

### Aufgabe 1: (max. 2 + 4 + 1 Punkte)

Gegeben sei ein HTML5-Dokument, das folgenden Abschnitt enthält:

```
<form action="form_response.php" method="get">
  <fieldset><legend>We want to know you!</legend>
  username:
  <input type="text" name="username" size="25" maxlength="30" required>
  password:
  <input type="text" name="password" size="25" maxlength="30" required>
  <input type="submit" value="send">
</fieldset>
</form>
```

a) Wie sieht der Request-String (den der Browser als URL anzeigt) aus, wenn das Formular mit *Mustermann* und *pw123* ausgefüllt und den Send-Button gedrückt wird?

b) Generieren Sie mit PHP eine HTML5-Seite, die auf einen Request - wie in a) beschrieben - mit folgender Meldung antwortet:

Herr / Frau Mustermann, Ihr Passwort pw123 ist nicht mehr gültig.

c) Was würde sich ändern, wenn die Request Method von *get* auf *post* geändert würde?

### Aufgabe 2:

(max. 1 + 1 Punkte)

a) Wie lassen sich die folgenden jQuery-Anweisungen verkürzt formulieren?

b) Begründen Sie, warum Ihre Lösung richtig ist.

```
<script>
  $(document).ready( function() {
    $("button").attr("disabled", "disabled");
    $("button").attr("title", "Bitte zuerst alle Daten eingeben!");
    $("button").css("background-color", "#E77,");
  });
</script>
```

### Aufgabe 3:

(max. 3 + 1 + 2 Punkte)

- Kapseln Sie einen Zähler in einer Closure, die eine passende Schnittstelle exportiert.
- Implementieren Sie einen Test, der demonstriert, dass der Zähler funktioniert.
- Erläutern Sie, wie die Closure funktioniert, und wofür sie sinnvoll einsetzbar ist.

