

IDA Digital Design

29.04.2021

Programm

Tag 7

- CSS Transform
- CSS Transitions
- Übungszeit
- Coachings
- Abschluss: Make, share, give, participate, support

17:30 Vortrag IDA Salon

Besprechung

05.05.2021 ?

Mit Nick vor Ort?

Layout

CSS Transformationen

transform Funktionen

x - Achse



y - Achse



Bin ich
blau?

Der CSS-Datentyp `<transform-function>` stellt eine Transformation dar, die das Aussehen eines Elements beeinflusst. Transformationsfunktionen können ein Element im 2D- oder 3D-Raum drehen, in der Größe verändern, verzerren oder verschieben. Er wird in der Eigenschaft "transform" verwendet.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function>

```
/* Beispiel */  
transform: translate(100px, 200px);
```

TRANSFORM UND TRANSLATE

translate

```
/* Single <length-percentage> values
*/
transform: translate(200px);
transform: translate(50%);
```

```
/* Double <length-percentage> values
*/
transform: translate(100px, 200px);
transform: translate(100px, 50%);
transform: translate(30%, 200px);
transform: translate(30%, 50%);
```

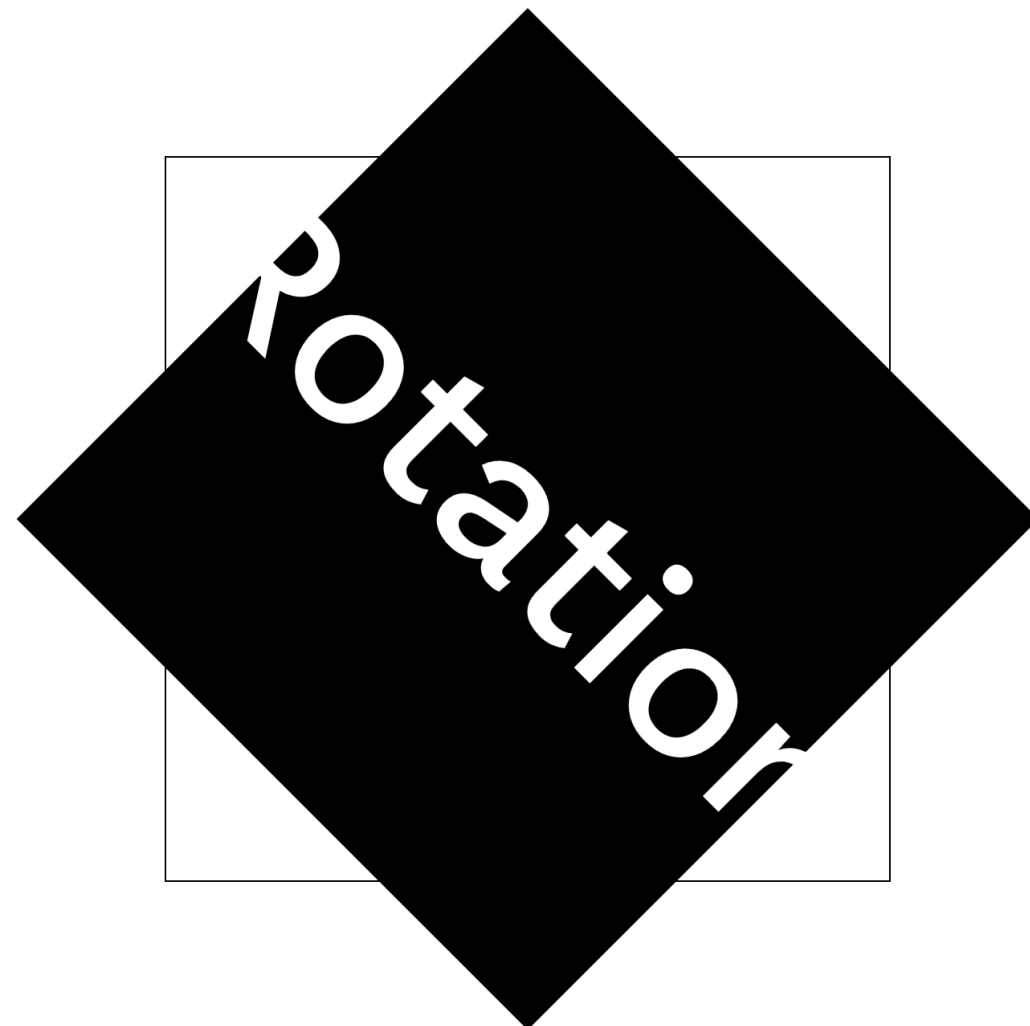
Die CSS-Funktion `translate()` positioniert ein Element in horizontaler und/oder vertikaler Richtung neu. Ihr Ergebnis ist ein `<transform-function>`-Datentyp.

[https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/translate\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/translate())

TRANSFORM UND ROTATE

rotate

```
/* Beispiel 45 Grad Uhrzeigersinn */  
transform: rotate(45deg);
```

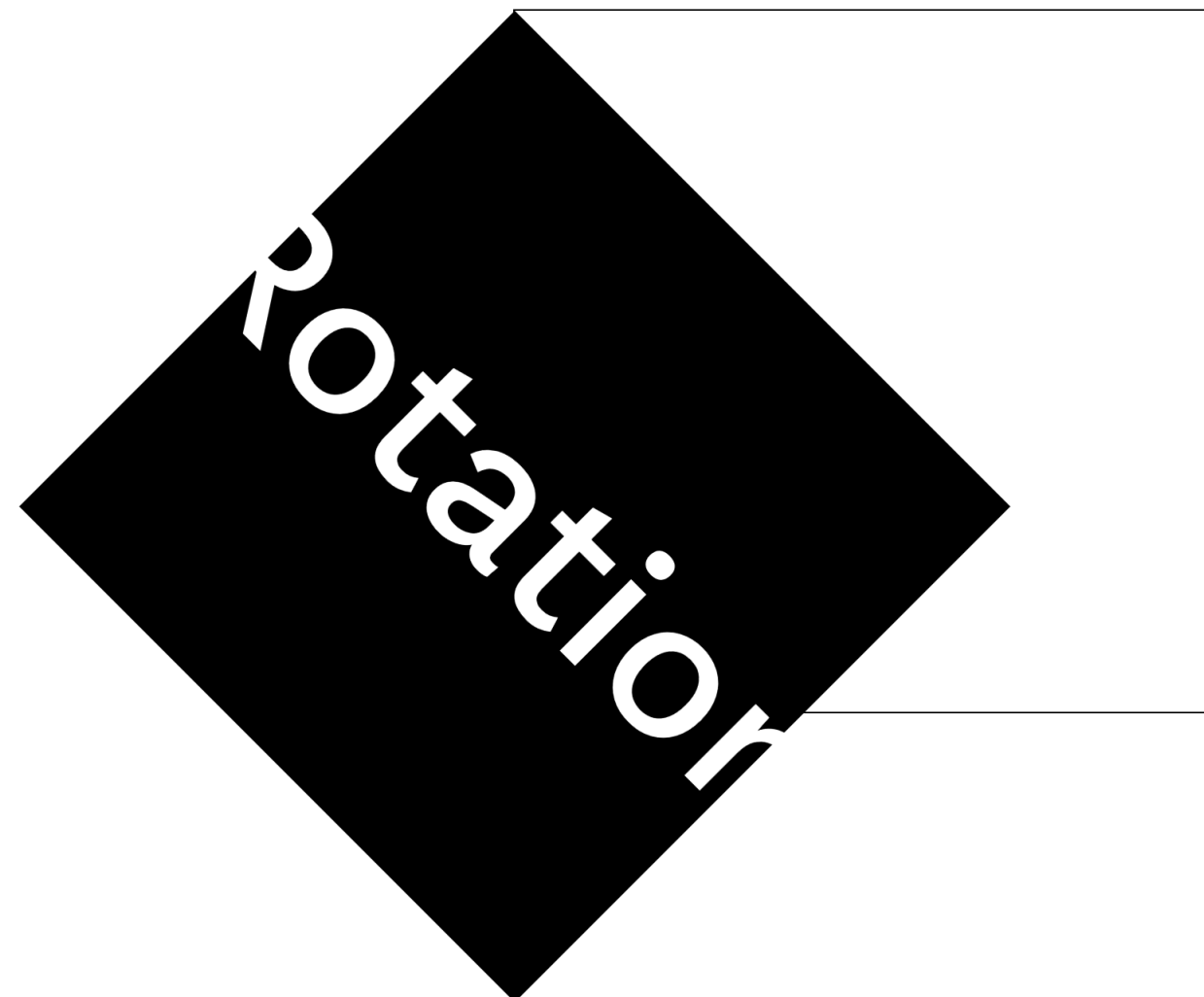


Die CSS-Funktion rotate() definiert eine Transformation, die ein Element um einen festen Punkt auf der 2D-Ebene dreht, ohne es zu deformieren. Ihr Ergebnis ist ein <transform-function>-Datentyp.

[https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/rotate\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/rotate())

transform-origin

```
/* Beispiel */  
transform-origin: top left;  
transform-origin: 50px 50px;
```



Der Fixpunkt, um den das Element rotiert - wie oben erwähnt - wird auch als Transformationsursprung bezeichnet. Standardmäßig ist dies der Mittelpunkt des Elements, aber Sie können Ihren eigenen Rotationsursprung mit der Eigenschaft transform-origin festlegen.

Standardmässig ist der Rotationsmittelpunkt im Zentrum des Elements.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform-origin>

TRANSFORM UND SCALE

scale

```
/* Beispiel */  
transform:scale(0.8);  
transform:scale(0.8, 2);
```



Scale

Die CSS-Funktion `scale()` definiert eine Transformation, die ein Element auf der 2D-Ebene skaliert. Ihr Ergebnis ist ein `<transform-function>`-Datentyp.

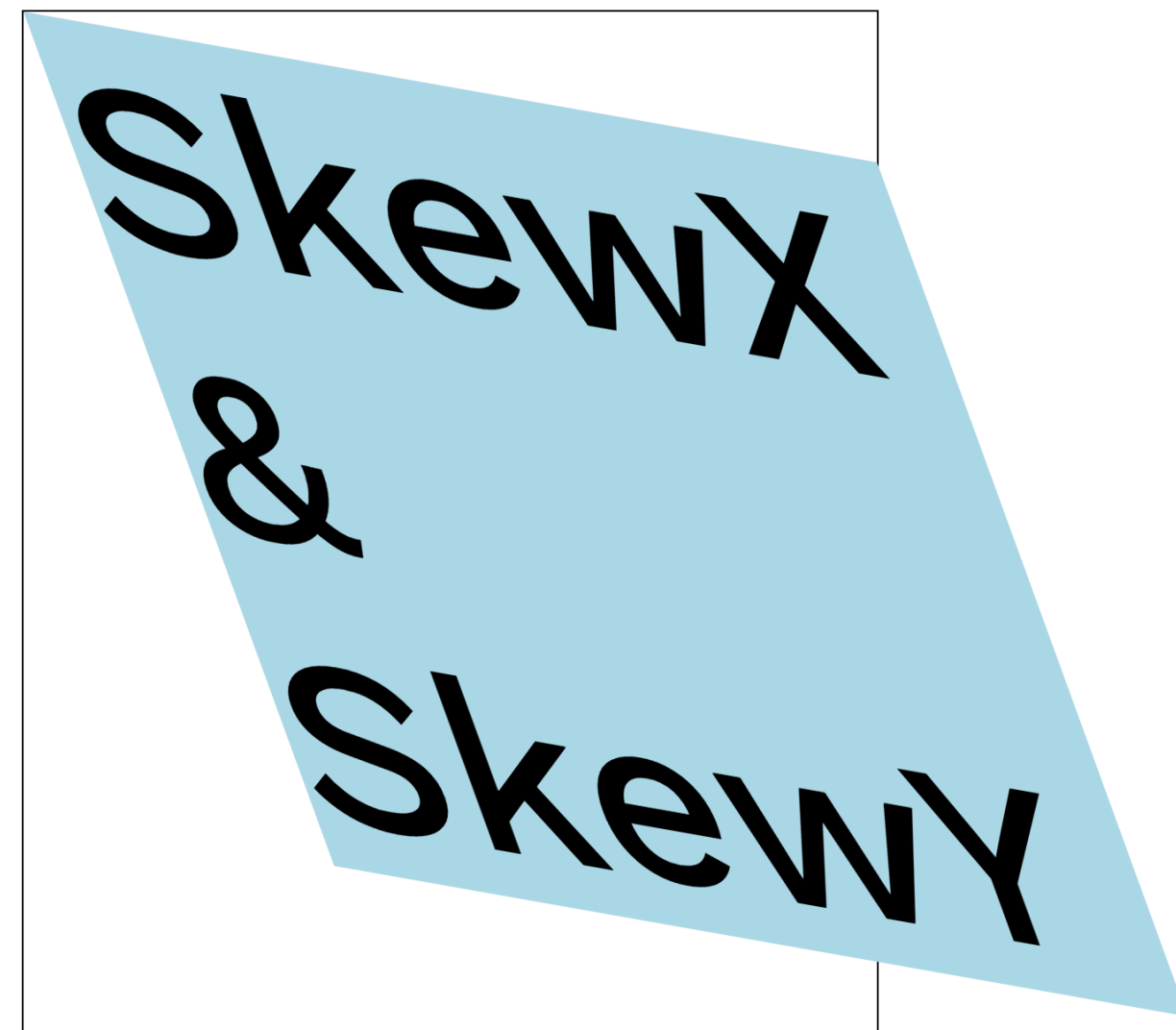
[https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/scale\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/scale())

(Beachtet, wie die Funktion alle Properties skaliert, auch die Schrift.)

TRANSFORM UND SKEW

skew

```
/* Beispiel */  
transform:skewX(10deg);  
transform:skew(20de, 10deg);
```



Diese Transformation ist eine Verzerrung, die jeden Punkt innerhalb eines Elements um einen bestimmten Winkel in horizontaler und vertikaler Richtung verzerrt. Der Effekt ist so, als ob Sie jede Ecke des Elements anfassen und sie entlang eines bestimmten Winkels ziehen würden.

skewX in der x-Richtung, skewY in der y-Richtung, skew in beiden Richtungen. Die Angaben erfolgen in Grad (10deg).

[https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/skew\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/skew())

Kombinationen

```
/* Falsch !!*/
```

```
transform:scale(0.8);  
transform:rotate(45deg);
```

```
/* Richtig !!*/
```

```
transform:scale(0.8) rotate(45deg);
```

Die Kombination von verschiedenen transformationen schreibt ihr hintereinander in einen einzige transform Angabe.

Bei der falschen Schreibweise oben würde bloss rotate ausgeführt, weil die Angabe scale überschreibt – sie kommt nachher (Cascading Rules).

Alle transform Funktionen im Überblick

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

Aufgabe 1

Transform Parts of your Robot

Todo

Arbeite durch die Exercises: [https://www.w3schools.com/css/exercise.asp?](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_2dtransforms1)

[filename=exercise_css3_2dtransforms1](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_2dtransforms1)

Dann:

Versuche, weitere Shapes deines Robots anzuzeigen und mit Transforms zu verformen. Es muss noch nicht einmal ein zusammenhängendes Wesen geben. Hab Spass, versuche mehrere Transformationen zu kombinieren.

Zustandsveränderungen über :hover

Um einen Zustand eines Elements zu verändern, benötigt man einen Auslöser. Wir schauen uns verschiedene Möglichkeiten an, hier als erstes die Möglichkeit, einen Zustand über die Pseudoklasse **:hover** zu verändern.

Eine CSS-Pseudoklasse ist ein Schlüsselwort, das zu einem Selektor hinzugefügt wird und einen speziellen Zustand des/der ausgewählten Elements/Elemente angibt. Zum Beispiel kann :hover verwendet werden, um die Farbe einer Schaltfläche zu ändern, wenn der Mauszeiger des Benutzers darüber schwebt.

Nicht alle Pseudoklassen eignen sich für das Auslösen von Animationen. Neben :hover kann zum Beispiel :focus benutzt werden. Die CSS-Pseudoklasse **:focus** repräsentiert ein Element (z. B. eine Formulareingabe), das den Fokus erhalten hat. Sie wird im Allgemeinen ausgelöst, wenn der Benutzer auf ein Element klickt oder tippt oder es mit der Tabulatortaste der Tastatur auswählt.

Übersicht über alle Pseudoklassen:

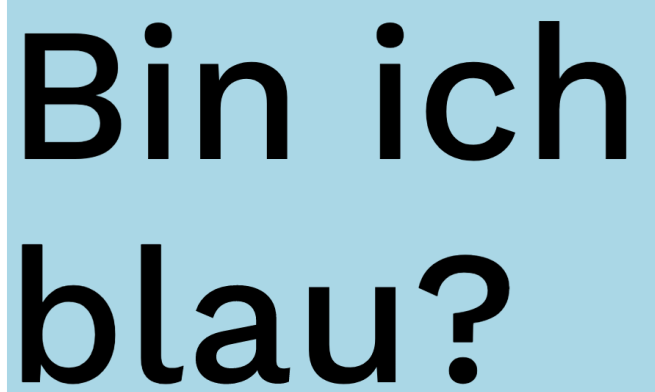
<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

PSEUDOKLASSE :hover

Beispiel

```
.box div { //Defaultzustand  
  width: 500px;  
  height: 500px;  
  background-color:lightblue;  
  color:black;  
}
```

```
.box div:hover{ //Veränderter Zustand  
  background-color:black;  
  color:white;  
}
```



Bin ich
blau?



Bin ich
blau?

Übergangseffekte

Bin ich
blau?

Bin ich
blau?

Bin ich
blau?

Bin ich
blau?

Bin ich
blau?

Transition

transition-property

```
/* Beispiele values */  
transition-property: none;  
transition-property: all;  
transition-property: font-size;
```

Mit none wird keine der Eigenschaften per Übergang verändert. Folge, harter Wechsel von einem Zustand in einen anderen.

Mit all werden alle Eigenschaften per Übergangseffekt verändert.

Mit der Angabe einer spezifischen Property (Bsp. font-size) wird eine Änderung dieser Angabe mit einem Übergangseffekt verändert. Alle anderen Properties würden sich ohne Übergang verändern.

Die CSS-Eigenschaft transition-property legt die CSS-Eigenschaften fest, auf die ein Übergangseffekt angewendet werden soll.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition-property>

Transition

transition-duration

```
/* Beispiele values */  
transition-duration: 1s;  
transition-duration: 500ms;
```

Die Zeitangabe kann in s für Sekunden oder ms für Millisekunden gemacht werden.

Die CSS-Eigenschaft "transition-duration" legt fest, wie lange eine Übergangsanimation dauern soll, bis sie abgeschlossen ist. Standardmäßig ist der Wert 0s, was bedeutet, dass keine Animation stattfindet.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition-duration>

Transition: <https://cubic-bezier.com>

transition-timing-function

```
/* Beispiele values */  
transition-timing-function: linear;  
transition-timing-function: ease;  
transition-timing-function: ease-in;  
transition-timing-function: ease-out;  
transition-timing-function: ease-in-out;  
transition-timing-function: steps(6, end);  
transition-timing-function: cubic-bezier(0.1, 0.7, 1.0, 0.1);
```

Die CSS-Eigenschaft transition-timing-function legt fest, wie Zwischenwerte für CSS-Eigenschaften berechnet werden, die von einem Übergangseffekt betroffen sind.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition-timing-function>

Transition

transition-delay

```
/* Beispiele values */  
transition-delay: 1s;  
transition-delay: 500ms;
```

Die Zeitangabe kann in s für Sekunden oder ms für Millisekunden gemacht werden.

Die CSS-Eigenschaft transition-delay gibt die Dauer an, die gewartet wird, bevor der Übergangseffekt einer Eigenschaft gestartet wird, wenn sich ihr Wert ändert.

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition-delay>

Transition

transition

```
/* Beispiele values */  
transition: all 0.5s ease-out;  
transition: margin-right 4s ease-in-out;
```

Die CSS-Eigenschaft "transition" ist eine Kurzform für "transition-property", "transition-duration", "transition-timing-function" und "transition-delay".

<https://developer.mozilla.org/en-US/docs/Web/CSS/transition>

Transition

transition

```
/* Beispiel */
```

```
.box div {  
  background-color:lightblue;  
  transition-property: all;  
  transition-duration: 3s;  
  transition-timing-function: linear;  
}
```

```
.box div:hover{  
  background-color:black;  
  color:white;  
}
```

Die Transition Properties müssen im Default State des Elements definiert werden, nicht im veränderten State!

Aufgabe 2

transform und transition

Todo

Arbeite durch die Exercises: [https://www.w3schools.com/css/exercise.asp?](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_transitions1)

[filename=exercise_css3_transitions1](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_transitions1)

Dann: Probiere einzelne Worte aus deinem Song zu transformieren und zu animieren. Spiel mit den transform Eigenschaften – aber du kannst auch alles andere ändern: Farbe, Position, Sichtbarkeit, border, margins, paddings, sogar font-family etc. Hab Spass und probiere verschiedenes aus, dann versuche mit der Bedeutung zu arbeiten und diese durch die Bewegung zu unterstützen. Führe Übergangseffekte ein, indem du die Properties von Transition benutzt. Benutze alle Möglichkeiten, die duration, den delay, die timing Funktion.

https://domiwilli.github.io/sfgz_gridandmodules/Tag3/Satz/satz.html

https://cwaeny.github.io/sfgz_gridsandmodules/day3/uebung2/index.html

https://phattrick.github.io/sfgz_gridandmodules/Animation_uebung-Teil-2_Vers2.html

Selbststudium

Todo

Wählt aus den folgenden drei Aufgaben/Vorschlägen zwei aus und bearbeitet diese.

Zwischenpräsentation bei Nick am 4.5.2021

Aufgabe 1

3 Moodboards

Todo

Mache drei Moodboards zum Song. Zum Beispiel:

Mache eines, das die Stimmung deines Songs erfasst.

Wie setzt du die Tonalität visuell um? Was heisst zum Beispiel «Slow down, go down» oder «Imagine» visuell?

Wie ist der Rhythmus, verändert er sich, wie kannst du den darstellen?

Wie stellst du dir den Raum/die Landschaft/Objekte/Formen zum Song vor?

Beachte

- «What you leave out, is as important what you choose» Wähle ganz bewusst mit was du eine Tonalität darstellst
- Suche das Ungewöhnliche, das «überraschend Andere»
- «Text it up» Isolierte Wörter / Sätze unterstützen das Verständnis der visuellen Sprache
- Skizzenhafte Darstellung, damit es Raum für das Wachsen der Idee gibt
- Hab Spass

Aufgabe 2

Auszug aus eurem Song

Todo

Entwurf, basierend auf einem 12-er Raster. Den ganzen Song, oder einen Auszug davon typographisch setzen. Wählt bewusst Grössen, Rhythmus und Weissraum. Gestaltet mindestens drei Sektionen.

Cause every night I lie in bed, The brightest colors fill my head, A million dreams are keeping me awake, I think of what the world could be, A vision of the one I see, A million dreams is all it's gonna take, A million dreams for the world we're gonna make

I close my eyes and I can see
That I call my own The world that's waiting up for me
Through the through the door
dark, But it feels like home
Through a where no one's been before

_____ das wort _____ nöd wär,
wär vilich alles andersch...

Möglicherwis gsechti alles besser us

Ich hätt ä frau ich hätt es auto und ich hätt es huus

Kei chrieg und Gwalt dä fernseh numä netti news

Weniger schlächtä hiphop humä fetti crews

Für neugier gäbs ä wunderpille zum dä hunger stillä

Und ich het ändlich es paar tåg zit zum umächillä

All wäred so druf wie einer flog über das kuckucks

Weniger koks, korupti cops, weniger luxus

weniger überfluss wo zu überdruss fühert

Alti lüt wäred nöd einsam und weniger fruschtriert

Und s wort _____ gäbs als antwort, nie es aber

Und mini reims wäred für frau wie latin lover gelaber

Models wäred fett, akne im gsicht wär im trend

D uhrä inexischtänt und ich hätt no nie verpennt

Als stichwort nur imagine, dr john wär no nöd tot

Hollywood wär abäbrännt, beverly hills in wohnigsnot

Sowieso wär i könig i mim eignä riich

Und trotz wortgwantheit blibt alles fiktiv

_____ das wort _____ nöd wär,
wär vilich alles andersch...

_____ das wort _____ nöd wär, wär vilich alles andersch

D chind würdet d wält regierä während dem d aliens länded

Wohlverstandä gäldproblem währet absofort verbi

Will ich hätt ihn, dr sächser i dr landesloterie

Dr ganzi jackpot für mich

_____ s wort _____ nöd gäb

Aber säb isch s'handycap

THE START
YOUR HEART
THE DAY YOU
WERE BORN
THIS IS THE SUN
THESE ARE YOUR LUNGS
THIS IS THE DAY YOU
WERE BORN

AND I AM
YS ALWAYS ALWAYS ALWAYS AL
YOURS

THESE ARE THE SCARS
DEEP IN YOUR HEART
THIS IS THE PLACE
YOU WERE BORN

THIS IS THE HOLE
WHERE MOST OF YOUR SOUL
COMES RIPPING OUT
FROM THE PLACES YOU'VE
BEEN TORN

AND I AM
YS ALWAYS ALWAYS ALWAYS AL
YOURS

HALLELUJAH
I'M CAVING IN

Let it go, let it go
Can't hold it back anymore

LET it

go

LET it

Aufgabe 2

Umsetzung HTML und CSS

Todo

Probiere mal, einen Teil aus deinem Song Entwurf in HTML und CSS umzusetzen.

(Vielleicht musst du vereinfachen oder du baust einfach mal das grobe Gerüst)

Aufgabe 3

Robot Page

Todo

Gestalte eine erste Version deines Robots/Android-Wesens in HTML und CSS.
Benutze Transforms, Positioning, allenfalls auch bereits Transitions bei hover.

Beachte

- Suche die Verbindung zum Song und zur Stimmung im Song
- Probiere, die technischen Konzepte als Gestalter zu verwenden
- Hab Spass