BOOTSTRAP FRAMEWORK

Themen: - Bootstrap Einführung & Setup - Grid System & Breakpoints - Komponenten (Navigation, Cards, Forms) - Utility Classes - Responsive Design mit Bootstrap - Customization & Best Practices

Was ist Bootstrap?

Bootstrap - Das beliebteste CSS Framework

Bootstrap ist ein **Open-Source CSS Framework** für schnelle und responsive Webentwicklung.

Warum Bootstrap? - **Schnelle Entwicklung**: Vorgefertigte Komponenten - **Responsive Design**: Mobile-first Ansatz - **Konsistenz**: Einheitliches Design-System - **Große Community**: Viel Support und Dokumentation - **Anpassbar**: Eigene Themes und Styles

Bootstrap 5 Features: - Kein jQuery mehr benötigt - Flexbox-basiertes Grid System - CSS Custom Properties - Verbesserte API - Kleinere Dateigröße

Setup:

```html
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
```

Bootstrap Grid System

Das 12-Spalten Grid System

Bootstrap verwendet ein **flexibles 12-Spalten Grid System** basierend auf CSS Flexbox.

Grid Struktur:

```html
<div class="container">           <!-- Container -->
    <div class="row">             <!-- Zeile -->
        <div class="col-md-4">    <!-- Spalte -->
            Content 1
        </div>
        <div class="col-md-8">    <!-- Spalte -->
            Content 2
        </div>
    </div>
</div>
```

Container-Typen: - `.container`: Feste max-width für verschiedene Breakpoints - `.container-fluid`: 100% Breite auf allen Breakpoints - `.container-{breakpoint}`: Fluid bis zum spezifischen Breakpoint

Spalten-Klassen: - `.col`: Gleiche Breite - `.col-{number}`: Spezifische Spaltenanzahl (1-12) - `.col-{breakpoint}-{number}`: Responsive Spalten

Responsive Breakpoints

Bootstrap’s Breakpoint System

Bootstrap definiert **5 Breakpoints** für verschiedene Gerätegrößen:

Breakpoint	Class Infix	Dimensions
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
X-Large	`xl`	≥1200px
XX-Large	`xxl`	≥1400px

Responsive Grid Beispiel:

```html
<div class="row">
    <div class="col-12 col-md-6 col-lg-4">
        <!-- Mobile: 12 Spalten, Tablet: 6 Spalten, Desktop: 4 Spalten -->
        Content 1
    </div>
    <div class="col-12 col-md-6 col-lg-8">
        <!-- Mobile: 12 Spalten, Tablet: 6 Spalten, Desktop: 8 Spalten -->
        Content 2
    </div>
</div>
```

Mobile-First Ansatz: Styles gelten ab dem Breakpoint aufwärts!

Komponenten - Navigation

```html
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
        <a class="navbar-brand" href="#">My Website</a>

        <!-- Toggler für Mobile -->
        <button class="navbar-toggler" type="button"
                data-bs-toggle="collapse" data-bs-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>

        <!-- Navigation Links -->
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ms-auto">
                <li class="nav-item">
                    <a class="nav-link active" href="#">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">About</a>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown">
                        Services
                    </a>
                    <ul class="dropdown-menu">
                        <li><a class="dropdown-item" href="#">Web Design</a></li>
                        <li><a class="dropdown-item" href="#">Development</a></li>
                    </ul>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

Komponenten - Cards

Bootstrap Cards - Flexible Content Container

Cards sind **vielseitige Content-Container** für Text, Bilder, Links, etc.

Basic Card:

```
<div class="card" style="width: 18rem;">
    <img src="image.jpg" class="card-img-top" alt="...">
    <div class="card-body">
        <h5 class="card-title">Card title</h5>
        <p class="card-text">Quick example text to build on the card title.</p>
        <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
</div>
```

Komponenten - Cards

Card Grid Layout:

```
<div class="row">
    <div class="col-md-4 mb-4">
        <div class="card h-100">
            <div class="card-body">
                <h5 class="card-title">Service 1</h5>
                <p class="card-text">Description...</p>
            </div>
        </div>
    </div>
    <div class="col-md-4 mb-4">
        <div class="card h-100">
            <div class="card-body">
                <h5 class="card-title">Service 2</h5>
                <p class="card-text">Description...</p>
            </div>
        </div>
    </div>
</div>
```

Card Variants: `.card-header`, `.card-footer`, `.text-center`, `.bg-primary`

Komponenten - Forms

Basic Form:

```
<form>
    <div class="mb-3">
        <label for="email" class="form-label">Email address</label>
        <input type="email" class="form-control" id="email" required>
        <div class="form-text">We'll never share your email.</div>
    </div>

    <div class="mb-3">
        <label for="password" class="form-label">Password</label>
        <input type="password" class="form-control" id="password" required>
    </div>

    <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="remember">
        <label class="form-check-label" for="remember">Remember me</label>
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Input Groups:

```
<div class="input-group mb-3">
    <span class="input-group-text">@</span>
    <input type="text" class="form-control" placeholder="Username">
</div>
```

Form Validation: `.is-valid`, `.is-invalid`, `.valid-feedback`, `.invalid-feedback`

Komponenten - Buttons & Alerts

Buttons - Verschiedene Styles und Größen

```
<!-- Button Variants -->
<button class="btn btn-primary">Primary</button>
<button class="btn btn-secondary">Secondary</button>
<button class="btn btn-success">Success</button>
<button class="btn btn-danger">Danger</button>
<button class="btn btn-outline-primary">Outline</button>

<!-- Button Sizes -->
<button class="btn btn-primary btn-lg">Large</button>
<button class="btn btn-primary">Default</button>
<button class="btn btn-primary btn-sm">Small</button>

<!-- Button Groups -->
<div class="btn-group">
    <button class="btn btn-outline-primary">Left</button>
    <button class="btn btn-outline-primary">Middle</button>
    <button class="btn btn-outline-primary">Right</button>
</div>
```

Alerts - Nachrichten anzeigen:

```
<div class="alert alert-success alert-dismissible fade show">
    <strong>Success!</strong> Your action was completed.
    <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>

<div class="alert alert-warning">
    <strong>Warning!</strong> Please check your input.
</div>
```

Utility Classes

Bootstrap Utilities - Schnelle Styles ohne Custom CSS

Spacing (Margin & Padding):

```html
<!-- Syntax: {property}{sides}-{size} -->
<div class="m-3">Margin auf allen Seiten</div>
<div class="mt-4">Margin top</div>
<div class="px-2">Padding left und right</div>
<div class="mb-5">Margin bottom</div>
```

Display & Flex:

```html
<div class="d-flex justify-content-center align-items-center">
    Zentrierter Content
</div>
<div class="d-none d-md-block">Versteckt auf Mobile</div>
```

Colors & Backgrounds:

```html
<p class="text-primary">Primary Text</p>
<div class="bg-success text-white p-3">Success Background</div>
```

Typography:

```html
<h1 class="display-4">Large Headline</h1>
<p class="lead">Lead paragraph</p>
<small class="text-muted">Muted text</small>
```

Spacing Scale: 0, 1 (0.25rem), 2 (0.5rem), 3 (1rem), 4 (1.5rem), 5 (3rem)

Responsive Design Patterns

Pattern 1: Hero Section

```html
<div class="container-fluid bg-primary text-white py-5">
    <div class="container">
        <div class="row align-items-center">
            <div class="col-lg-6">
                <h1 class="display-4">Welcome</h1>
                <p class="lead">Your success starts here.</p>
                <a href="#" class="btn btn-light btn-lg">Get Started</a>
            </div>
            <div class="col-lg-6">
                <img src="hero.jpg" class="img-fluid">
            </div>
        </div>
    </div>
</div>
```

Responsive Design Patterns

Pattern 2: Feature Cards

```
<div class="container my-5">
    <div class="row g-4">
        <div class="col-md-6 col-lg-4">
            <div class="text-center">
                <i class="bi bi-gear fs-1 text-primary"></i>
                <h4 class="mt-3">Feature 1</h4>
                <p>Description...</p>
            </div>
        </div>
        <!-- Weitere Features... -->
    </div>
</div>
```

JavaScript Komponenten

Modal Dialog:

```html
<!-- Button trigger -->
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">
    Open Modal
</button>

<!-- Modal -->
<div class="modal fade" id="myModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Modal title</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body">
                Modal content goes here...
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">
                    Close
                </button>
                <button type="button" class="btn btn-primary">Save</button>
            </div>
        </div>
    </div>
</div>
```

Weitere JS Komponenten: - Carousel: Bildslider - **Collapse**: Ausklappbare Bereiche
- **Tabs**: Tab-Navigation - **Tooltips**: Hover-Informationen - **Popovers**: Click-Informationen

Customization & Themes

1. CSS Custom Properties überschreiben:

```css
:root {
    --bs-primary: #your-color;
    --bs-font-sans-serif: 'Your Font', sans-serif;
}
```

2. SCSS Variables (bei Sass-Kompilierung):

```scss
// _custom.scss
$primary: #your-brand-color;
$font-family-sans-serif: 'Your Font', sans-serif;

// Dann Bootstrap importieren
@import "bootstrap/scss/bootstrap";
```

3. Eigene Utility Classes:

```css
.text-brand { color: #your-color; }
.bg-gradient-brand {
    background: linear-gradient(45deg, #color1, #color2);
}
```

4. Bootstrap Icons:

```html
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/bootstrap-icons.css" rel="stylesheet">
<i class="bi bi-heart-fill text-danger"></i>
```

Best Practices & Performance

Bootstrap effizient einsetzen

✅ **DO:** - Verwende das **Grid System** für Layout - Nutze **Utility Classes** für kleine Anpassungen - Implementiere **Mobile-First** responsive Design - Verwende **Semantic HTML** mit Bootstrap Classes - Teste auf **verschiedenen Geräten**

❌ **DON’T:** - Überschreibe nicht zu viele Bootstrap-Styles - Vergiss nicht die **Accessibility** - Lade nicht unnötige JS-Komponenten - Verwende keine veralteten Bootstrap-Versionen

Performance-Tipps:

```html
<!-- Nur benötigte Komponenten laden -->
<link href="bootstrap.min.css" rel="stylesheet">

<!-- CSS vor Body-Ende -->
<script src="bootstrap.bundle.min.js"></script>
```

Customization Workflow: 1. Starte mit Standard Bootstrap 2. Definiere eigene Farben/Fonts 3. Erweitere mit Custom CSS 4. Teste responsive Verhalten

Accessibility: Bootstrap ist WCAG-konform, achte auf `alt`-Attribute und ARIA-Labels!

Ressourcen: - Bootstrap Official Documentation - Bootstrap Examples - Bootstrap Icons - Bootstrap Themes