

(10P) HTML

Validieren Sie folgendes HTML und finden Sie Fehler und Bad Practices. Korrigieren Sie die Fehler in der Zeile unterhalb oder beschreiben Sie die Lösung.

```
<!--Document type html-->
```

```
<html>
```

```
<header>
```

```
<title>Cars</title>
```

```
<meta charset="uft-8" />
```

```
</header>
```

```
<body>
```

```
<div font="Arial" id="theUniqueContainer">
```

```

```

```
</img>
```

```
<p id="theUniqueContainer">
```

```
</div>
```

```
</p>
```

```
<form action="./submit" method="POST">
```

```
<textarea name="comment" size=5></textarea>
```

```
</form>
```

```
<input type="submit" value="send" />
```

<body>

<foot>

<script src="./jquery.js">

</foot>

</html>

(8P) Evaluieren

Evaluieren Sie die alert()-Ausgaben und schreiben Sie den Output, welcher im Alert Fenster sichtbar wäre, rechts auf die Linie.

| | |
|---|---|
| <i>p:nth-of-type(2)</i>
<i>p:nth-of-type(2n)</i> | <i>The :nth-of-type(n) selector matches every element that is the nth child, of a particular type, of its parent.
n can be a number, a keyword, or a formula.
Using a formula (an + b). Description: a represents a cycle size, n is a counter (starts at 0), and b is an offset value.</i> |
| <i>p:last-child</i> | <i>Selects every <p> element that is the last child of its parent</i> |

```
<!DOCTYPE html><html><head> <script src="jquery-2.1.3.js"></script></head>
<body>
<html><body>
<h1>HTML Example</h1>
<div>
  <span class="first">1</span>
  <span class="span">2</span>
  <span>3</span>
  <span>4</span>
</div>
<div>
  <p>
    <span first="1">5</span>
    <span>6</span>
    <span>7</span>
  </p>
</div><script>

//alert($(".span[first]").text() );

//alert( $(".div > span:last-child").text());

//alert( $(".div :nth-of-type(2n)").text());

//alert( $(".div:nth-of-type(2)").text());

</script></body></html>
```

(8P) JavaScript

Was geben die folgenden console.log() Statements aus?

Hinweis: Geben Sie den Output von Strings wie folgt an: "ein Text"

```
console.log("3" + 2);
```

Output:

Begründung:

```
console.log(3 + "2");
```

Output:

Begründung:

```
console.log(typeof(parseInt(3)));
```

Output:

Begründung:

```
var x = 1;
var y = 2;
function print(x) {
    console.log(x); //I
    console.log(y); //II
    var x = 3;
    var y = 4;
}
print();
```

I

Output:

Begründung:

II

Output:

Begründung:

(6P) jQuery

Folgendes HTML und JavaScript File ist gegeben. Finden Sie 2 **grobe** Fehler und begründen Sie diese. Ziel dieses HTML ist, dass der eingegebene Text des jeweiligen Feld in den **Output** geschrieben wird und dessen Inhalt überschreibt. Es ist gewollt, dass nur immer ein Text im Output steht!

index.html:

```
<html>
<head lang="en">
  <script src="jquery.js"></script>
  <script src="ausgangslage.js"></script>
</head>
<body>
  <p>Vorname: <input id="vorname" type="text" /></p>
  <p>Nachname: <input id="nachname" type="text"/> </p>
  <div>
    <span>Hallo: </span>
    <span id="output"></span>
  </div>
</body>
</html>
```

ausgangslage.js:

```
function showField()
{
  $("#output").text($(this).val());
}

$("#vorname").change(function()
{
  showField();
});

$("#nachname").change(function()
{
  $("#output").text($(this).val());
});
```

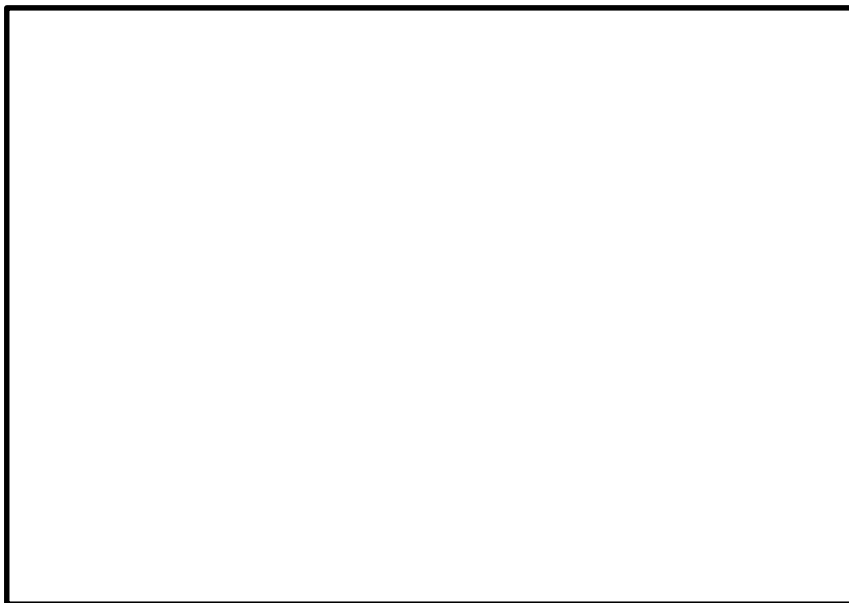
Fehler 1:

Fehler 2:

(10 P) CSS Interpretation

```
<html>
<head lang="en">
  <style>
    #first{
      border: 2px solid green;
    }
    .div{
      background: blue;
      padding: 10px;
      border: 2px solid black;
      width: 33.3%;
      box-sizing: content-box;
    }
  </style>
</head>
<body>
<div style="width: 50%">
  <input class="div" value="1" id="first"><!--
--><input class="div" value="2"><!--
--><input class="div" value="3">
</div>
</body>
</html>
```

A) Zeichnen Sie das resultierende Layout in folgendes Browser-Fenster:



(10 P) Template Engine

Ihre Aufgabe ist eine primitive
 Template Funktionalität
 nachzubauen ohne Compile
 Funktionalität. Das Template ist
 vorgeben.

id name

Id: myId

Name: myName

Die Eingabe soll aufs Template gemapped werden und dann wie im Screenshot angezeigt, in den
 vorgesehenen Container hinzugefügt werden. **Die Lücke** im HTML sollte sinnvoll ergänzt werden.

Hinweis: JQuery ist vorhanden

```

<!DOCTYPE html>
<html><head> <script src="jquery-2.1.3.js"></script></head><body>
<div>
  <div id="template" _____ >
    <p>Id: {{{id}}}</p>
    <p>Name: {{{name}}}</p>
  </div>
  <span>id </span><input id="id">
  <span>name </span><input id="name">
  <button id="add">Add</button>
</div>
<div id="container">
<div>
<script>

function applyTemplate(template, dataObject) {

```

```

}
$("#add").click(function () {

```

```

});
</script></body></html>

```

(8 P) Allgemeine Fragen

Welche Bedeutung haben die width und height Angaben bei den folgenden Tags? (2P)

```
  
<video src="..." width="720" height="480" poster="..." />
```

Wieso sollten Sie **br** nicht benutzen, um eine neue Zeile zu erzeugen und **small** nicht benutzen, um kleinen Text anzuzeigen? (2P)

Ihr Kollege hat einen Header erstellt, indem sich Logo und Text nebeneinander befinden. Dazu hat er eine Tabelle benutzt. Wieso ist das keine gute Idee? (2P)

Schreiben Sie folgendes Snippet um und nutzen Sie jQuery: (2P)

```
var request = new XMLHttpRequest();  
request.onreadystatechange = function() {  
    if(request.readyState == 4 && request.state == 200) {  
        console.log(request.responseText);  
    }  
};  
request.open('GET', '/time');  
request.send();
```