Strukturierte Klassenbenennung in Webflow

Warum Klassennamen wichtig sind

- Basis für sauberen Code und konsistentes Design
- Verhindern Chaos durch automatisch generierte Namen wie „Div 3“
- **Vorteile:**
 - Einfache Änderungen
 - Globale Wiederverwendung
 - Bessere Zusammenarbeit
 - Performance-Optimierung

Namenskonventionen aus der Informatik

- **Snake Case:** primary_button – gut lesbar, mit Unterstrichen
- **Camel Case:** primaryButton – kompakt, aber weniger lesbar bei langen Namen
- **Kebab Case:** primary-button – Standard in CSS, sehr verbreitet

Frameworks für Webflow-Klassen

- **BEM (Block-Element-Modifier)**
 - Block: eigenständige Komponente (button)
 - Element: Bestandteil (button__text)
 - Modifier: Variation (button--large)
 - Vorteile: klare Struktur, weniger Konflikte
 - Nachteil: längere Klassennamen
- **Client-First (von Finsweet)**
 - Einheitlicher Aufbau von Sektionen und Seiten
 - Globale Klassen für Konsistenz (container-large)
 - Deskriptive Namen für eigene Klassen (faq-list_image-wrapper)
 - Max. 1-2 Kombo-Klassen, kein Deep-Stacking
 - Fokus auch auf Responsiveness & Barrierefreiheit
- **Lumos**
 - Breakpointless und Barrierefreiheit im Fokus
 - Aufbau der Seite mit Komponenten zur einfachen Verwaltung

Mein angepasstes System

- Seitenspezifische Präfixe: z. B. h_hero_collection-list
- Mix aus Snake & Kebab Case: Unterstriche = neue Komponente, Bindestriche = Element
- Kombo-Klassen: Variationen mit is--blue
- Ziel: Vorteile von Client-First behalten, aber flexibler anpassen

Fazit

- Strukturierte Klassennamen = Basis für saubere, skalierbare Webflow-Projekte
- BEM und Client-First sind bewährte Frameworks
- Systeme dürfen (und sollen) an den eigenen Workflow angepasst werden
- Ergebnis: bessere Zusammenarbeit, Performance und Zukunftssicherheit