it's good to have you here.

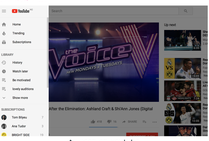
let's get straight to business ...

1

LIGHTBOXES & DIMMED BACKGROUNDS

It is very common to dim everything behind an element to focus on a certain UI element

EXAMPLE:



As seen on youtube.

The Trick: Use CSS Positioning!

The backdrop for dimming

```
.dimmed {
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  position: fixed;
  background: rgba(0,0,0,0.4);
}
```

- fill the entire screen, from top to bottom
- take the element out of the document flow, unaffected by scrolling
- use a dark transparent background

the element to grab attention

```
.lightbox {
  position: absolute;
  z-index: 1;
}
```

- take the element out of the document flow, and put it on top.

LIVE CODE:
https://codepen.io/ohansemmanuel/full/BmXaEO/

2

CUSTOM UNDERLINES

You can't always get all that you need with the text-underline property.

You may get away with using the border-bottom property, but it is hardly the perfect solution.

EXAMPLE:

Hello World

Hello World, Again

Hello World, Again!

Hello World, Even More!

The Trick: Use the Background Property!

The element with the underline

```
.underline {
  background: linear-gradient(
              90deg, blue, red)
              0 2em/100% 2px
              no-repeat;
}
```

- use the background shorthand for brevity.
- define your colors. You don't have to use Blue or Red. If you want a black underline, use Black!
- don't repeat the background. You can create a dashed underline effect by repeating the background though

0 2em/100% 2px

1. This portion of the property declaration is very important.
2. The shorthand sets both the background position and size values. i.e background position / background size.

0 2em/100% 2px

background-position — background-size

1. Each pair of values represent x and y measured from left to right, top to bottom
2. Be sure to play with these values to see how they affect the overall design.

LIVE CODE:
https://codepen.io/ohansemmanuel/full/mpqryz/

3

MORE CLICKABLE AREA

No user wants to click a dozen times to hit that little button of yours.

In cases where you don't want to increase the size of the button, there's a trick for that

EXAMPLE:

The Trick: Use Pseudo Elements!

A Pseudo element will capture mouse interactions for its parent

```
button:before {
  content: '';
  position: absolute;
  top: -35px;
  left: -35px;
  width: 140px; height: 140px;
}
```

- You need this, you know!
- You must have set position: relative on the button itself.
- position the pseudo element
- the width and height that represent the expanded clickable area

1. As seen in the example above, the area represented by the white border refers the the area of the pseudo element.
2. Clicking anywhere within this area, will also mean clicking the button.

LIVE CODE:
https://ohansemmanuel/full/gpVjBO/

check the console when you click :)

4

MORE CURSOR TYPES. USE 'EM

Don't just use cursor: pointer.

You can make your web app more usable by using the right cursor at the right time.

EXAMPLE:

The Trick: Use the helpful one-liners!

```
button {
  cursor: copy;
}
```

- change the value here

TRY THESE

no-drop | not-allowed | cell | vertical-text | alias | zoom-in | zoom-out | row-resize | col-resize

LIVE CODE:
https://ohansemmanuel/full/PEOOrx/

5

ACCESSIBLE WAY TO HIDE CONTENT

Sometimes you want to hide an element BUT still have it accessible to screen readers.

You can't achieve this with just display: none

The Trick: Use the Clip property

NB: there are other ways to achieve the same effect, but I use this technique.

```
.sr-only {
  position: absolute;
  clip: rect(1px 1px 1px 1px);
  clip: rect(1px, 1px, 1px, 1px);
}
```

- the clip property ONLY works on absolute or fixed positioned elements
- IE 6 to 7
- IE 8+ and other browsers

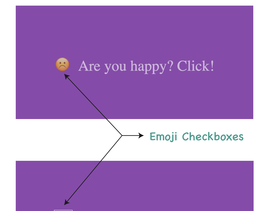
LIVE CODE:
https://codepen.io/ohansemmanuel/full/dJZJdq/

6

STYLING CHECKBOXES

Styling a checkbox can be a pain, but don't worry there's a trick for that.

EXAMPLE:



The Trick: Hide the Default Checkbox & use the label Pseudo Element!

NB:
Incase you didn't know, clicking the label will also toggle the input 'checked' property.

Assuming you've rightly set the label's 'for' attribute

The checkbox to be styled

```
input[type="checkbox"] {
    position: absolute;
    clip: rect(1px,1px,1px,1px);
}

input[type="checkbox"] + label::before {
    content: '\2639';
    display: inline-block;
    width: 1.2em;
    height: 1.2em;
    margin-right: .5em;
    border-radius: .1em;
    text-indent: .15em;
    line-height: 1.4;
}
```

hide the default checkbox in an ACCESSIBLE manner.

the Emoji unicode character

use the adjacent selector

target the label ::before pseudo element

LIVE CODE:
https://codepen.io/ohansemmanuel/full/xpPaKd/

7

TRANSPARENT BORDERS

This one is subtle. Let me explain.

Assume you have an element with a background color & a border.

You want the border to be transparent so you can show off the background-image on 'body'

How do you do that? Simple, huh?

The element with the border

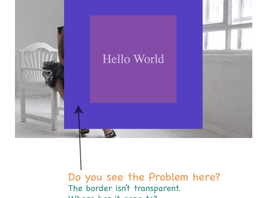
```
body {
  background:
    url('http://bit.ly/2xMHeVk')
    0 0/cover;);
}

section {
  border: 100px solid rgba(0,0,255,0.36);
  background: rgba(142,68,173 ,1);
}
```

the Background image on'body'

the transparent border

Don't forget the background

The Result



Do you see the Problem here?
The border isn't transparent.
Where has it gone to?

The Trick: Use Background Clip!

By default, backgrounds will extend underneath the borders.'

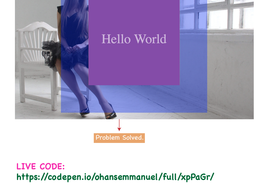
This explains why we can't see the image

Underneath the transparent border is the background color NOT the image.

You must instruct the background to NOT slide under the border

```
section {
  border: 100px solid rgba(0,0,255,0.36);
  background-clip: padding-box;
}
```

Hey Background, don't slide under the border area.
Only cover the padding area of the element



Problem Solved

LIVE CODE:
https://codepen.io/ohansemmanuel/full/xpPaGr/

I don't understand ...

If some CSS properties I've used here are alien to you, look them up on codrops CSS reference.

Just search and read up. Don't worry, you'll get the hang of it.

Codrops css reference: https://tympanus.net/codrops/css_reference/

HOORAY!!

Glad you made it to the end!

1. Now that know these stuff, actually practice them!
2. Got questions? Hit me up on Twitter :) - @ohansemmanuel
3. Aye! Go have fun with CSS!

much love,
Ohans Emmanuel