



**Hochschule  
Albstadt-Sigmaringen**  
Albstadt-Sigmaringen University

# Cross Site Scripting (XSS)

**Fakultät Informatik - Studiengang IT Security B.Sc.**

**Modul 13000 - Einführung offensive Security-Methoden**

Betreuer: Prof. Dr. Christian Henrich, Dr. Demir, Nurullah

Vortrag von Fabian Hans Flad - 10.01.2025

# Übersicht



1. **Definition:** Was ist Cross Site Scripting (XSS)
2. **Grundlagen:** Aufbau von Internetseiten
3. **Praktische Umsetzung**
4. **Gegenmaßnahmen**
5. **Fazit**
6. **Quellen**

# 1. Definition: Was ist Cross Site Scripting (XSS)




*„Bei einem Cross-Site-Scripting-Angriff (...), wird **Schadcode** in ansonsten vertrauenswürdige Webseiten eingeschleust. Der Browser des Opfers erkennt nicht, dass die Skripte nicht vertrauenswürdig sind, und führt sie bedenkenlos aus.“*  
(vgl. AO Kaspersky Lab 2024)

*"Dieser bösartige Code ist dann in der Lage, den **Inhalt der Webseite zu verfälschen** (z. B. eine Falschmeldung auf der Webseite einer Zeitung zu platzieren) oder **vertrauliche Daten auszulesen** (z.B. das Passwort des Opfers)." (vgl. Schwenk 2020)*

# Kategorien

Stored XSS 	Reflected XSS 
Schadcode wird auf System der Zielwebsite <b>dauerhaft</b> gespeichert.	Schadcode wird auf System der Zielwebsite <b>nicht</b> gespeichert.
Schadcode wird <b>bei allen Besuchern</b> der Website ausgeführt.	Gezielter Angriff <b>einzelner User</b> über <b>manipulierte URL</b> .

## 2. Grundlagen: Aufbau von Internetseiten

Komponente	Funktion	Symbolisch
HTML	Grundgerüst: Ordnet Inhalte Kategorien zu.	 Skelett
CSS	Styling: Farbe, Schriftart und Formatierung.	 Haut
JavaScript	Interaktive Funktionen: Nachladen von Inhalten, Ausführung von Funktionen	 Gehirn

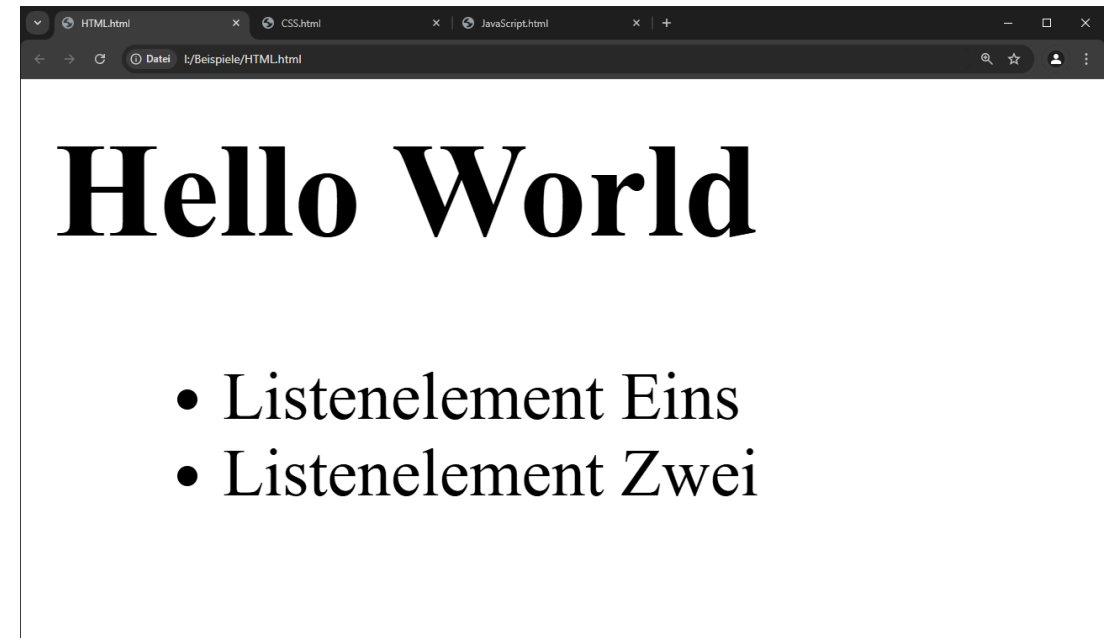
# HTML

HTML-Code besteht aus Tags `</>`

Tags geben Inhalten Kontext

Tags besitzen Attribute `id`, `src`

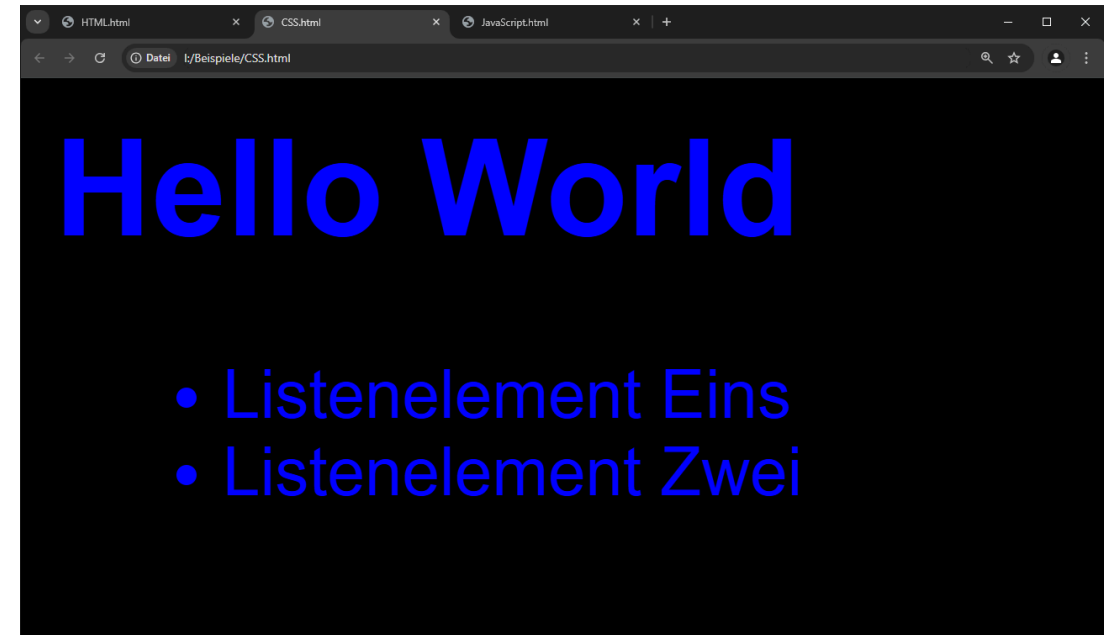
```
<html>
<body>
  <h1 id="Titel">Hello World</h1>
  <ul>
    <li>Listenelement Eins</li>
    <li>Listenelement Zwei</li>
  </ul>
</body>
</html>
```



# CSS

Ordnet HTML-Tags über geschweifte Klammern { } Styling-Eigenschaften zu

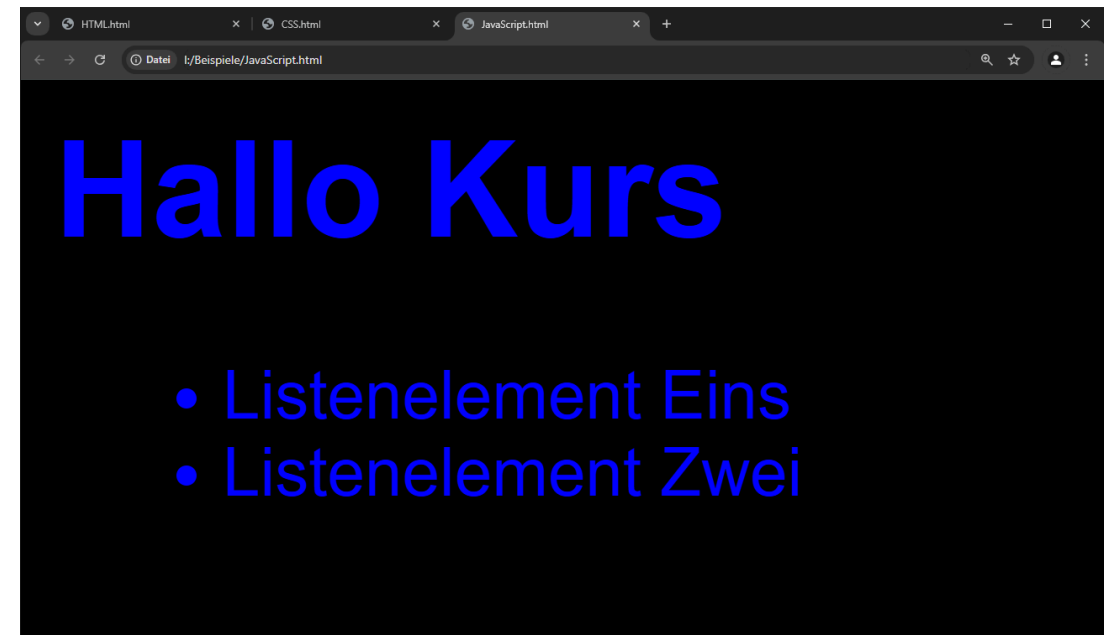
```
body {  
    /* Schriftfarbe */  
    color: blue;  
    /* Schriftart */  
    font-family: sans-serif;  
    /* Hintergrund Farbe */  
    background: black;  
}
```



# JavaScript

Ermöglicht interaktive und dynamische Funktionen

```
// Zugriff auf Tags mit ID "Titel"  
let Titel = document.getElementById("Titel");  
  
// Ersetzt Inhalt des HTML-Tags  
Titel.innerHTML = "Hallo Kurs"
```





# Einbindung von JavaScript

1. Code direkt zwischen `<script>` -Tags
2. Einbindung über `src` -Attribut

```
<html>
  <head>
    <script>alert("Hello World")</script>
    <script src="Quelle.js"></script>
  </head>
</html>
```

## 3. Praktische Umsetzung eines XSS Angriffs

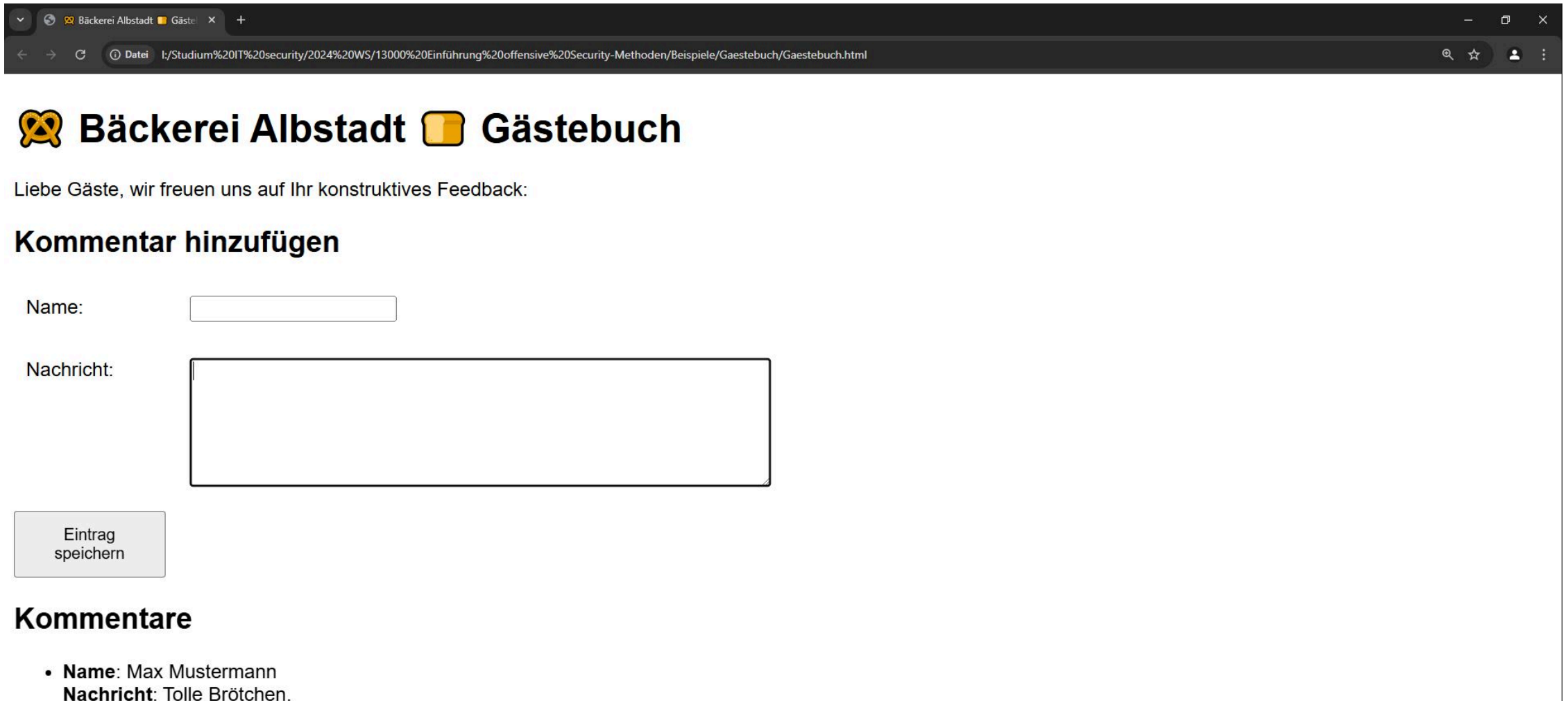
Wie gelangt schädlicher Code auf fremde Internetseiten?

- Such- und Eingabefelder
- URL-Parameter

Gefahr droht überall wo User-Input wieder ausgegeben wird:

- Suchergebnis
- Kommentarspalten / Gästebuch

# Beispiel: Stored XSS Angriff - Gästebuch



**Bäckerei Albstadt Gästebuch**

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

### Kommentar hinzufügen

Name:

Nachricht:

### Kommentare

- **Name:** Max Mustermann  
**Nachricht:** Tolle Brötchen.



## Bäckerei Albstadt Gästebuch

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

### Kommentar hinzufügen

Name:

Nachricht:

Eintrag  
speichern

### Kommentare

- **Name:** Max Mustermann  
**Nachricht:** Tolle Brötchen.



## Bäckerei Albstadt Gästebuch

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

### Kommentar hinzufügen

Name:

Nachricht:

Eintrag  
speichern

### Kommentare

- **Name:** Max Mustermann  
**Nachricht:** Tolle Brötchen.
- **Name:** Test  
**Nachricht:** Test

Bäckerei Albstadt

Gästebuch

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

**Kommentar hinzufügen**

Name:

Nachricht:

Eintrag speichern

**Kommentare**

- li#Kommentar\_1 924 x 36
- Name: Test  
Nachricht: Test

Elemente
 Konsole
 Quellcode
 Netzwerk
 Leistung

```

<!DOCTYPE html>
<html lang="de">
  <head>
  </head>
  <body>
    <header id="ueberschrift">
    </header>
    <main>
      <section id="guestbook-form">
      </section>
      <section>
        <h2>Kommentare</h2>
        <ul id="guestbook-entries">
          <li id="Kommentar_0">
          </li>
          <li id="Kommentar_1"> == $0
            ::marker
            <b>Name: </b>
            "Test"
            <br>
            <b>Nachricht: </b>
            "Test"
            <br>
          </li>
        </ul>
      </section>
    </main>
  </body>
</html>

```

html
 body
 main
 section
 ul#guestbook-entries
 li#Kommentar\_1

Style
 Berechnet
 Layout
 Ereignis-Listener
 DOM-Umschaltpunkte
 Eigenschaften

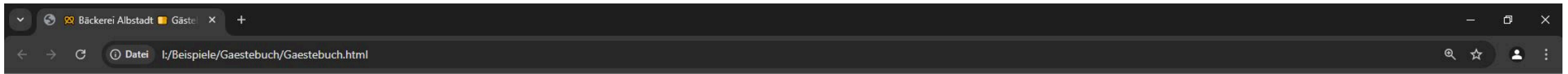
Filtern
 :hov .cls

```

element.style {
}

li {
  display: list-item;
  text-align: -webkit-match-parent;
}

```



## Bäckerei Albstadt Gästebuch

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

### Kommentar hinzufügen

Name:

Nachricht: 

```
<script>
let MaxMustermann = document.getElementById("Kommentar_0");
MaxMustermann.innerHTML = "<b>Name:</b> Max Mustermann <br>
<b>Nachricht:</b> Schmeckt überhaupt nicht!";
</script>
Mir schmeckt's auch nicht!
```

Eintrag  
speichern

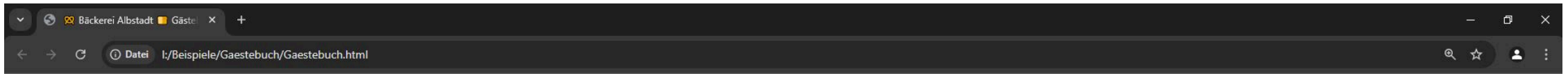
### Kommentare

- **Name:** Max Mustermann  
**Nachricht:** Tolle Brötchen.
- **Name:** Test  
**Nachricht:** Test

```
<script>
let MaxMustermann = document.getElementById("Kommentar_0");
MaxMustermann.innerHTML = "<b>Name:</b> Max Mustermann <br><b>Nachricht:</b> Schmeckt überhaupt nicht!";
</script>
Mir schmeckt's auch nicht!
```

- `document.getElementById()` Zugriff auf Tag mit `id` *"Kommentar\_0"*
- Zugriff auf Tag über Variable `MaxMustermann`
- `MaxMustermann.innerHTML` Zieltag wird neuer Inhalt zugeordnet
- Alle Teile außerhalb des `<script>` -Tags werden nicht als Skript interpretiert





## Bäckerei Albstadt Gästebuch

Liebe Gäste, wir freuen uns auf Ihr konstruktives Feedback:

### Kommentar hinzufügen

Name:

Nachricht:



Eintrag  
speichern

### Kommentare

- **Name:** Max Mustermann  
**Nachricht:** Schmeckt überhaupt nicht!
- **Name:** Test  
**Nachricht:** Test
- **Name:** Mallory  
**Nachricht:** Mir schmeckt's auch nicht!



## 4. Gegenmaßnahmen

Web Entwickler 	Endnutzer 
Wie lassen sich Websites <b>sicher gestalten</b> ?	Wie lassen sich Websites <b>sicher nutzen</b> ?

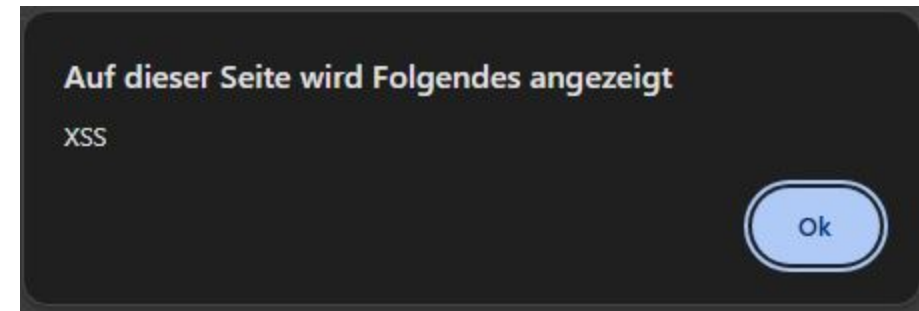
## 4.1 Web Entwickler

### Security by Design

Sicherheitsaspekte bereits im Entwicklungsprozess berücksichtigen

```
const userInput = "<script>alert('XSS')</script>"
```

```
// Input wird als HTML gelesen  
let element = document.getElementById("id")  
element.innerHTML = userInput
```



```
// Input wird als Text gelesen  
let element = document.getElementById("id")  
element.textContent = userInput
```

**<script>alert('XSS')</script>**

## User Input Filtern

	● Black List Verfahren	● White List Verfahren
<b>Grundsatz</b>	Alle Zeichen sind erlaubt	Alle Zeichen sind Verboten
<b>Ausnahme</b>	<code>&lt;script&gt;</code> , <code>&lt;</code> , <code>&gt;</code> , <code>&lt;\</code> , ...	<code>a</code> , <code>b</code> , <code>c</code> , <code>d</code> , <code>e</code> , <code>1</code> , <code>2</code> , ...
<b>Bewertung</b>	Substitution droht: Sicherheit ist fragil.	Effiziente Reduktion der Angriffsfläche

## 4.2 Endnutzer

### JavaScript abschalten ✖

Funktion, die fast alle Browser anbieten.

Website möglicherweise nicht mehr nutzbar.

### Vorsicht bei Links 🔗

Moderne Internet-Browser und E-Mail Programme können dabei helfen XSS Angriffe frühzeitig zu erkennen.

## 5. Fazit

**Auch vermeintlich vertrauenswürdige Websites können mit Schadcode injeziert sein.**

## 6. Quellen

- **Ackermann, Philip** - Rheinwerk Verlag - JavaScript - Das umfassende Handbuch - 3 Auflage 2021 S. 416 f.
- **Kofler Michael et al** - Rheinwerk Verlag - Hacking & Security - Das umfassende Handbuch 3 Auflage 2023 - S. 850 ff.
- **Schwenk Jörg** - Springer - Sicherheit und Kryptographie im Internet 5. Auflage S. 449 ff.
- **Bundesamt für Sicherheit in der Informationstechnik (BSI) Goldene Regeln Baustein B 5.21 Webanwendungen**
- **Kaspersky Lab** - Definitions - What is a cross site scripting attack



## Link zu den Beispielen und den Unterlagen

