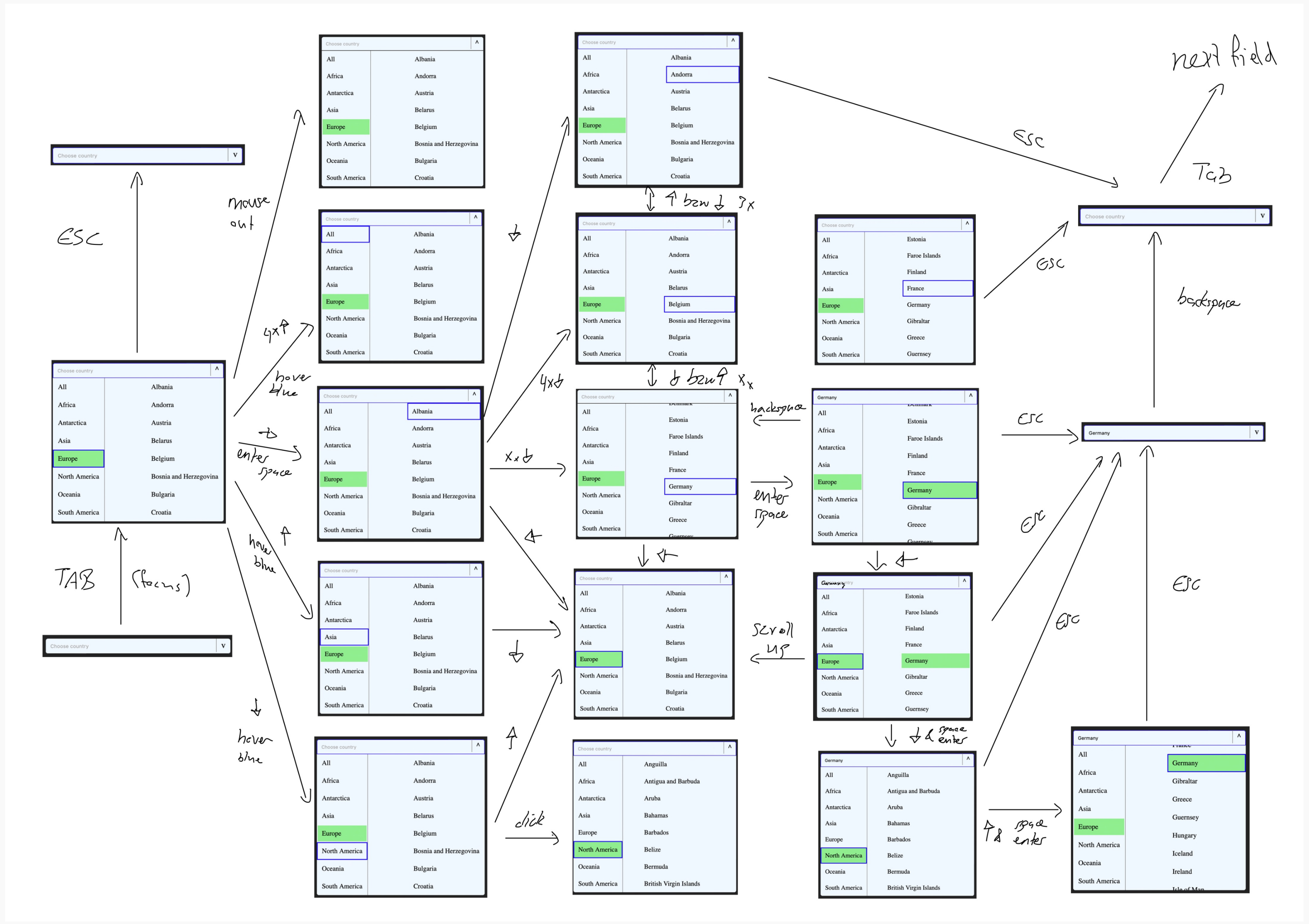






Asia	
Europe	
Africa	
Oceania	AS - American Samoa
North America	AU - Australia
Antarctica	CK - Cook Islands
South America	TL - Timor-Leste
	FJ - Fiji
	PF - French Polynesia

Design
Methodik



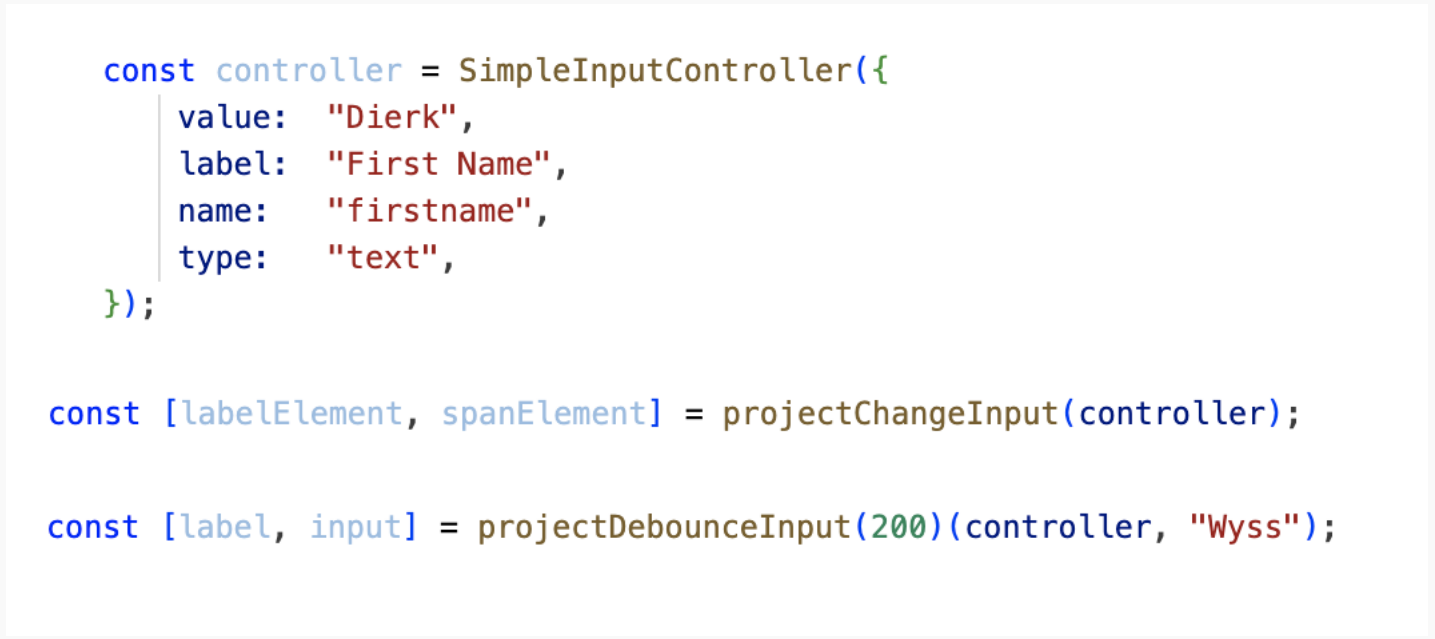
Design

Methodik

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

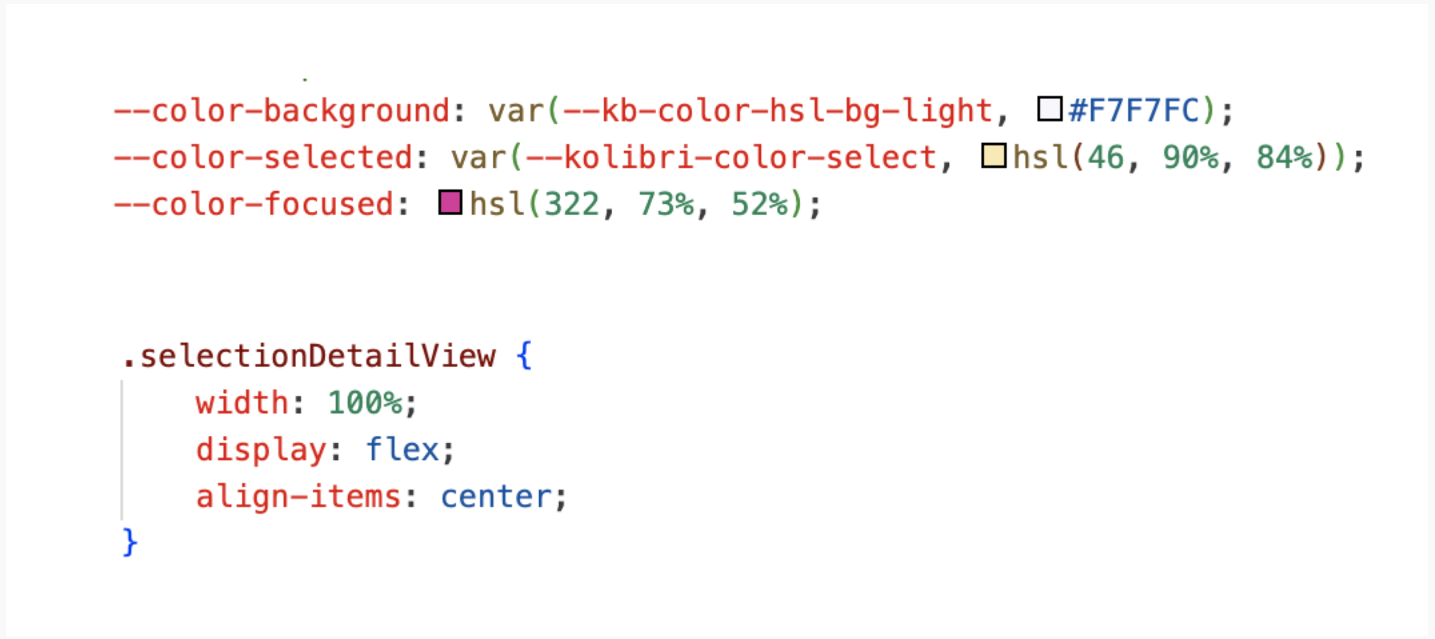


```
   const controller = SimpleInputController({
       value:  "Dierk",
       label:  "First Name",
       name:   "firstname",
       type:   "text",
   });

const [labelElement, spanElement] = projectChangeInput(controller);

const [label, input] = projectDebounceInput(200)(controller, "Wyss");
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

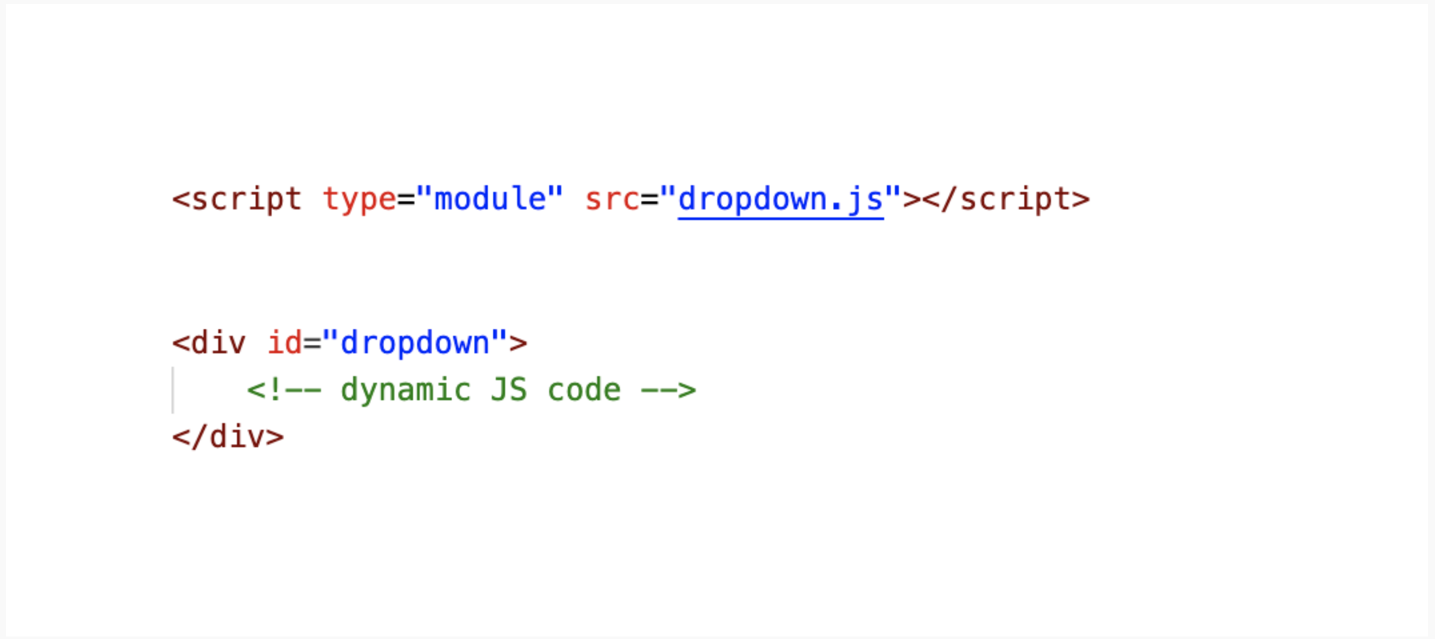


```css
--color-background: var(--kb-color-hsl-bg-light, #F7F7FC);
--color-selected: var(--kolibri-color-select, hsl(46, 90%, 84%));
--color-focused: hsl(322, 73%, 52%);


.selectionDetailView {
    width: 100%;
    display: flex;
    align-items: center;
}
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

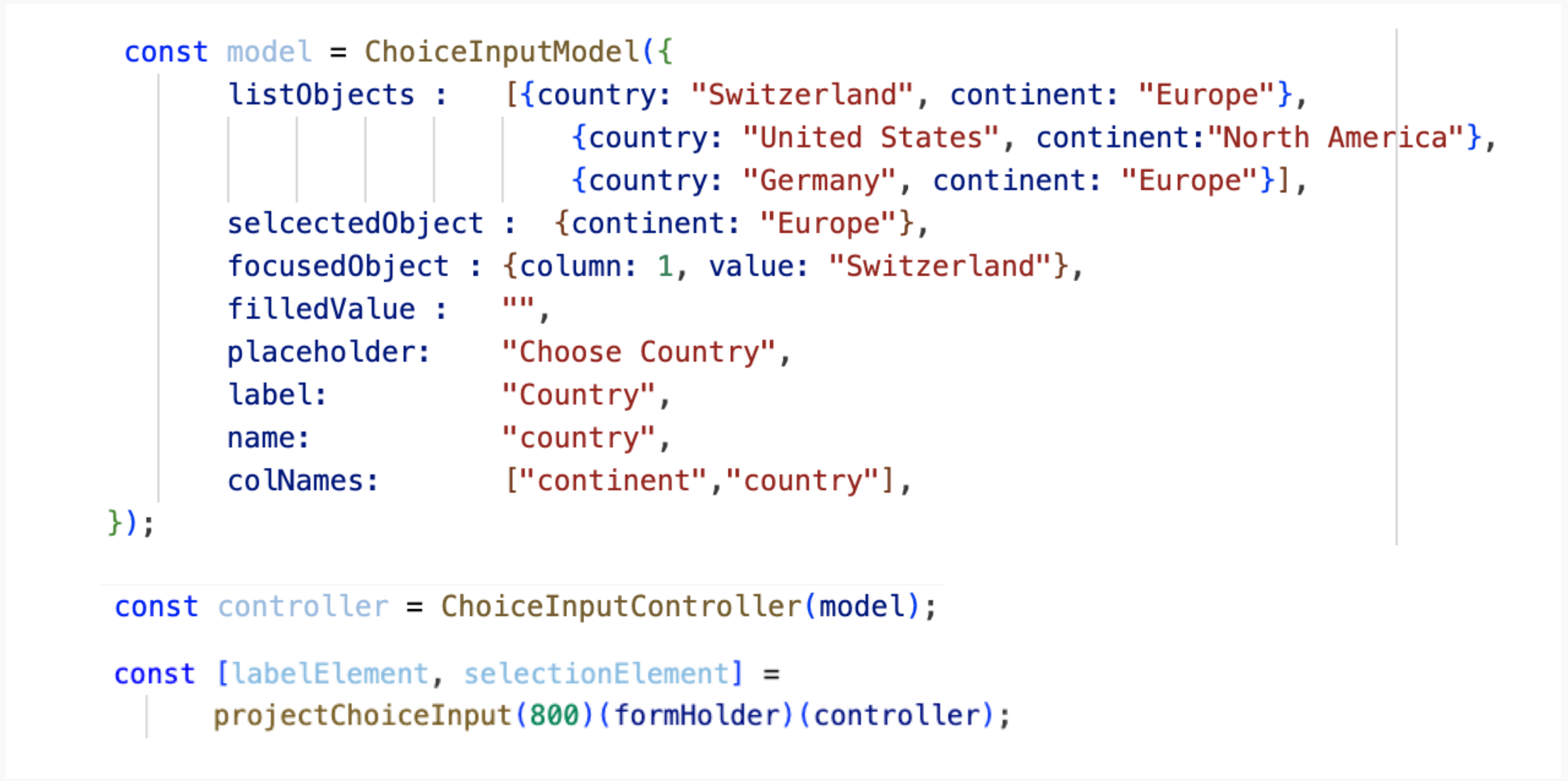


```html
<script type="module" src="dropdown.js"></script>


<div id="dropdown">
    <!-- dynamic JS code -->
</div>
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

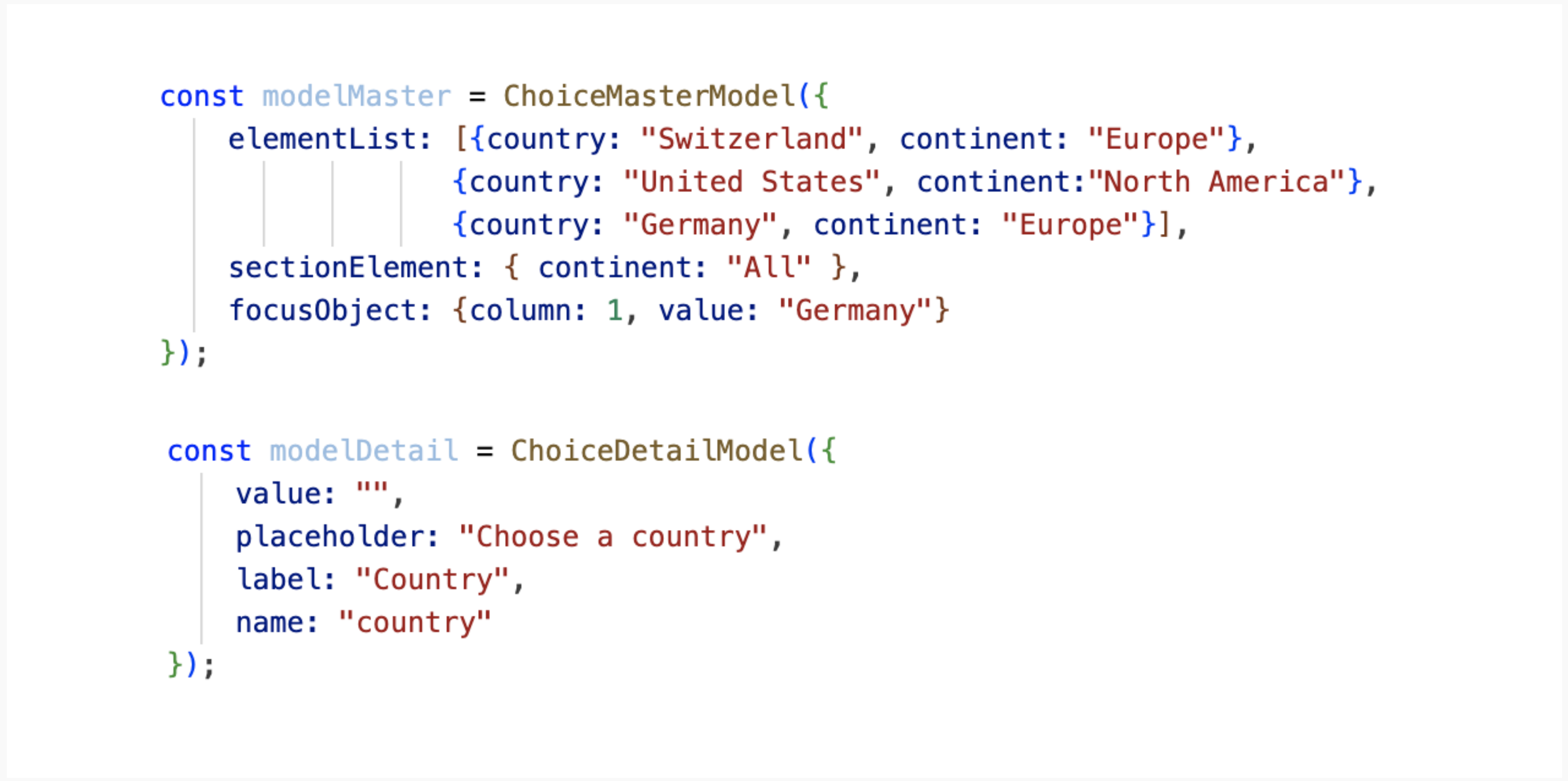


```
const model = ChoiceInputModel({
        listObjects :    [{country: "Switzerland", continent: "Europe"},
                            {country: "United States", continent:"North America"},
                            {country: "Germany", continent: "Europe"}],
        selcectedObject :  {continent: "Europe"},
        focusedObject : {column: 1, value: "Switzerland"},
        filledValue :    "",
        placeholder:     "Choose Country",
        label:           "Country",
        name:            "country",
        colNames:        ["continent","country"],
});

const controller = ChoiceInputController(model);

const [labelElement, selectionElement] =
      projectChoiceInput(800)(formHolder)(controller);
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

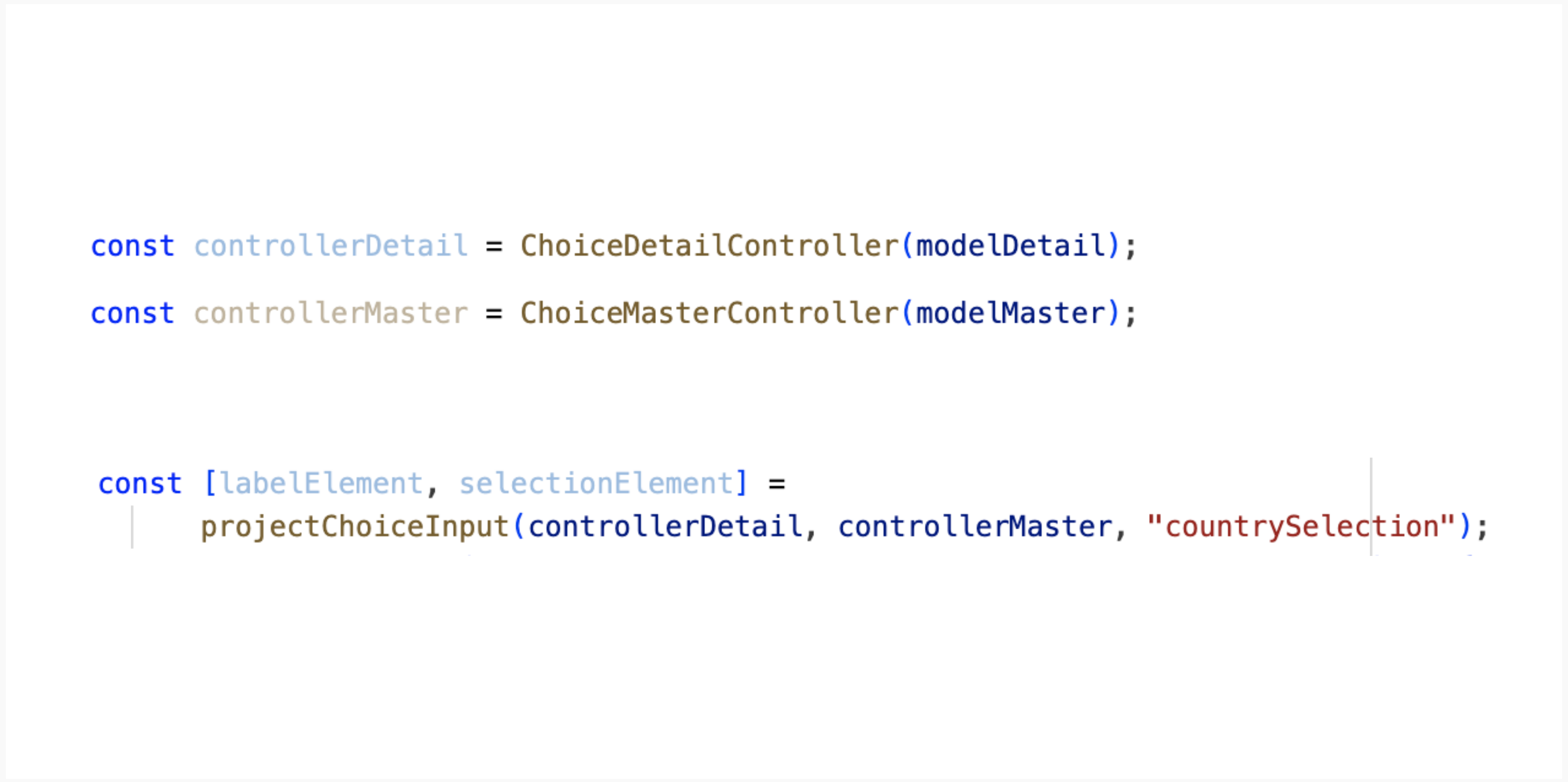


```
const modelMaster = ChoiceMasterModel({
    elementList: [{country: "Switzerland", continent: "Europe"},
                  {country: "United States", continent:"North America"},
                  {country: "Germany", continent: "Europe"}],
    sectionElement: { continent: "All" },
    focusObject: {column: 1, value: "Germany"}
});

const modelDetail = ChoiceDetailModel({
    value: "",
    placeholder: "Choose a country",
    label: "Country",
    name: "country"
});
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
  - CSS
  - HTML
  - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation

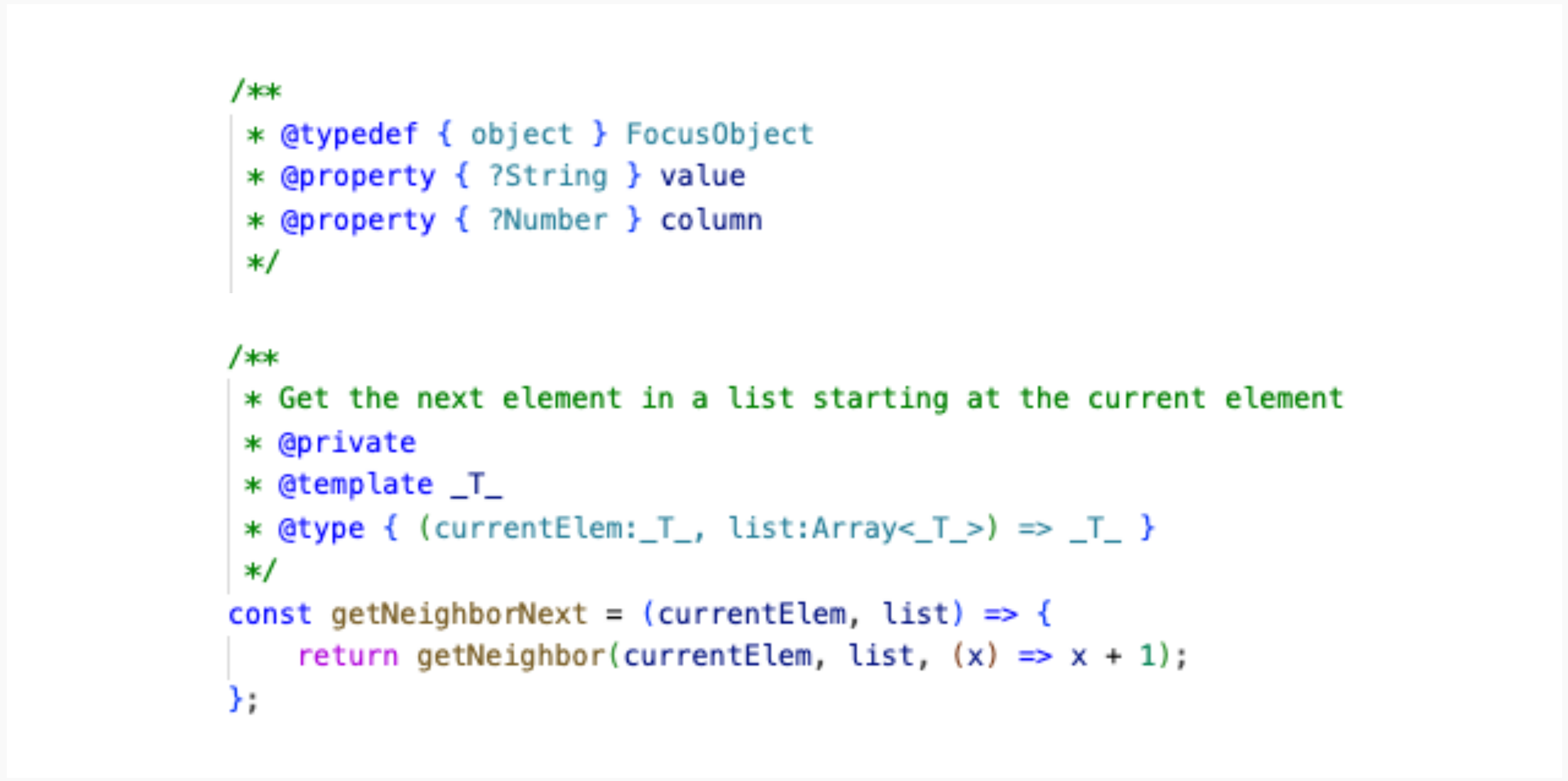


```
const controllerDetail = ChoiceDetailController(modelDetail);

const controllerMaster = ChoiceMasterController(modelMaster);


const [labelElement, selectionElement] =
    projectChoiceInput(controllerDetail, controllerMaster, "countrySelection");
```

Implementation

Methodik

- Analyse Codebasis
- Refactoring
    - CSS
    - HTML
    - JS
- Projector Pattern
- Master Detail View
- Code Dokumentation



```
/**
 * @typedef { object } FocusObject
 * @property { ?String } value
 * @property { ?Number } column
 */

/**
 * Get the next element in a list starting at the current element
 * @private
 * @template _T_
 * @type { (currentElem:_T_, list:Array<_T_>) => _T_ }
 */
const getNeighborNext = (currentElem, list) => {
    return getNeighbor(currentElem, list, (x) => x + 1);
};
```

Herausforderungen

- Gestaltung für umfangreiche Datensätze
- Inkrementelle Suche und Kategorisierung
- Integration und Anpassungsfähigkeit
- Performance-Optimierung
- Accessibility und Usability

Ausblicke

- Erweitertes Testen
- Generalisierung
- Optimierung der Scroll-Logik
- Optimierung der Navigation
- Entwicklung weiterer Varianten
- Ergänzung weiterer Funktionen

Thank you

Diskussion & Fragen