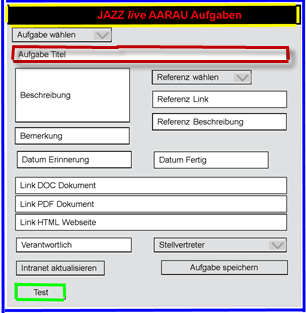
**Datei: TasksLessonFirstDivsTestButton.docx 2020-05-22**

**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



***Im Haupt-HTML-Datei die folgenden Elemente zufügen:***

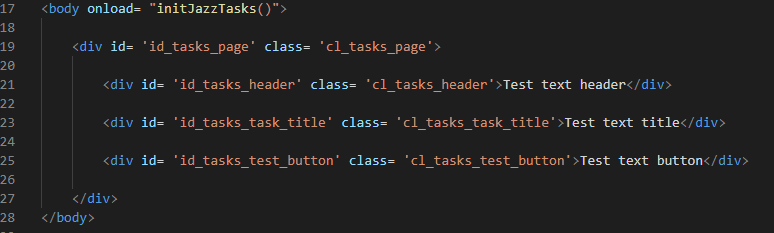
**<div id= 'id\_tasks\_page' class= 'cl\_tasks\_page'>**

**<div id= 'id\_tasks\_header' class= 'cl\_tasks\_header'>Test text header</div>**

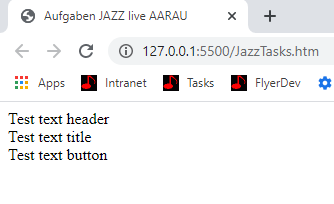
**<div id= 'id\_tasks\_task\_title' class= 'cl\_tasks\_task\_title'>Test text title</div>**

**<div id= 'id\_tasks\_test\_button' class= 'cl\_tasks\_test\_button'>Test text button</div>**

**</div>**



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**

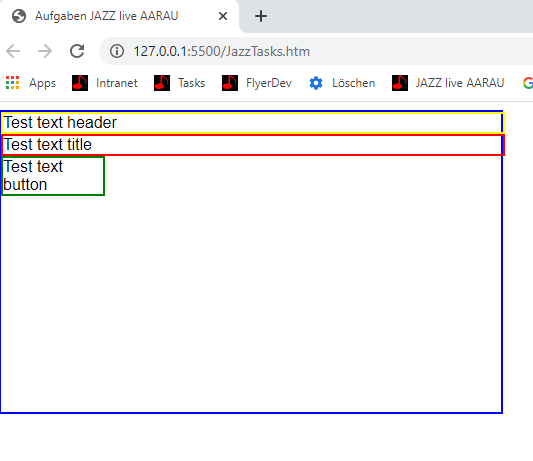
**{**

**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

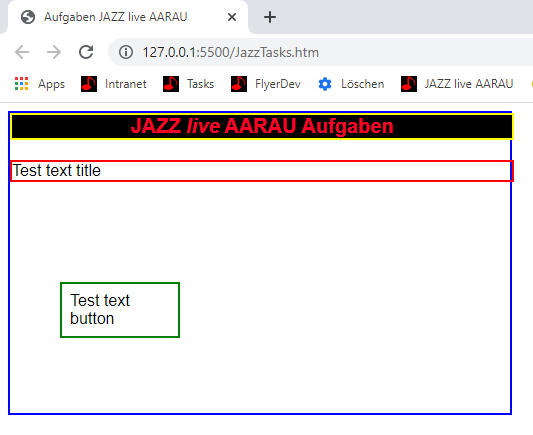
In der Klasse cl\_tasks\_test\_button folgendes zufügen

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 (Eigenshaften) wie identity (id), padding, margin, border, backgroundcolor, etc.

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

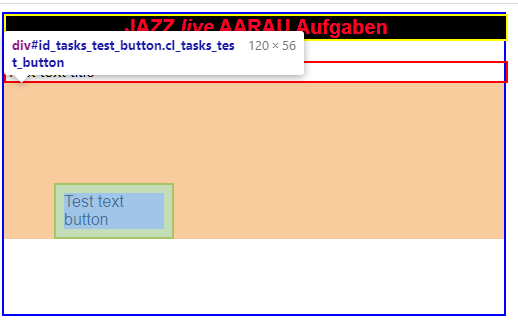
Mit dem Debugger kann man alle Werte anschauen. Zum Beispiel für den Test-Button.

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).

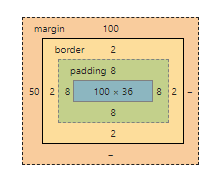
Bitte Zeile/<div> Test-Button markieren und lass den Cursor auf der Zeile.



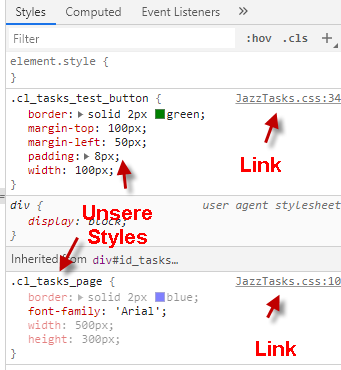
Identität (#) und Klasse wird gezeigt

margin, padding, werden mit Farben gezigt

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()’ zufügen.

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 zufügen

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

function onClickOfTestButton()

{

alert("Enter onClickOfTestButton");

} // onClickOfTestButton

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



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 zufügen:

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");

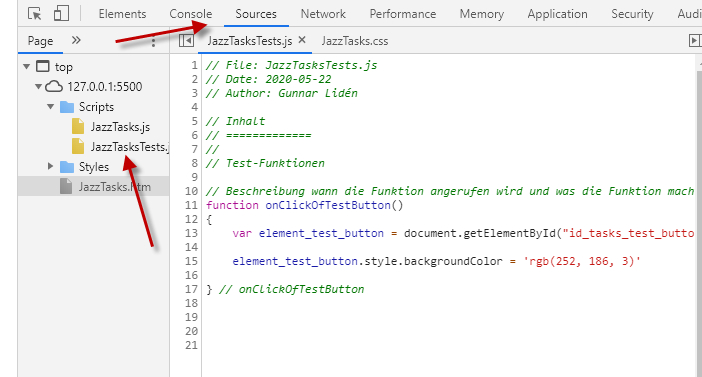
element\_test\_button.style.backgroundColor = 'rgb(252, 186, 3)';

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

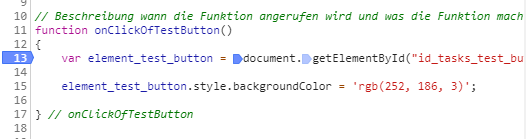
Speichern und testen.

***Debugger Breakpoints***

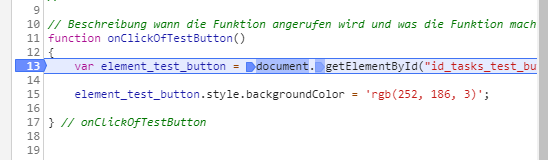
Debugger mit F12 starten und Sources JazzTasksTests.js wählen



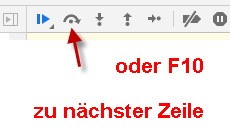
Auf der Zahlen-Spalte klicken (für mich Zeile 13)

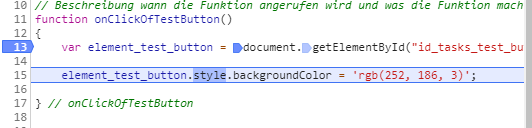


Auf den Test-Button klicken.



Mit F10 oder Icon zu nächster Zeile





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

