**SASS ( Syntactically Awesome StyleSheets )**

Sass 2 uzantili sekilde kullanilabiliyor. .scss .sass.

.scss daha sik tercih ediliyor

Sass tarayicilar direk okuyamiyor onun icin node-sass kurmamiz sart

**SASS in Bize Sagladiklari**

\*degiskenler tanimlayabilir ve onlari cagirip kullanabiliyoruz

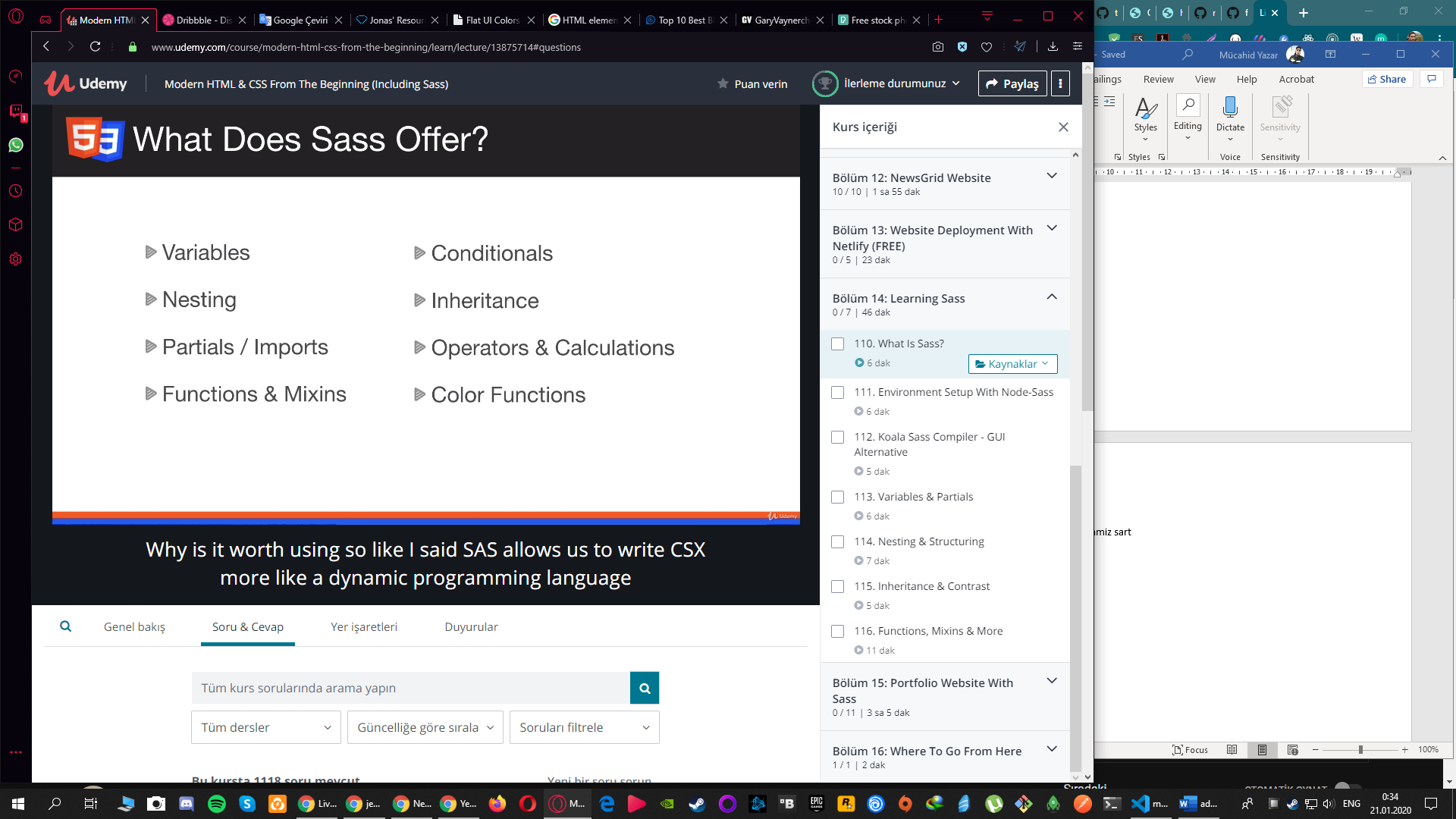
\*baska css dosyalarini iceri import edip kullanabiliyoruz

\*fonksiyon seklinde kullanimlar yapabiliyoruz

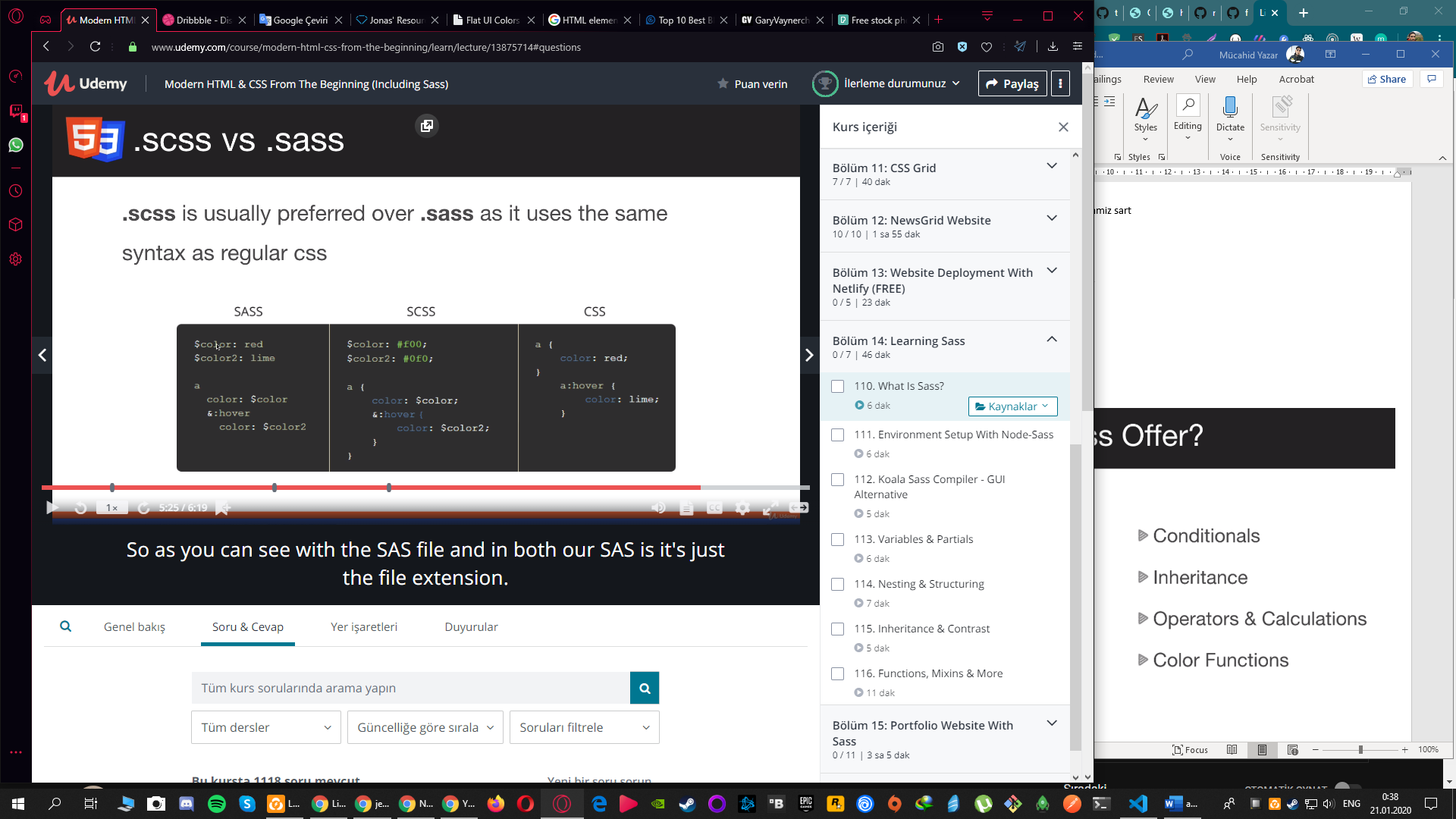
\*if else kosullari koyabiliyoruz

\*Kendi css bloklarimizi olusturabiliyoruz

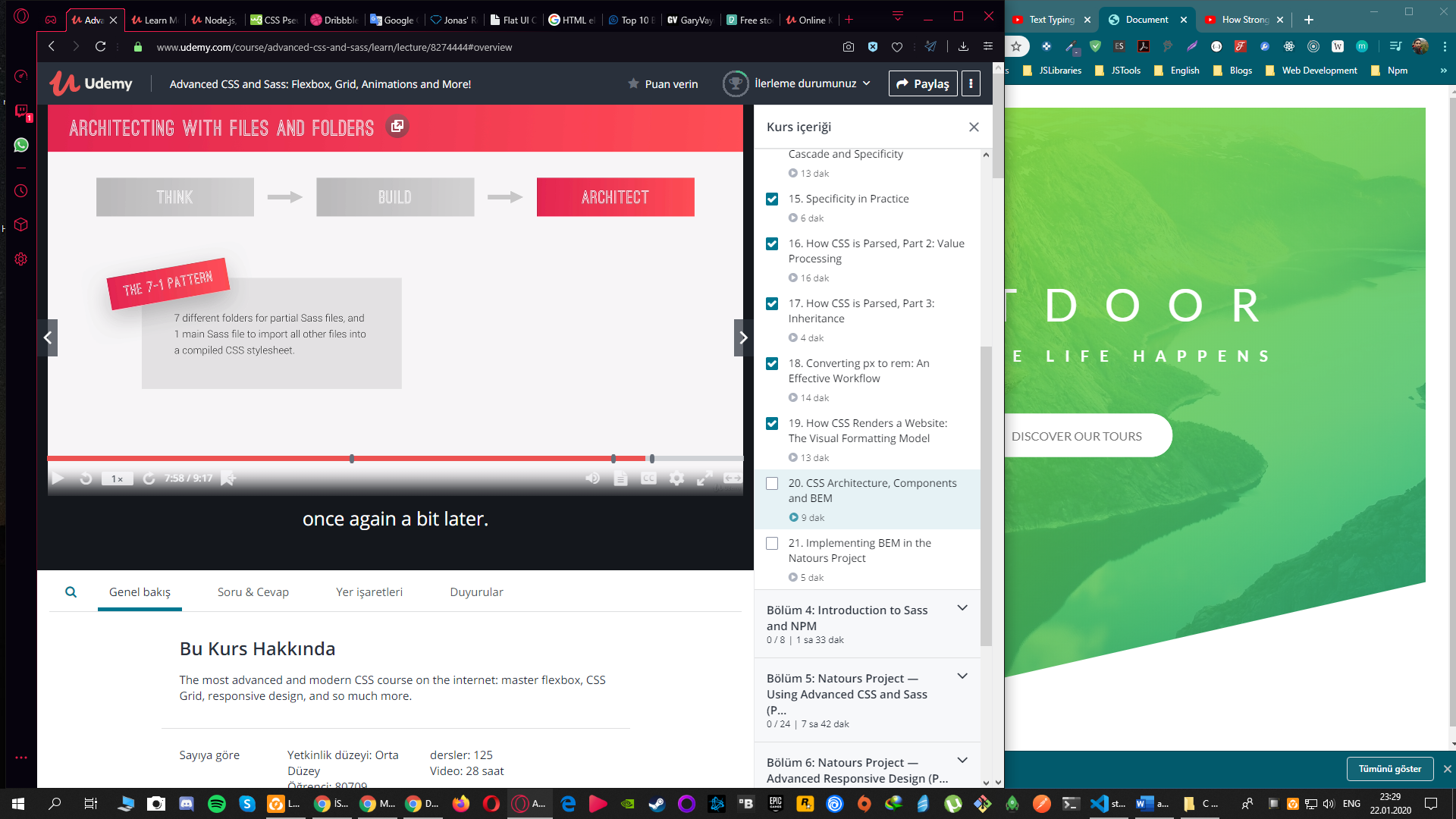
\*renk fonksiyonlari tanimlayabiliyoruz

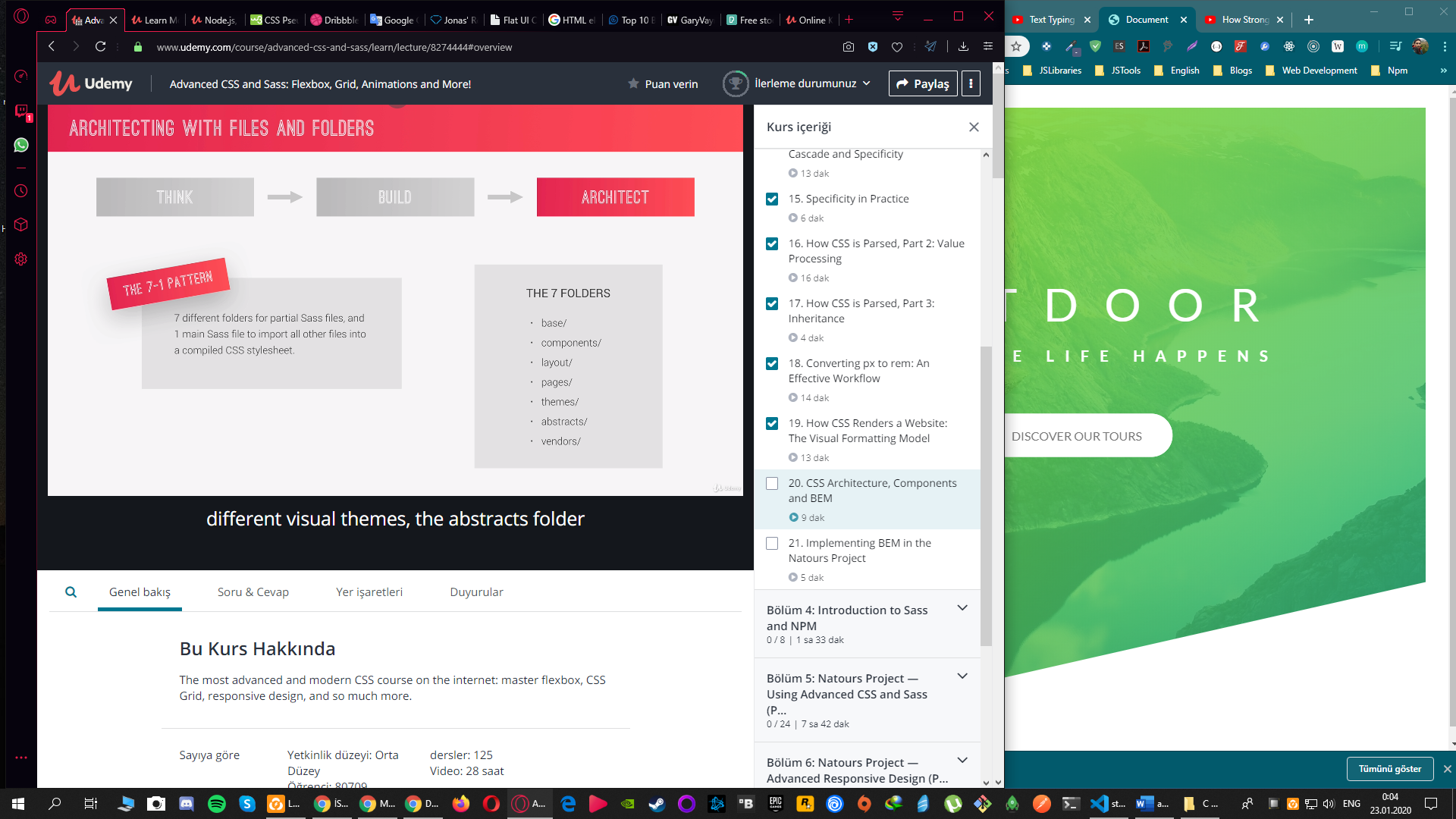


**SASS vs SCSS vs CSS**



SASS KULLANIMI





SASS Kurulum

1. Npm init y
2. Npm i node-sass
3. Package.json => scripts => “sass”: “node-sass -w scss/ -o dist/css/ --recursive”

Yukarida scss/ ile scss dosyalarimizin bulundugu konumu belirliyoruz

Dist/css/ ilede onlari css donusturdukten sonra birakacagi yeri belirliyoruz. Biz burada ki css i html sayfamiza ekleyecegiz

-w watch demek

--recursive de sanirim otomatik olarak her degisikliktenb sonra bize gerek kalmadan kendisi degisiklikleri css e gonderiyor

Ve npm run sass ile baslatiyoruz.

SASS OZELLIKLERI

* SASS ici degisken tanimlayarak kullanim

$color: blue;

$priamry-color: steelblue;

$secondary-color: skyblue;

$light-color: #f4f4f4;

$dark-color: #333;

$font-stack: Arial, Helvetica, sans-serif;

body {

    background: $light-color;

    color: $dark-color;

    font-family: $font-stack;

}

* Yada \_variables diye bir degisken olusturup,

$ ile tanimlanan degiskenleri oraya alip,

Daha sonra @import variable yazip

Degiskenlerimizi baska dosyada tutup import ederekde calsiabiliriz.

* Eger \_ yi variables a koymazsak node-sass bunu css klasorunun icine variable.css olarak compiled ederdi yani cikartirdi.
* Nesting yapisi ile de ic ice kodlar yazabiliriz asagidaki gibi
* Burada header icinde h1 i aliyoruz yani seciyoruz sadece.

header {

    background: $dark-color;

    color: $light-color;

    padding: 1rem;

    h1 { //Bu bır NESTED SELECTORS dur

        text-align: center;

    }

}

**& ile EKLEME YAPMA**

* Asagida ise .section a padding ve h3unu bazi degisiklikler yapmisiz. Onun altinda ise &-a yaparak .section-a yi seciyoruz. Cunku & sonra yazilan seyler &’nin scss icindeki parent elementine eklenir yani alt ornekte .section’a eklenir. Ve secim yapilir.

.section {

    padding: 3rem;

    h3 {

        font-size: 2rem;

    }

    &-a { // => .section-a

        background: $primary-color;

        color: #fff;

    }

}

* Yada bu sekilde ekleme yapariz ve hover i seceriz

a {

    color: #333;

    &:hover {

        color: coral;

    }

}

**% ile TOPLU OZELLIK TANIMLAMA @EXTEND ILE IMPORT ETMEK**

* Yada asagidaki gibi basina % isareti koyarak %btn-shared ile bir degisken tanimlariz ve daha sonra bu degiskenin ayni ozelliklere sahip olmasini istediklerimize @extend %btn-shared yaparak ozelliklerini aktaririz. Asagidaki gibi

%btn-shared {

    display: inline-block;

    padding: 0.7rem 2rem;

    border: none;

    cursor: pointer;

    text-decoration: none;

    margin-top: 1rem;

}

.btn {

    &-light {

        @extend %btn-shared;

        background: $light-color;

        color: #333;

    }

    &-dark {

        @extend %btn-shared;

        background: $dark-color;

        color: #fff;

    }

}

**BAGLANTILI ILISKILI ELEMENT LERDA EXTENDS, BAGLANTILI ILISKILI OLMAYAN ELEMENT LERDE MIXIN KULLAN**

**FUNCTIONS**

* \_functions.scss adinda bir dosya olusturalim.

Ve icine main.scss de kullanmak icin asagidaki fonksiyonu kuralim

* @fonksiyon tanimlamamiza yardimci olur.
* Lightness bir javascript komutudur aslinda rengin tonunun ne kadar acik oldugunu belirliyor ve 0ile100 arasinda bir deger donuyor bize.
* Yani burada demisiz ki set-text-color adinda bir fonksiyon olusturmusuz. Ve bunun icinde gelen degeri kontrol et demisiz. Eger gelen renk 50 den daha buyukse KOYU bir renk donecek, daha kucukse acik bir renk donecek.

// Set Text Color

@function set-text-color($color) {

    @if(lightness($color) > 50) {

        @return #000;

    } @else {

        @return #fff;

    }

}

* Kullanma isinide asagidaki gibi yapiyoruz. Kirmizi ile yazdim.

header {

    background: $dark-color;

    color: set-text-color($dark-color);

    padding: 1rem;

    h1 {

        text-align: center;

    }

}

**MIXIN**

--webkit-transform: rotate(-20deg);

-ms-transform: rotate(-20deg);

transform: rotate(-20deg);

* Transform tum tarayicilarda olmadigi icin yukaridaki kodlarin hepsini tanimlamamiz gerekiyor kullandigimiz zaman diger tarayicialrdada sorunsuz calismasi icin. Fakat bunun yerine mixini kullanarak onlari baska yerde tanimlar ve main.scss dede kullanabiliriz.

1. Once \_function.scss i olusturup @import ‘functions’ diyerek import ediyoruz main.scss dosyamiza

// Transform mixin

@mixin transform($property) {

    --webkit-transform: $property;

    -ms-transform: $property;

    transform: $property;

}

1. Sonra yukaridaki sekilde mixini cagirip transform adinda degiskenimizin ismini tanimliyoruz ve parantez icinde gelecek tanimlamalarimizda kullanacagimiz property i tutuyoruz. Bu property ‘i main.scss den biz yollayacagiz.
2. Daha sorna main.scss e gelip, ornegin hover olunca mixini kullanalim diyerek asagidaki gibi @include ile mixini cagiriyorum. @include dedikten sonra tanimladigimiz mixin adini yaziyoruz oda transform. Ve bu degiskene gonderecegimiz property i : koyduktan sonra rotate(20deg) gibi asagidaki gibi yaziyoruz.

&:hover {

            @include transform(rotate(20deg));

            // aslinda yukarida olan asagidakinin aynisi

            // transform: rotate(20deg);

        }

**DARKEN and LIGHTEN**

Bu sekildede fonksiyonlu koyulastirma vveya renk acma islemleri yapabiliriz. Amountta yazan miktar % miktaridir.

            background-color: darken($color: $light-color, $amount: 10);

            background-color: lighten($color: $dark-color, $amount: 5);

backface-visibility: hidden;

//Animasyon bittikten sonra konum bozulmasini duzeltiyor

Tiklama efekti

Hover oldugumuzda 3px yukari gider buton

Tikladigimizda ise oldugu yerden 1 px asagi iner

ve tiklama efekti olusur

&:hover {

        transform: translateY(-3px);

    }

    &:active { //Tiklandiginda

        transform: translateY(-1px);

    }

Bununlada disariya acilan ayri bir beyaz golge veririz en son acildiginda opacity 0 olur hoverafter sayesinde

    &::after {

        content: '';

        display: inline-block;

        height: 100%;

        width: 100%;

        border-radius: 100px;

        position: absolute;

        top: 0;

        left: 0;

        z-index: -1;

        transition: all .4s;

    }

    &:hover::after {

        transform: scaleX(1.4) scaleY(1.8);

        opacity: 0;

    }