



Vorlesung GUI

Übungsblatt 03: Events

Theoriefragen

1. Befülle den Callstack. Überlege dir zu jedem Zeitpunkt im Programmablauf, was auf dem Stack liegt.

```
function greet() {  
  sayHello()  
  sayBye()  
}  
  
function sayHello() {  
  console.log("Hallo!")  
}  
  
function sayBye() {  
  console.log("Tschüss!")  
}  
  
greet()
```

2. Was passiert mit deiner Applikation bei langlaufenden Tasks?

3. Welche Queues gibt es in der JavaScript Runtime und wozu braucht man sie?
 4. Welche Aufgabe hat die JavaScript Event Loop?
 5. Erkläre Event Capturing und Bubbling.
-

Praxisteil

Ziel:

Erstelle eine **Eingabeseite** für neue Blogeinträge, bei der bestimmte Benutzereingaben verhindert werden und aktiv auf Benutzerinteraktionen reagiert wird.

Neuen Blogbeitrag erstellen

Titel:

Test

Zeichen ! ist im Titel nicht erlaubt

Inhalt:

Hier kannst du den Blogtext schreiben...

Speichern

Blogbeitrag erfolgreich gespeichert!

Anforderungen

1. Titel-Feld mit Zeichenvalidierung

Es gibt ein Textfeld für den Titel des Blogbeitrags.

Wenn die Nutzer:in das Zeichen ! oder @ (Option: alle Sonderzeichen) eingeben möchte, wird die Eingabe verhindert.

Zusätzlich erscheint unter dem Feld eine Fehlermeldung wie z. B. "Zeichen ! ist im Titel nicht erlaubt"

💡 Tipp: Du kannst auf Elemente im DOM über die ID sehr gut zugreifen und dann z.B. den `textContent` eines `<p>` Elements verändern:

```
const errorEl = document.getElementById('title-error')
```

2. Textfeld mit Fokus-Info

Beim Klicken auf das Inhaltsfeld (Textarea) erscheint ein Hinweis darunter:
"Hier kannst du den Blogtext schreiben..."
Wenn das Feld wieder verlassen wird, verschwindet der Hinweis.

💡 Tipp: Du kannst auch den Style eines Elements im Code verändern und so Elemente ausblenden:

```
document.getElementById('info-box').style.display = 'none'
```

3. Speichern-Button mit Bestätigung

Beim Klick auf den Button wird zuerst "Speichern..." angezeigt.
Nach einer Sekunde erscheint die Nachricht: "Blogeintrag erfolgreich gespeichert!"

💡 Tipp: Wenn du Aktionen verzögern möchtest, hilft dir die `setTimeout` Methode:

```
setTimeout(() => {  
    msg.textContent = 'Blogeintrag erfolgreich gespeichert!'  
}, 1000)
```

Zusatzaufgabe: Responsive Design mit Media Queries

Du erweiterst die Blog-Eingabeseite um einen **Kategorien-Bereich**. Dort können Nutzer:innen Kategorien für ihren Blogeintrag auswählen:

- Der ganze Bereich hat einen übergeordneten Klick-Handler.
- Die einzelnen Kategorie-Buttons haben eigene Aktionen.

Teste die unterschiedlichen Event-Phasen und Optionen, um Events zu stoppen (sowohl im Parent als auch im Child), aus. Du kannst dir mit `console.log` Ausgaben behilflich sein, um deine Überlegungen zu überprüfen.