

## Aufgabe 1 (Wiederholung Formulare)

Wie lautet der Query String, wenn Sie Ihren Namen, Ihre Größe etc. in Ihr BMI-Formular von Übung 1 (Aufgabe 4) eingeben und das Formular durch Anklicken des Submit-Buttons absenden? Das Name-Attribut des Namensfeldes soll dabei noch den Wert "uname" bekommen!

(Das Formular werden Sie übrigens in dieser Form in einer der nächsten Aufgaben weiter ergänzen.)

**BMI-Rechner**

Angaben zur Berechnung

Gewicht

Größe

Geschlecht

☐ Männlich ☒ Weiblich

Name

Email

Anmerkungen

Ergänzen Sie Ihr Formular außerdem noch um einen Bereich für Anmerkungen (wie oben gezeigt), mit Namen "anmerk".

Spiele Sie nun den Bösewicht und fügen Sie HTML und JavaScript-Code in die Textarea ein (z.B. für ein Pop-Up) und senden Sie Ihr Formular ab. Versuchen Sie insbesondere auch programmatisch die URL der Seite umzusetzen. Testen Sie das Ergebnis – mit geöffneter Entwicklerkonsole – auf Firefox, Chrome und ggfs. Internet Explorer bzw. Safari!

Welches Problem besteht hier vor allem beim Firefox? Recherchieren Sie, was die in der Chrome-Konsole genannte Abkürzung bedeutet.

## Aufgabe 2 (Grundlagen JS und Funktionen)

1. Geben Sie mit Hilfe einer ‚switch‘-Anweisung, die den Wert von `getDay()` des Date-Objekts<sup>1</sup> als Bedingung nimmt, den aktuellen Wochentag als String aus, wobei die Ausgabe mit Hilfe einer ‚alert‘-Box geschehen soll.
2. Implementieren Sie eine JavaScript-Funktion, die zwei Zahlen, die als Parameter übergeben werden, addiert und den Ergebniswert zurückliefert. Implementieren Sie weiterhin eine Funktion, der zwei Strings als Argumente übergeben werden und die als Ergebnis die Konkatination beider Strings zurückgibt.

<sup>1</sup> Vgl. z.B. [http://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](http://www.w3schools.com/jsref/jsref_obj_date.asp)

3. Schreiben Sie in JavaScript eine rekursive Funktion namens *summe(n)*, welche für eine natürliche Zahl  $n \geq 0$  die Summe aller Zahlen von 0 bis einschließlich  $n$  berechnet. Berücksichtigen Sie auch eventuelle Problemfälle!
4. Implementieren Sie jeweils eine rekursive sowie eine iterative JS-Funktion zur Berechnung der Fakultät einer natürlichen Zahl  $n$  (zur Erinnerung:  $0! = 1$ ).
5. Schreiben Sie eine Funktion *pattern(numLines)*, die das nachfolgende Muster anzeigt. Die Anzahl der Zeilen, aus denen das Muster besteht, soll variabel sein und der Funktion mit Hilfe des Parameters *numLines* übergeben werden. Verwenden Sie für die Ausgabe das `<pre>`-Tag, damit das Muster korrekt (mit Leerzeichen) dargestellt wird.

```
  +
+++
+++++
```