Budget Calculator – Teil 2

Lernziele

- DOM Manipulation (Elemente erstellen, Attribute setzten, an bestimmter Stelle einfügen, …)
- CSS
  - Styling mit Klassen
  - Klassen über JS setzen (Conditional Styling)
  - CSS Layouting üben
- Zusammenhang HTML, CSS, JS besser verstehen

Recap – Teil 1

1. Usereingaben/Formulare (HTML) ✔️

1. Eingabefeld für Budget
2. Eingabeformular für Ausgaben (Beschreibung + Betrag)

2. Speichern der Eingaben ✔️

1. Referenz zu Einagbefelder und Buttons in JS als Variable speichern
2. EventListener für Speichern Buttons
3. Budget in globaler Variable speichern
4. Ausgaben (Text + Betrag) in Array speichern

3. Berechnen der Werte ✔️

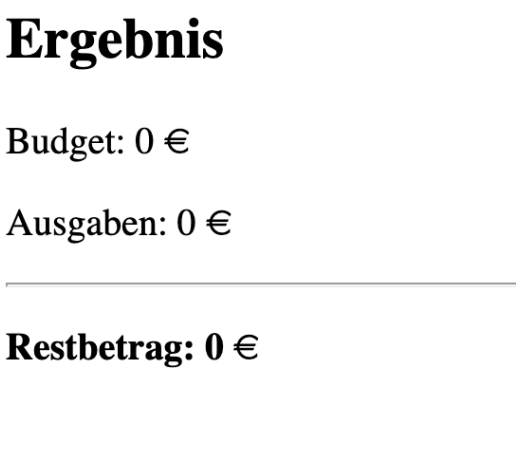
1. Summe der Ausgaben berechnen
2. Differenz von Budget und Summe Ausgaben berechnen
3. Berechnung nach Speichern von Eingaben ausführen
4. Resultat in Konsole ausgeben

a. z.B. 2500 (Budget) - 800 (Ausgaben) = 1700 (Restbetrag)

Aufgaben – Teil 2

1. Ausgabe der Ergebnisse

Bei jeder Eingabe von neuen Beträgen (Aktualisierung Budget, Neue Ausgaben) soll der Ergebnisbereich unterhalb der Formulare aktualisiert werden.



2. Formulare zurücksetzen

Nach dem **erfolgreichen Speichern** eines Formulars soll das jeweilige Formular geleert und zurückgesetzt werden.

3. Eingaben validieren

Die Eingaben für Budgen und Ausgaben sollen folgenderweise validiert werden

- Bei Zahleingaben
  - Keine leeren Eingaben
  - Keine negativen Zahlen
- Bei Texteingaben
  - Keine leeren Eingaben

- Bei Ausgaben → Ausgabe + Betrag müssen valide sein

Bei invalider Eingabe soll ein **Alert** mit einer Fehlermeldung ausgegeben werden, um dem User den Grund für die invalide Eingabe mitzuteilen. In diesem Fall soll das Formular **nicht zurückgesetzt** werden und die eingegebenen Werte sollen nicht weiterverarbeitet werden.

4. Conditional Styling

- Bei **invaliden Eingaben** sollen die Input Felder (nach dem Speichern) einen **roten Rahmen** haben. Erst bei einer weiteren validen Eingabe soll der rote Rahmen wieder entfernt werden
- Der Restbetrag soll je nach Betrag rot oder grün gestylt werden
  - Betrag größer gleich 0 → grün
  - Betrag kleiner 0 → rot

5. Styling der App

- Als Vorlage für das Layout/Styling der App soll dieser Link dienen: https://romeojeremiah.github.io/javascript-oop-budget-project/
- **WICHTIG:** Ziel ist es nicht, die App eins zu eins perfekt nachzustylen. Es soll rein als **Vorlage** dienen. Details wie z.B. genaue Abstände, Schriftgrößen, Icons, responsives Verhalten etc. können vernachlässigt werden.