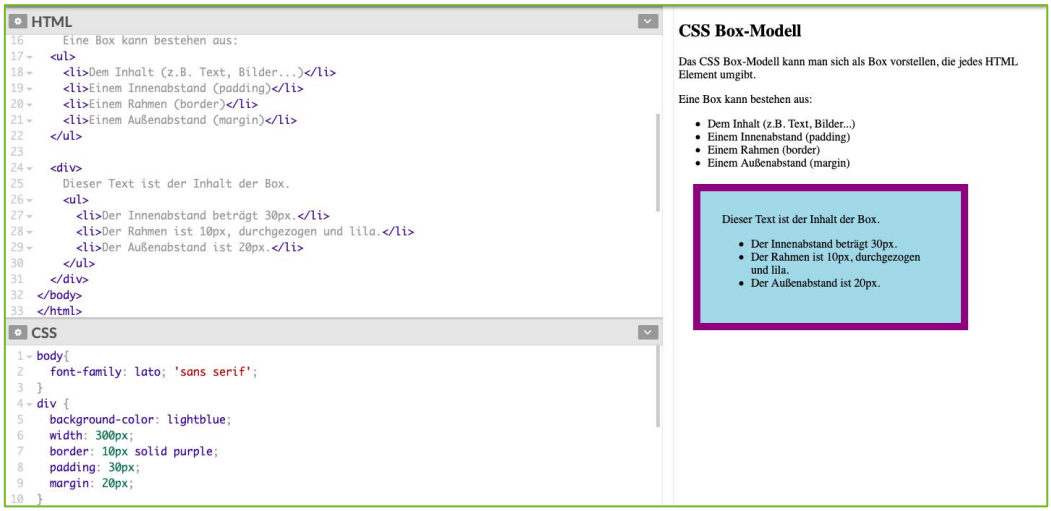
3.1 CSS Box-Modell

APP CAMPS

Worum geht es in dieser Übung?

- Abstände und Rahmen von HTML Elementen (CSS Box-Modell)

URL: https://codepen.io/appcamps/pen/zbJNvN



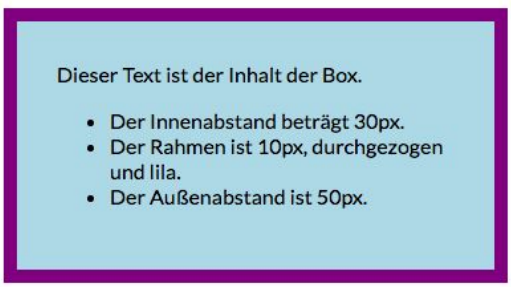
Aufgaben

1. Lies dir unten die Infos zum CSS Box-Modell durch.
2. Passe im Beispiel folgende Werte wie folgt an:
   a. Breite: 500 Pixel.
   b. Rahmen: 20 Pixel, gepunktet (dotted), grün (green).
   c. Innenabstand: 10 Pixel.
   d. Außenabstand: 60 Pixel.
3. Füge im HTML ein zweites div Element hinzu mit einer Klasse, z.B. . Als Inhalt kannst du einen beliebigen Text einfügen.
4. Füge im CSS eine Klasse .meindiv{...} ein und definiere Breite, Rahmen, Innenabstand, Außenabstand.

Infos

Man kann sich das **CSS Box-Modell** als Box vorstellen, die jedes HTML Element (nicht nur div Elemente) umgibt.
Eine Box kann bestehen aus:

- Dem Inhalt (z.B. Text, Bilder...)
- Einem Innenabstand (padding) zwischen Inhalt und Rahmen.
- Einem Rahmen (border) der den Inhalt (und ggf. den Innenabstand) umgibt.
- Einem Außenabstand (margin) der den Abstand zu anderen Elementen definiert.



Man kann padding, border und margin definieren für alle Seiten (d.h. oben, rechts, unten und links). Dann schreibt man z.B. **padding: 30px;**

Oder man gibt an, dass man nur für eine bestimmte Seite gilt, z.B. padding-top: 30px;
Manchmal sieht man auch so eine Schreibweise: **padding: 50px 30px 50px 80px;** Der erste Wert gilt für top, der zweite für right, der dritte für bottom und der vierte Wert gilt für left. So kann man unterschiedliche Werte für alle Seite angeben, ohne es explizit mit -top, -right, -bottom und -left zu schreiben.

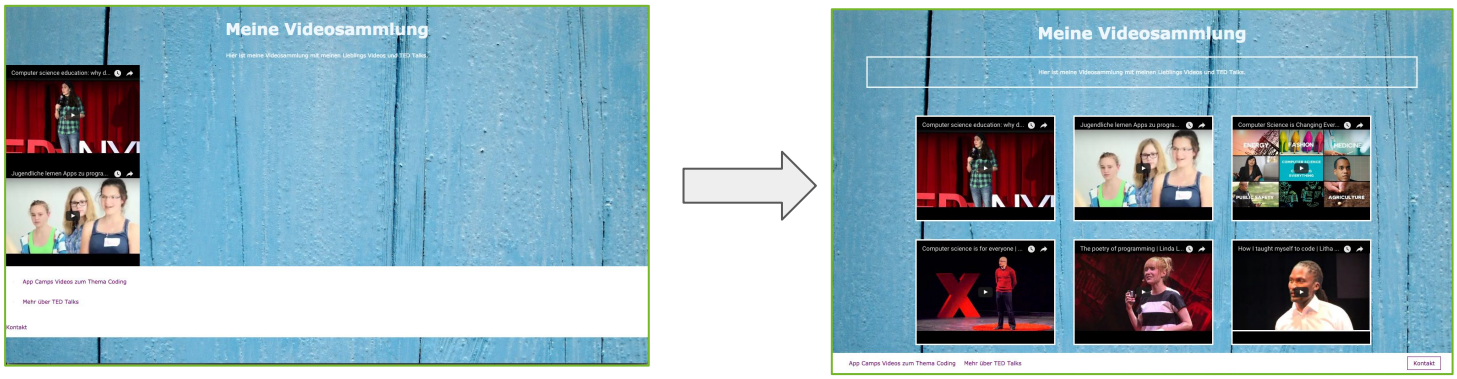
Wichtig bzgl. Rahmen: Bei Rahmen muss man immer die Linienart angeben, z.B. durchgehende Linie (solid), gepunktete Linie (dotted), doppelte Linie (double) usw. Der Rahmen in dem Beispiel oben ist so beschrieben: **border: 10px solid purple;**

3.2a Videoseite gestalten (1/2)

Worum geht es in dieser Übung?

- Abstände und Rahmen für eine Beispiel-Videoseite einfügen
- Anordnung von Elementen

URL: https://codepen.io/appcamps/pen/ywxggO



Aufgaben

Wir arbeiten uns Schritt für Schritt von der linken Abbildung zur rechten Abbildung. Passe zuerst alle Schritte im CSS an. Dann kannst du anschließend eigene Inhalte (Videos, Texte usw.) einfügen.

1. Wir starten mit Rahmen und Abständen.

a. Finde **.video .thumbnail { }** und trage dort ein bzw. ergänze:
border: 5px solid #fff; → #fff; ist die HEX Schreibweise für white;

b. Finde **.description{ }** und trage dort ein bzw. ergänze:
border: 4px solid aliceblue;
margin: 30px 100px; → 30px gelten für top und bottom, 100px für right und left.

c. Finde **.video-gallery{ }** und trage dort ein bzw. ergänze:
margin-top: 20px; → Außenabstand der nur für top (oben) gilt.

d. Finde **.video{ }** und trage dort ein bzw. ergänze:
margin: 30px; → Außenabstand der für alle vier Seiten gilt (top, right, bottom, left)

2. Jetzt geht es um die Anordnung der Elemente.

a. Finde **.video-gallery { }** und trage dort ein bzw. ergänze:
display: flex; → display bestimmt wie Elemente angeordnet werden, flex = nebeneinander
flex-wrap: wrap; → bricht in neue Zeile um, wenn Platz nicht ausreicht
justify-content: center; → Platzierung der Elemente wenn nicht der gesamte Platz gebraucht wird

b. Finde **nav li { }** und **nav ul { }** und trage dort jeweils ein bzw. ergänze:
display: inline; → li ist ein Block Element, d.h. die Elemente werden untereinander angezeigt. Durch display: inline werden die Elemente nebeneinander angezeigt.

3.2b Videoseite gestalten (2/2)

Aufgaben

3. Links und Button, um Kontakt aufzunehmen.

a. Finde **contact-button { }** und trage dort ein bzw. ergänze:
   float: right; → das Element “umfließt” (liegt) rechts neben anderen Elementen

b. Finde **contact-button a { }** und trage dort ein bzw. ergänze:
   cursor: pointer; → Macht aus dem Mauszeiger ein Handsymbol
   margin-right: 30px;
   padding: 8px 18px;
   border: 1px solid darkmagenta;
   position: relative; → Keine fixe Positionierung (passt sich an)

c. Finde **a:hover { }** und trage dort ein bzw. ergänze:
   color: black; → Wenn man mit der Maus über einen Link (a) fährt, ändert sich die Farbe

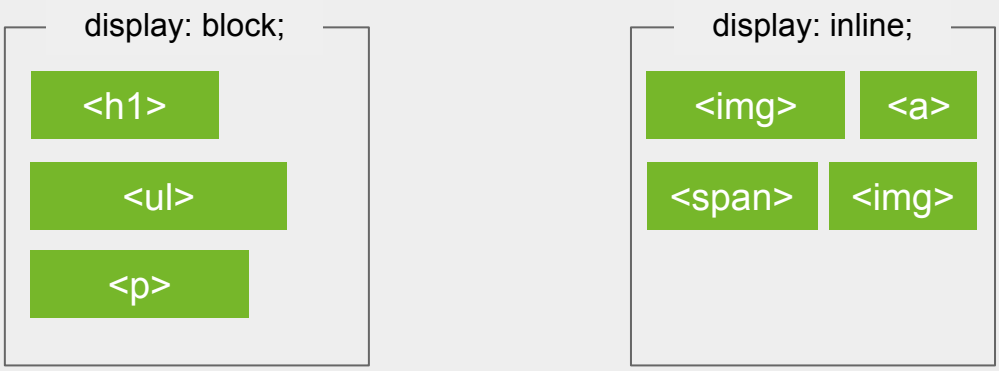
Hast du alle CSS Anweisungen angepasst? Super, dann sollte deine Seite jetzt so aussehen, wie auf dem rechten Bild.
Die Musterlösung findest du auch hier: https://codepen.io/appcamps/pen/gEdgRE

Infos

Man unterscheidet in HTML zwischen **Block** Elementen und **Inline** Elementen.

- Block Elemente kannst du dir wie Bauklötze vorstellen, die gestapelt werden. Sie liegen standardmäßig übereinander. Nachfolgende Elemente werden in einer neuen Zeile positioniert. h1 bis h6, ul, li, div und p Elemente sind Block Elemente.

- Inline Elemente liegen nebeneinander und brechen erst in die nächste Zeile um, wenn kein Platz mehr ist. img, a, span sind Inline Elemente.



- Die display-Eigenschaft der Elemente kann mit CSS geändert werden. Man kann Block Elementen die Eigenschaft **display: inline;** zuweisen. So machen wir das im Beispiel mit <li>.

```
li {
  display: inline;
}
```

Genauso kann man auch Inline Elementen **display: block;** zuweisen. Dann werden sie untereinander angezeigt.

3.3 Seite lokal speichern und anpassen

APP CAMPS

Worum geht es in dieser Übung?

- Die Seite lokal speichern
- HTML (und evtl. CSS) anpassen für eigene Seite

URL: https://codepen.io/appcamps/pen/gEdgRE (oder dein Ergebnis nach 3.2b)

Aufgaben

1. Speichere die Seite lokal bei dir ab. Falls du nicht mehr genau weißt, wie das geht, kannst du auf Lernkarte 1.5 nachschauen. Erinnerung: CSS musst du über <link …> im <head> Bereich einbinden.

2. Ändere den Inhalt der Seite, d.h. das HTML:
   a. Füge eigene Videos ein.
   b. Schreibe einen eigenen Beschreibungstext.
   c. Füge Links ein.
   d. Trage im Kontakt Button eine eigene Kontakt E-Mail Adresse ein.

3. Ändere das Design der Seite, d.h. das CSS:
   a. Verwende ein anderes Hintergrundbild.
   b. Passe die Farben an.
   c. Ändere die Rahmen.

4. Bist du mit allem fertig? Super. Dann kannst du dir jetzt überlegen, was du noch anpassen kannst.

Wenn du mehr über das **Box Modell** und **Inline & Block Elemente** wissen willst, empfehlen wir dir folgende Links:

- Box Modell: http://www.w3schools.com/css/css_boxmodel.asp
- Display (Block vs. Inline): http://www.w3schools.com/css/css_display_visibility.asp