

Web-Applikation Aufgaben: Erste divs und Test-Button zufügen

Das <div> Element ist sehr wichtig. Die Webseite werden mit <div> Elemente aufgebaut.

Die erste <div> Elemente

Vier Rechtecken (divs) werden zugefügt: Blau, gelb, rot und grün

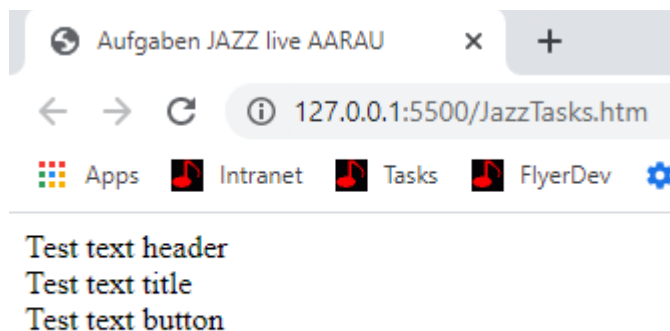
The screenshot shows a web application form titled "JAZZ live AARAU Aufgaben". The form is enclosed in a blue border. At the top, there is a yellow header bar with the title. Below the header, there is a dropdown menu labeled "Aufgabe wählen". Below this, a red rectangle highlights the "Aufgabe Titel" input field. To the left of the "Aufgabe Titel" field is a large text area labeled "Beschreibung". Below the "Beschreibung" field is a text field labeled "Bemerkung". Below the "Bemerkung" field is a text field labeled "Datum Erinnerung". To the right of the "Beschreibung" field is a dropdown menu labeled "Referenz wählen". Below the "Referenz wählen" dropdown is a text field labeled "Referenz Link". Below the "Referenz Link" field is a text field labeled "Referenz Beschreibung". Below the "Referenz Beschreibung" field is a text field labeled "Datum Fertig". Below the "Datum Erinnerung" and "Datum Fertig" fields are three text fields labeled "Link DOC Dokument", "Link PDF Dokument", and "Link HTML Webseite". Below the "Link HTML Webseite" field is a text field labeled "Verantwortlich". To the right of the "Verantwortlich" field is a dropdown menu labeled "Stellvertreter". Below the "Verantwortlich" field is a button labeled "Intranet aktualisieren". To the right of the "Intranet aktualisieren" button is a button labeled "Aufgabe speichern". At the bottom left, a green rectangle highlights a button labeled "Test".

Im Haupt-HTML-Datei die folgenden Elemente zufügen:

```
<div id= 'id_tasks_page' class= 'cl_tasks_page'>  
    <div id= 'id_tasks_header' class= 'cl_tasks_header'>Test text header</div>  
    <div id= 'id_tasks_task_title' class= 'cl_tasks_task_title'>Test text title</div>  
    <div id= 'id_tasks_test_button' class= 'cl_tasks_test_button'>Test text button</div>  
</div>
```

```
17 <body onload= "initJazzTasks()">  
18  
19     <div id= 'id_tasks_page' class= 'cl_tasks_page'>  
20  
21         <div id= 'id_tasks_header' class= 'cl_tasks_header'>Test text header</div>  
22  
23         <div id= 'id_tasks_task_title' class= 'cl_tasks_task_title'>Test text title</div>  
24  
25         <div id= 'id_tasks_test_button' class= 'cl_tasks_test_button'>Test text button</div>  
26  
27     </div>  
28 </body>
```

Speichern und 'Open with Live Server'



Styles zufügen

In der Style Datei JazzTasks.css Klassen zufügen.

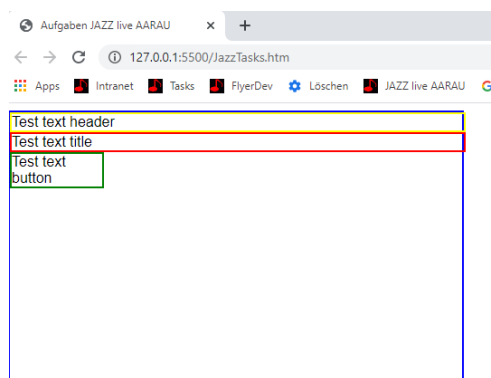
```
.cl_tasks_page
{
border: solid 2px blue;
width: 500px;
height: 300px;
}

.cl_tasks_header
{
border: solid 2px yellow;
width: 100%;
}

.cl_tasks_task_title
{
border: solid 2px red;
width: 100%;
}

.cl_tasks_test_button
{
border: solid 2px green;
width: 100px;
}
```

Speichern



CSS für Farben und die Platzierung von divs und Texte

Ändere Text 'Test text header' zu 'JAZZ <i>live</i> AARAU Aufgaben'

In der Klasse cl_tasks_page folgendes zufügen

```
font-family: 'Arial';
```

In der Klasse cl_tasks_header folgendes zufügen

```
text-align: center;
```

```
background-color: black;
```

```
color: rgb(255, 0, 40);
```

```
font-size: 20px;
```

```
font-weight: bold;
```

In der Klasse cl_tasks_task_title folgendes zufügen

```
margin-top: 20px;
```

Test-Button sollte nicht so nahe 'header' und die Kante sein

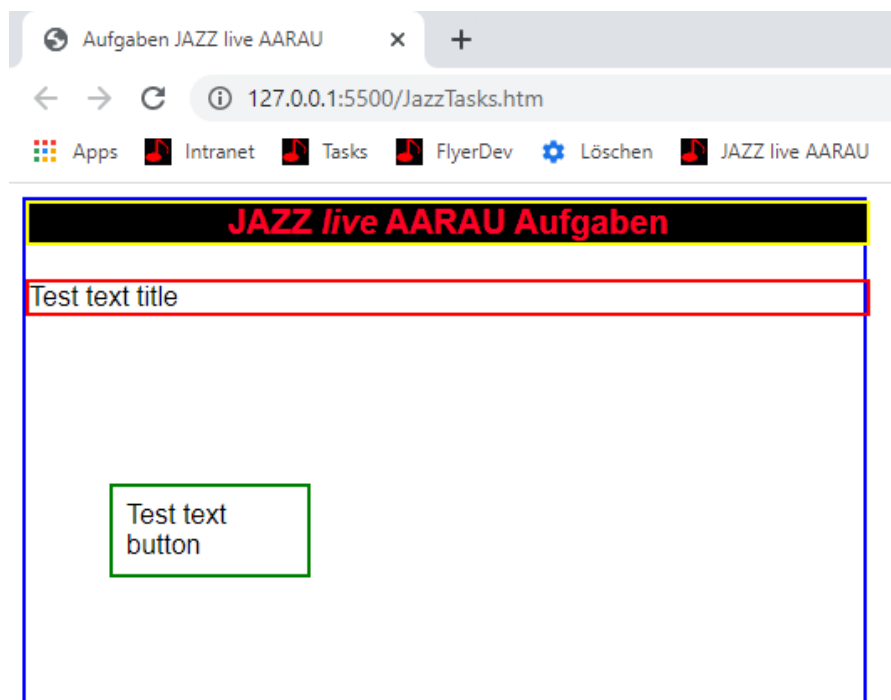
In der Klasse cl_tasks_test_button folgendes zufügen

```
margin-top: 100px;
```

```
margin-left: 50px;
```

```
padding: 8px;
```

So sollte es jetzt aussehen



Mit Debugger Elemente anschauen

Debugger mit Funktion F12 starten, Elements wählen und Texte zeigen (öffnen) mit den Pfeilen



Alle HTML Elemente (Objekte) haben Styles (Eigenschaften) wie identity (id), padding, margin, border, backgroundcolor, etc.

Der Browser setzt vorgestellte Werte (default values) für alle Elemente. Zum Beispiel keine 'borders' zu zeigen.

Mit dem Debugger kann man alle Werte anschauen. Zum Beispiel für den Test-Button.

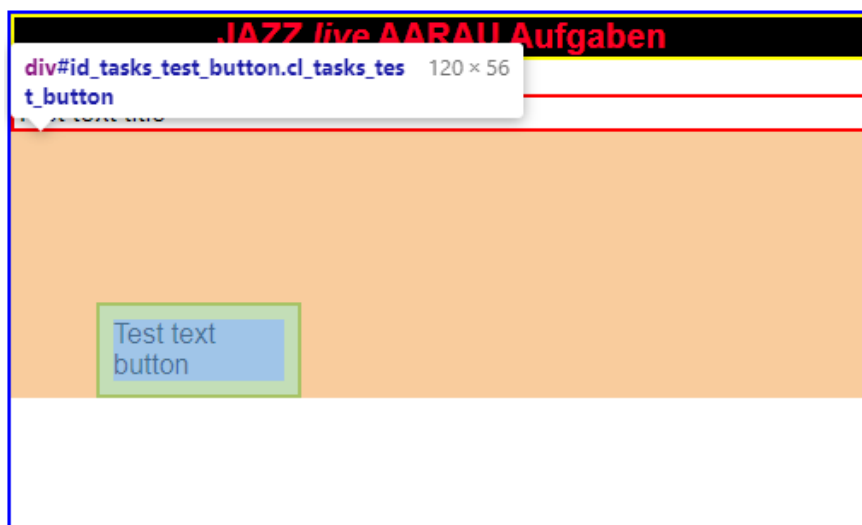
Fehler wie zum Beispiel, dass in der HTML Datei und in der CSS Datei ein Klassenname nicht gleich geschrieben wurde, kann man schnell mit dem Debugger herausfinden. (Und dieser Fehler kommt ziemlich häufig vor).

Bitte Zeile/<div> Test-Button markieren und lass den Cursor auf der Zeile.

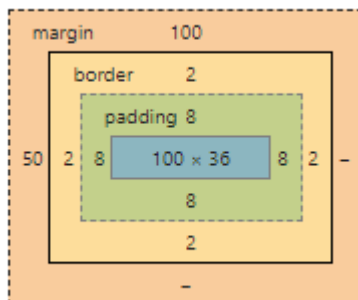
```
</div>
<div id="id_tasks_task_title" class="cl_tasks_task_title">Test text title</div>
... <div id="id_tasks_test_button" class="cl_tasks_test_button">Test text button</div>
```

Identität (#) und Klasse wird gezeigt

margin, padding, werden mit Farben gezeigt



Auch in einem separaten Bild mit den Zahlen



Die in der CSS Datei definierte Styles werden gezeigt und es gibt Links



Test-Button

Wir machen jetzt keinen richtigen Button (mit `<button>` oder `<input>`). Die meisten HTML Elemente können zu Buttons werden, d.h. wenn man das Element klickt passiert etwas. Man muss nur eine 'on-click' Funktion (event function) mit JavaScript definieren und das `<div>` Element das Attribut `onclick= 'eventFunctionName()'` hinzufügen.

Zuerst die Event-Funktion in einer separaten JavaScript Datei definieren. Name zum Beispiel `JazzTasksTests.js`. Wie man das macht ist in `TasksLessonCreateDirectoriesAndStartfiles.docx` beschrieben.

Die Funktion kann man zum Beispiel `onClickOfTestButton` nennen.

Bitte hinzufügen

// Beschreibung wann die Funktion angerufen wird und was die Funktion macht

```
function onClickOfTestButton()
{
    alert("Enter onClickOfTestButton");
} // onClickOfTestButton
```

In der Haupt-HTML-Datei `onclick` hinzufügen (bitte zuerst w3school `onclick` Beispiele anschauen und Übungen machen)

```
<div id= 'id_tasks_test_button' class= 'cl_tasks_test_button' onclick="onClickOfTestButton()">Test text button</div>
```

Man soll machen so dass der Benutzer sieht, dass man das `<div>` Element klicken kann. Das macht man mit CSS

In der Klasse `cl_tasks_test_button` folgendes hinzufügen:

`cursor: pointer;`

(Bitte zuerst `cursor pointer` in w3schools anschauen und Übungen machen)

Speichern und testen.

Event: Farbe ändern

Hintergrundfarben kann mit CSS gesetzt werden, aber auch mit JavaScript. (Diese Übung hast du auch mit den vier Lektionen gemacht, glaube ich).

Um ein Style (Eigenschaft ändern zu können) muss man das Element «erreichen».

Ein Variabel machen, der das Element «ist».

Die ganze Webseite ist ein 'document'. Wenn man ein spezielles Element möchte kann man die Identität verwenden: Gib mir das Element mit id= ...

Bitte in der Event-Funktion folgendes zufügen

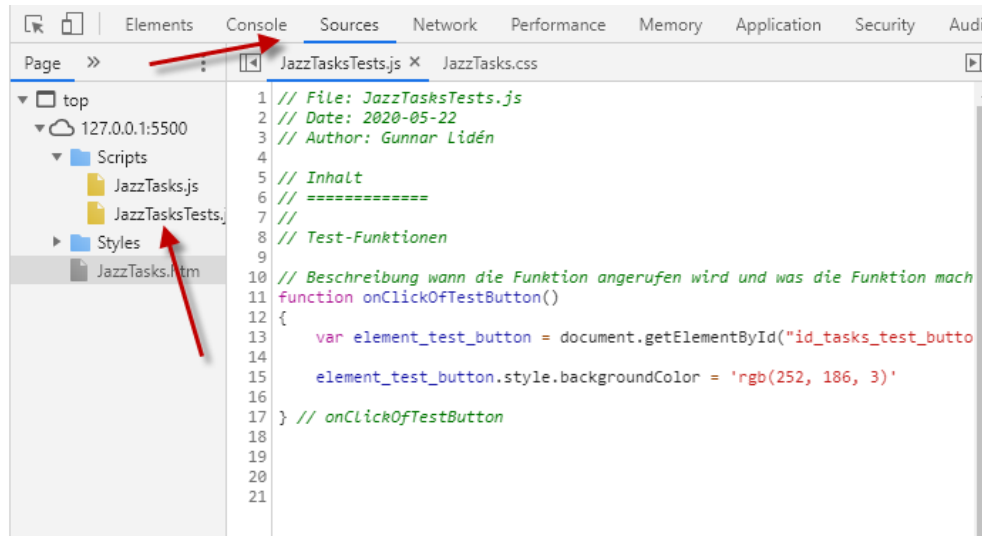
```
var element_test_button = document.getElementById("id_tasks_test_button");  
element_test_button.style.backgroundColor = 'rgb(252, 186, 3)';
```

(Bitte zuerst Information über Javascript getElementById und backgroundColor in w3schools suchen und Übungen machen).

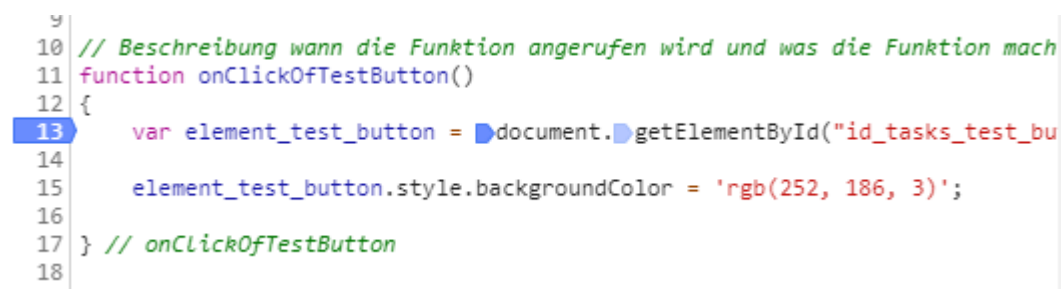
Speichern und testen.

Debugger Breakpoints

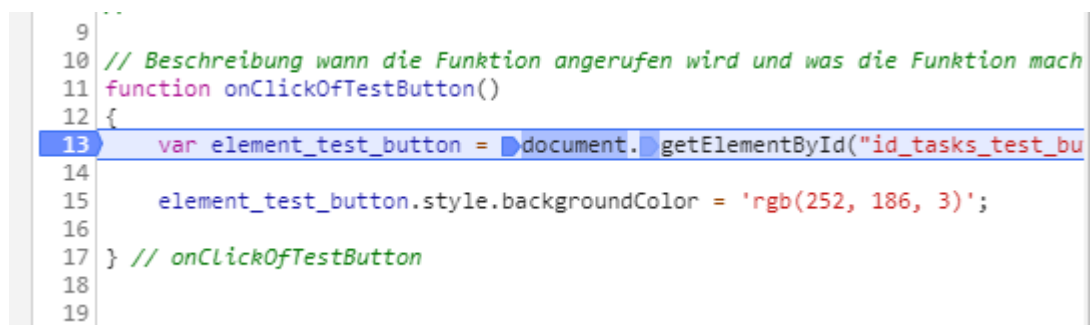
Debugger mit F12 starten und Sources JazzTasksTests.js wählen



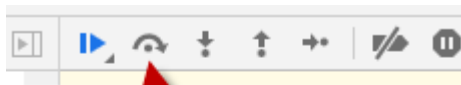
Auf der Zahlen-Spalte klicken (für mich Zeile 13)



Auf den Test-Button klicken.



Mit F10 oder Icon zu nächster Zeile



oder F10

zu nächster Zeile

```
10 // Beschreibung wann die Funktion angerufen wird und was die Funktion mach
11 function onClickOfTestButton()
12 {
13     var element_test_button = document.getElementById("id_tasks_test_bu
14     element_test_button.style.backgroundColor = 'rgb(252, 186, 3)';
15 } // onClickOfTestButton
16
17
18
--
```

Man kann also jede Zeile machen, und schauen ob man richtig programmiert hat.

Noch einmal F10, dann ändert sich die Farbe

Mit F8 oder Icon wird das Programm wieder laufen



oder F8

**resume
execution**