

# Programm-Beschreibung: Ein flexibles Template für Anwendungen

Das vorgestellte Programm dient als **Template**, das eine klare, modulare Struktur für Webanwendungen bereitstellt. Es wurde speziell für Anwendungen entwickelt, die **Seitenleisten-Navigation**, **dynamische Inhalte** und **Fenster-Management** erfordern.

Durch die **Harbourino-Syntax** ist das Template einfach erweiterbar und anpassbar, sodass es sich für zahlreiche Anwendungsfälle eignet.

---

## Hauptbestandteile des Templates

### 1. HTML-Struktur

- **HEAD** :
    - Enthält die Meta-Informationen, Style-Sheets (z. B. Bootstrap, FontAwesome) und Skripte.
    - Kann erweitert werden, um weitere Ressourcen einzubinden, z. B. benutzerdefinierte Schriftarten oder zusätzliche JavaScript-Bibliotheken.
  - **NAVBAR** :
    - Fixierte Navigationsleiste oben.
    - Wird verwendet, um globale Navigationselemente wie Home, Einstellungen oder Logout bereitzustellen.
    - Kann durch Dropdown-Menüs oder zusätzliche Links erweitert werden.
  - **SIDEBAR** :
    - Dynamische Seitenleiste mit Navigationselementen.
    - Unterstützt Submenüs und verschiedene Darstellungsmodi (z. B. Vollansicht, Icon-Ansicht, ausgeblendet).
    - Die Sidebar kann erweitert werden, indem neue Menüeinträge oder Submenüs hinzugefügt werden.
  - **CONTENT** :
    - Der Hauptinhalt der Seite.
    - Dynamisch, basierend auf Benutzerinteraktionen, z. B. Klicks auf die Sidebar.
- 

### 2. JavaScript-Funktionen

- **Globale Variablen (GL\_)**:

- Enthält Konfigurationen und Zustände, z. B. die Liste offener Fenster oder Programmcounter.
  - **Hauptfunktionen ( FN\_ ):**
    - Zentrale Logik für das Programm-Management:
      - FN\_OPENPROGRAM : Öffnet Fenster für Programme.
      - FN\_CLOSEPROGRAM : Schließt Fenster.
      - FN\_SHOWOPTIONS : Zeigt ein Menü für spezifische Aktionen.
      - FN\_RENDEROPENPROGRAMS : Aktualisiert die Darstellung der offenen Fenster.
  - **Event-Handler ( CL\_ ):**
    - Reagieren auf Benutzerinteraktionen:
      - CL\_SIDEBAR : Handhabt Klicks auf Sidebar-Links.
      - EV\_SIDEBAR\_TOGGLE : Ermöglicht das Ein-/Ausklappen der Sidebar.
  - **Hilfsfunktionen ( FN\_ / HF\_ ):**
    - Unterstützen die Hauptfunktionen:
      - FN\_SHOWCONTENTBYID : Zeigt Inhalte basierend auf einer ID.
      - FN\_GETVISIBLESIDEBARLINKS : Gibt nur sichtbare Links in der Sidebar zurück.
      - FN\_UPDATEACTIVEITEM : Markiert ein aktives Sidebar-Item.
- 

## Einsatzmöglichkeiten

Das Template eignet sich für viele Arten von Anwendungen:

### 1. Administrationsoberflächen

- **Beispiel:**
  - Eine Verwaltungsoberfläche für ein Hotel, mit Modulen wie „Zimmerplan“, „Reservierungen“ und „Abrechnung“.
- **Anpassung:**
  - Neue Sidebar-Einträge für Module hinzufügen.
  - Inhalte (z. B. Tabellen, Formulare) in den Content-Bereich laden.

### 2. Dynamische Dashboards

- **Beispiel:**
  - Ein Dashboard, das Statistiken und KPIs anzeigt.
- **Anpassung:**
  - Inhalte im CONTENT -Bereich dynamisch durch API-Aufrufe oder Datenbanken laden.
  - Widgets und Diagramme in die Sidebar integrieren.

### 3. Single-Page-Anwendungen (SPA)

- **Beispiel:**

- Eine Single-Page-Anwendung, die ohne vollständige Seitenladevorgänge auskommt.
  - **Anpassung:**
    - Inhalte per `fetch` oder AJAX laden und in den `CONTENT`-Bereich einfügen.
- 

## Erweiterungsmöglichkeiten

### 1. Neue Funktionen

- Füge neue JavaScript-Funktionen mit der `FN_`-Syntax hinzu:

```
-> FN_NEWFUNCTION /* Implementiert eine neue Logik */
function FN_NEWFUNCTION(param) {
    console.log('Neue Funktion wurde aufgerufen:', param);
}
```

- 

### 2. Zusätzliche Navigation

- Erweiterung der Sidebar:

```
-> BL_SIDEBAR
<li class="nav-item">
    <a href="#" class="sidebar-link" data-target="new-section">
        <i class="fas fa-new-icon"></i>
        <span class="sidebar-text">Neue Sektion</span>
    </a>
</li>
```

- 

### 3. Fenster-Management

- Neue Programme oder Module können durch `FN_OPENPROGRAM` hinzugefügt werden:

```
FN_OPENPROGRAM('Neues Modul', 'neuesModul.html');
```

- 

### 4. Dynamische Inhalte

- Inhalte können aus APIs oder Datenbanken geladen werden:

```
fetch('api/data')
    .then(response => response.json())
    .then(data => {
        FN_SHOWCONTENTBYID('content-area');
        console.log('API-Daten:', data);
    });
```

- 

---

## Vorteile des Templates

### 1. Modularität:

- Der Code ist in logische Blöcke (`BL_`), Funktionen (`FN_`) und Event-Handler (`CL_`) gegliedert.

### 2. Wiederverwendbarkeit:

- Funktionen und Blöcke können in anderen Projekten wiederverwendet werden.

### 3. Klarheit durch Harbourino-Syntax:

- Die klare Syntax und Benennung erleichtert das Verständnis und die Zusammenarbeit.

### 4. Flexibilität:

- Unterstützt sowohl statische als auch dynamische Inhalte.
- 

## Schritte zur Anpassung

### 1. Ergänzen von Inhalten:

- Füge neue Abschnitte zur Sidebar oder zum Content hinzu.

### 2. Erweitern von Funktionen:

- Implementiere neue Funktionen und Hilfslogik, um zusätzliche Features zu unterstützen.

### 3. Testen:

- Teste das Programm in verschiedenen Browsern und auf unterschiedlichen Geräten, um sicherzustellen, dass alle Zustände korrekt funktionieren.
- 

Mit diesem Template erhältst du eine solide Grundlage, die sich leicht an deine spezifischen Anforderungen anpassen lässt. Es bietet nicht nur **Klarheit und Struktur**, sondern auch die Flexibilität, komplexe Anwendungen effizient zu entwickeln.