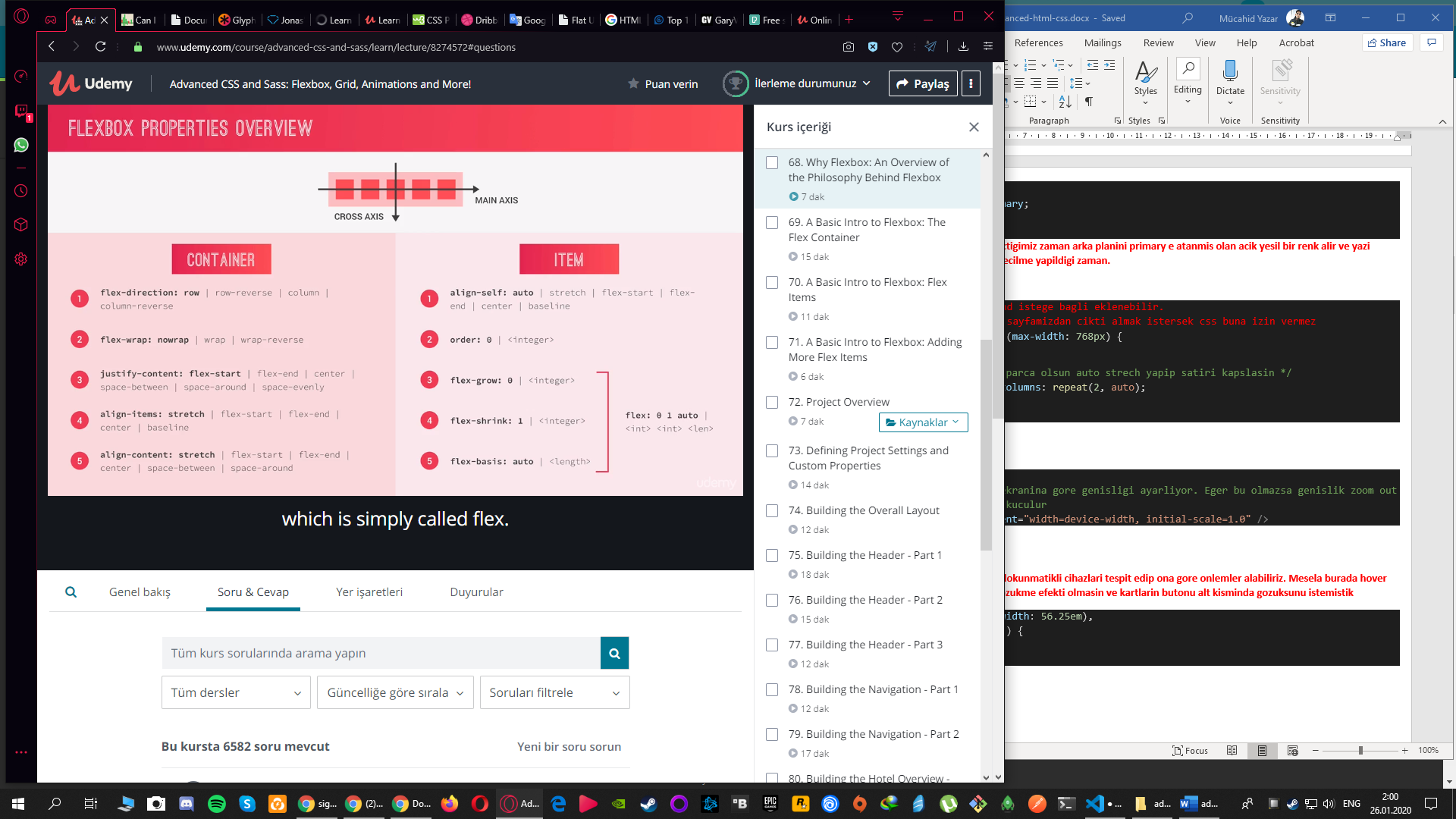
**FLEXBOX**



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Flex Align</title>

    <style>

        #container {

            background: #555;

            height: 600px;

            display: flex;

        }

        .item {

            background: #f4f4f4;

            border: #ccc solid 1px;

            text-align: center;

            padding: 1rem;

            margin: 0.5rem;

        }

    </style>

</head>

<body>

    <div id="container">

        <div class="item item-1"><h3>Item 1</h3></div>

        <div class="item item-2"><h3>Item 2</h3></div>

        <div class="item item-3"><h3>Item 3</h3></div>

    </div>

</body>

</html>

**DISPLAY: FLEX**

//PARENT ELEMENTE UYGULANIR

Display: flex => Tum ıtemler yan yana aynı satırda hizalanır

**###FLEX-DIRECTION**

//ITEM’E UYGULANIR (DISPLAY FLEX iken uygulanir)

Flex-direction: row => Default olarak olandir. Itemler soldan saga hizalanir. Sola yapisik olur ilk bastaki.

Flex-direction: row-reverse => Tum itemleri main axis de en sagdan 1. 2. 3. 4. … diye siralar. Yani sagdan sola hizalama yapar.

Flex-direction: column => Tum itemlari sutunda siralama yapar yani Cross Axisde sirasiyla asagiya dogru siralama yapar. En ust sutunda 1 sonra 2.3.4.5. diye devam eder. Column u kullaninda display flex ile gelen ayni satirda kalma zorunlulugu gider

Flex-direction: column-reverse => Tum itemleri sutunda asagidan yukariya dogru en altta ilk item olacak sekilde cross Axisde sirasiyla siralama yapar. Yani column in tam tersidir.

**###NOT: COK ONEMLI**

**JUSTIFIY-CONTENT ve ALIGN-ITEMS ile hizalama yaparken FLEX-DIRECTION cok onemlidir.**

**Default olarak FLEX-DIRECTION ROW ‘dur.**

**ROW ike MAIN-AXIS soldan saga gider CROSS-AXIS yukaridan asagiya.**

**COLUMN iken MAIN-AXIS yukaridan asagiya gider CROSS-AXIS soldan saga gider.**

**###JUSTIFIY-CONTENT =>** Justify-content = main axis boyunca siralamamiza yarar

justify-content: space-evenly;

justify-content: flex-start  => default olarak gelen degerdir ve tum itemlari solda siralar \*/

justify-content: flex-end  => tum itemleri ayni satirda sagda hizalar. Item1 - Item 2 - Item3 olarak \*/

justify-content: center  => tum itemlari ortada hizalar \*/

justify-content: space-arround  => tum bosluklar itemlarin etrafinda esit olarak dagitilir. Evently ile aralarindaki fark ise bosluklar tum itemlere ayri ayri esit dagitildigindan item 1 ve2 arasinda 2 ve 3 arasinda daha fazla bosluk olur. \*/

justify-content: space-between => tum bosluklaritem aralarinda hizalanir. Ilk bastaki itemin solunda ve sonraki itemin saginda bosluk olmaz. \*/

justify-content: space-evenly  => itemlerin tum taraflarina bosluklar esit dagitilir. \*/

**###ALIGN-ITEMS =>** cross axis boyunca siralamamiza yarar

align-items: stretch;

alighn-items: stretch  => itemlarin hepsi containerin uzunlugu kadar asagi boyu uzatilir. Default olarak gelen degerdir.

alighn-items: flex-start  => containerin en ust hizasinda tutar cisimleri ve itemlarin parent elemente gore boyutlarinin uzamasini engeller

alighn-items: flex-end  => Containerin en alt hizasinda tutar cisimleri ve uzamasini engeller

alighn-items: center  => Itemlari diker olarak containerin ortasinda konumlandirir ve kendi boyutlarinda tutar ve stretch uzatmasini yine onler

alighn-items: baseline  => cisimlerin iceriklerini alt kisimlarindan ayni hizada hizalama yapar.

**###ALIGN-CONTENT =>** Cross Axisde extra alan oldugunda hizalamamiza yarar

align-content: center;

SENARYO => container 600px ve bizim flew-wrap ile 2 satira dusurulmus 100px uzunlugunda 12 tane kutumuz var diyelim. 6 tanesi bir 6 tanesi bir. Fakat normalde 1. 6 kutuyu en ustten siralar, 2. 6 kutuyu 300. px yukseklikten siralar

align-content: flex-start => yaparak tum satirlari yani 2 6 kutulu satiri alt alta aralarinda bosluk olmadan siralariz.

align-content: flex-end => yaparak tum satirlari en alta dayali hizalariz.

align-content: center => tum satirlari dip dibe aralarinda bosluk olmadan ortada hizalariz.

align-content: space-around => 2 kutu grubu arasinda her birine bosluklari esit olarak dagitir.

align-content: space-between => default olarak gelen degerdir tum bosluk input gruplari arasina konulur.

align-content: space-evenly => item gruplarinin her bir grubuna dikey olarak aralarina esit bosluk konulur.

**###ALIGN-SELF =>** ITEM lara ozel olarak verilir ve Tek bir itemi dikeyde yani cross-axisde hizalamak icin kullanilir

        .item-2 {

            align-self: flex-start;

        }

Align-self: flex-start  => dikeyde en basta konumlandirilir.

Align-self: flex-end  => dikeyde en dipte konumlandirilir.

Align-self: center  => dikeyde ortada konumlandirir

Align-self: stretch  => secilen itemi yayar yani uzatir cross axis boyunca

Align-self: baseline  => asil konumunda konumlandirir.

**###ORDER =>** Order ilede itemlari istedigimiz sira ile konumlandirilmalarini saglayabiliriz.

        .item-1 {

            order: 3;

        }

        .item-2 {

            order: 1;

        }

        .item-3 {

            order: 2;

        }

**###FLEX-BASIS and FLEX-GROW & FLEX**

#FLEX-GROW

Eger genel olarak item icinde 1 olarak kullanilirsa tum itemlar alabildikleri kadar genislik alirlar.

Margin ve paddingi ezemezler. Margin ve paddinge uymak zorundadir.

Fakat asagidaki yani item-3 deki gibi ozel olarak item icinde kullanilirsa flex-grow:2 ise diger itemlerin 2 kati 3 ise diger itemlerin 3 kati daha fazla buyuk olur. Veya item-3 icinde kullanildiginda flex-grow 1 deseydik, item-3 uzayabildigi kadar uzardi main axis de.

#FLEX

FLEX-GROW ile AYNI islevlere sahiptir. Flex: 1 uzayabildigin kadar uza demektir. Tum elemanlara verilrise herkes esit uzar.

        .item {

            flex-grow: 1;

flex: 1;

        }

        .item-3 {

            flex-grow: 1;

flex: 1;

        }

#FLEX-BASIS

width verir. Width ile ayni islevi gorur

        .item-3 {

            flex-basis: 200px;

            flex-basis: 20%;

        }

#FLEX-SHRINK

Itemlarin buzulmesini yani kisalmasini acmaya veya kapatmaya yarar. 0 kapatir 1 acar.

        .item-3 {

            flex-shrink: 0;

        }

###FLEX or FLEX-GROW / FLEX-SHRINK / FLEX-BASIS

        .item {

flex: 0 0 300px;

        }

###FLEX-WRAP => Itemleri WRAP lemeye yarar

flex-wrap: no-wrap;

Default olarak gelen deger no-wrap dir.Bu deger olursa itemler CONTAINER ‘a sigdirilmaya calisilir.

Wrap olursa itemler container’a sigdirilmaya calisilmaz ve alt satira alinir.

            /\* Bu sekilde satirda cok fazla yani 12 den fazla eleman oldugunda scrool bar cikmasini onleriz ve scrool bar cikartmaz bize fazla elemanlari alt satira gonderir. \*/

            /\* Alt satira tarayici ekran boyuttuna gore responsive olarak elemanalri gonderir. Yani ilk 2 gonderir 10 kalir biraz daha kuculur 2 taha gonderir 8 kalir ilk satida gibi... \*/

            /\* Bu ise flex-wrap: wrap; ile birebir ayni islemi yapar \*/

            /\* flex-flow: row wrap; \*/

            /\* Hersey tek satirda kalmaya devam eder. Birnevi flex-wrap: wrap; yazilmadigini dusunun. \*/

            /\* flex-flow: row nowrap; \*/

