**Ders Öncesi Hazırlıklar:**

1. Atom Text Editor Kurulumu: https://atom.io/
2. Github Üyeliği: https://github.com

**Bilgisayar Bilimine Giriş**

Kara kutu örneği: Bilgisayar bilimi (Computer Science) girdi ile çıktı arasındaki işlemi gerçekleştiren bir karakutuya benzer. Algoritma adı verilen bu işlem veriyi değiştirme, depolama ve dijital bilginin iletimi işlerinden sorumludur.

output

input

Fihrist ile isim bulma örneği: Algoritma nasıl çalışıyor?

**Bit:** En küçük depolama birimi. Yanızca 0 veya 1 depolayabilir.

**Byte:** 8 bit’in bir araya gelmesinden oluşur. (Ör: 1 0 0 1 0 1 1 0) Bir byte bir karakter depolayabilir. (Ör: ‘A’ veya ‘#’)

|  |  |  |
| --- | --- | --- |
| **Bit Sayısı** | **Pattern** | **Pattern Sayısı** |
| 1 | 0 1 | 2 |
| 2 | 00 01 10 11 | 4 |
| 3 | 000 001 011 111 100 110  010 101 | 8 |
| 4 | … | 16 |
| N | …. | 2n |

**1 Byte:** 8 bitten oluşur ve 0,1,2,3 … 255 değerlerini depolayabilir.

**Byte ve Karakterler**

|  |  |  |
| --- | --- | --- |
| 32 space  33 !  34 "  35 #  36 $  37 %  38 &  39 '  40 (  41 )  42 \*  43 +  44 ,  45 -  46 .  47 /  48 0  49 1  50 2  51 3  52 4  53 5  54 6  55 7  56 8  57 9  58 :  59 ;  60 <  61 =  62 >  63 ?  64 @ | 65 A  66 B  67 C  68 D  69 E  70 F  71 G  72 H  73 I  74 J  75 K  76 L  77 M  78 N  79 O  80 P  81 Q  82 R  83 S  84 T  85 U  86 V  87 W  88 X  89 Y  90 Z  91 [  92 \  93 ]  94 ^  95 \_  96 ` | 97 a  98 b  99 c  100 d  101 e  102 f  103 g  104 h  105 i  106 j  107 k  108 l  109 m  110 n  111 o  112 p  113 q  114 r  115 s  116 t  117 u  118 v  119 w  120 x  121 y  122 z  123 {  124 |  125 }  126 ~ |

**Bilgisayarda Sayılar**

Tam sayılar genellikle 4 veya 8 byte ile depolanırlar. 32 bit/64 bit

**4 byte’ın depolayabileceği tam sayılar:** -2147483648 ile 2147483647 arasında

**8 Byte’ın depolayabileceği tam sayılar:** -9223372036854775808 ile 9223372036854775807 arasında.

2147483647 sayısına 1 eklenirse -2147483647 ‘ye döner. Buna **integer overflow** denir.

**integer overflow örneği için C kodu**

<https://repl.it/languages/c>

#include <stdio.h>

#include <math.h>

int main(void) {

int n = pow(2,31)-1;

printf("number: %i", n);

return 0;

}

Ör: Gangnam Style şarkısının Youtube’u 32 bitten 64 bite geçmeye zorlaması.

Gangnam Style: <https://www.youtube.com/watch?v=9bZkp7q19f0>

**DOM (=Document Object Model)**

DOM**, HTML, XML** gibi belgelerin diğer programlama dilleri veya script dilleriyle iletişim kurabilmesini sağlamak için geliştirilmiş bir arabirimdir.



**HTML**

## **HTML Temelleri**

HTML, web sitelerini oluşturmak için tanımlanan standart bir metin işaretleme dilidir. HTML ile web sitesi oluştururken HTML diline tanımlı etiketleri kullanırız. HTML5 ise bu dilin son sürümüdür. Bazı etiketler güncellenmiş, bazıları kaldırılmış ve bazıları da yeni eklenmiştir. HTML, bir programlama dili olarak tanımlanamaz.



Görselde görmüş olduğunuz gibi **< >** işaretleri arasına yazılanlar bizim etiketlerimiz oluyor ve her biri farklı bir iş yapıyor. Projenizi gerçekleştirmeniz için kullanacağınız etiketleri uygulamalı olarak nasıl kullanıldığını öğreneceksiniz. Ama onlara geçmeden önce HTML dilinin yazım kurallarından bahsedelim.

Öncelikle etiketlerin kullanımıyla ilgili kritik bilgiyi verelim. Bir etiketi açtığınız zaman onu tekrar kapatmanız gerekiyor. Örneğin; **<p>** Paragraf **</p>.** Etiket açılır, içerik eklenir ve kapatılırken etiketin içerisine **/** işareti eklenir. Böylece tarayıcı o etiketin kapatıldığını anlar ve diğer etiketleri okumaya devam eder.

HTML sayfası oluştururken ilk yapacağımız şey belge tipini belirtmektir. En üst satıra **<!DOCTYPE html>** ekleyerek sayfamızın bir HTML sayfası olduğunu tarayıcımıza bildirmiş oluruz. Daha sonra **<html> </html>** etiketlerini açarız. Bütün web sitesi bu etiketin içerisinde yer alır. Daha sonra **<head> </head>** ve **<body> </body**> etiketlerini açarız. **<head>** etiketi tarayıcımız tarafından görüntülenmez sadece oradan alacağı ayarları alır ve sayfaya uygular. **<body>** etiketi ise tüm web sitesinin içeriğinin yazıldığı yerdir. Tüm bunları oluşturduktan sonra HTML web sitesi oluşturmak için temel yapıyı kurmuş oluruz.

**Not:** Eğer web sitenizde Türkçe karakter sorunu yaşıyorsanız **<head>** etiketinin içerisine

**<meta charset=”utf-8” />** satırını eklersek sorun düzelecektir.

Utf-8 nedir? Derseniz. İçerisinde Türkçe karakterleri barındıran standart bir karakter dizinidir.

Bundan sonraki kısımda tek tek kullanacağımız etiketleri anlatacağız.

* **Sayfayı Adlandırmak - <title>**

Tarayıcımızda sayfamızı açtığımız zaman en üstteki sekme alanına web sitemizin adını veya o sayfanın başlığını yazabiliriz. Bunun için oluşturduğumuz **<head>** etiketinin içine **<title>** etiketini yazarak sayfamıza başlık atamış olacağız. Örnek vermek gerekirse benim sayfamın başlığı “**Anasayfa**” olsun. Bunu yapmak için aşağıdaki kodları yazarız.



**Not:** Yazdığımız kodları daha sonra anlamak için yorum satırları kullanırız. Yorum satırları tarayıcının hiçbir işlemine müdahale etmeden sadece sizi yönlendirmek amaçlı çalışırlar. **<!—Yorum -->** şeklinde kullanılır.

## **Başlıklar, Paragraflar, Bağlantılar ve Görseller**

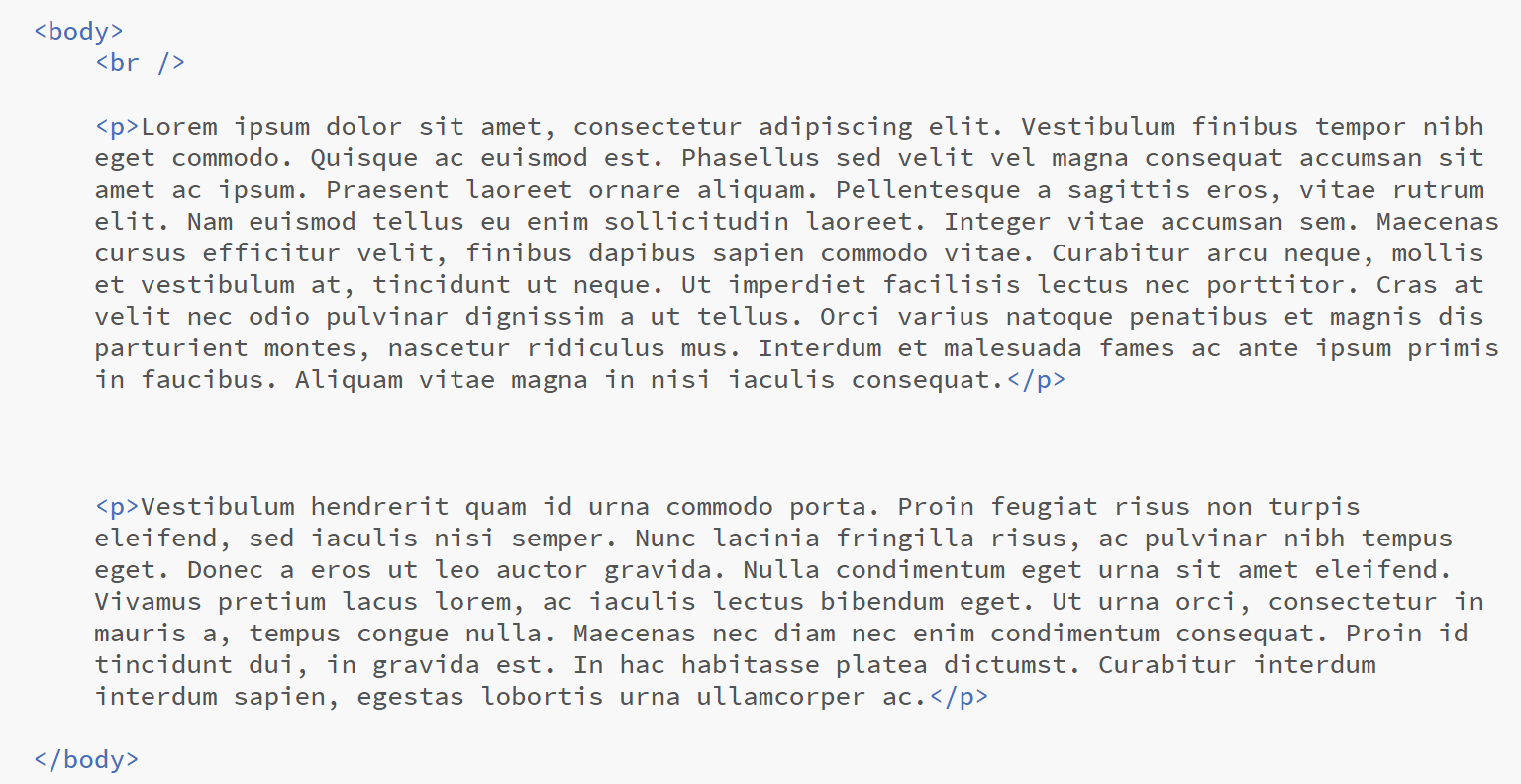
* **Başlıklar - <h1>, <h2>…<h6>**

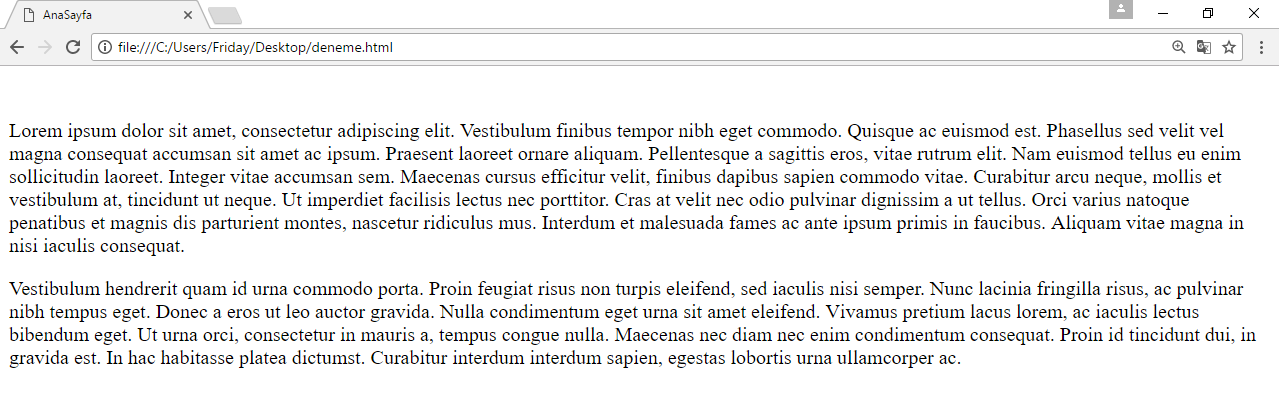
<h1> etiketi arasına yazılanları kendi standardındaki karakter özelliklerini ekleyerek yayımlar. h1,h2,…,h6 arasında başlık değerleri vardır. h1 en büyüğü h6 en küçüğüdür. Kullanımı şu şekildedir;



* **Paragraflar - <p>**

Web sitemizde bir paragraf yayınlamak istersek **<p>** etiketleri arasına paragrafımızı yazarak bunu yapabiliriz. **<p>** etiketi **<body>** etiketinin içerisinde kullanılır. Örneğin;





**Not: <br />** etiketi bir satır boşluk bırakmak için kullanılır.

Aynı zamanda paragrafın içerisinde kullanacağımız özellikler var. Bunlar **kalınlık**, *italik* ve altı çizili sözcükler. Bunları yapmak için yine bazı etiketlerimiz var onların kullanımı şu şekildedir;

* **Kalınlık - <b>**

Etiketler içerisine yazılanları **kalın** olarak çıktı verir. Örneğin;kalınlk

* **İtalik Yazma - <i>**

Etiketler içerisine yazılanları *italik (eğik)* olarak çıktı verir. Örneğin;

italik

* **Altı Çizgili Yazma - <u>**

Etiketler içerisine yazılanları altı çizili olarak çıktı verir. Örneğin;

altıçizli

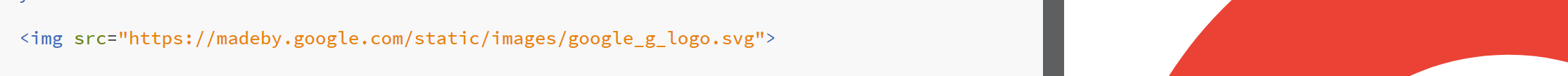
* **Bağlantı Verme - <a>**

Etiketlerin arasına girilen yazıya, fotoğrafa vs. tıklandığında ayrı bir sayfaya yönlendirmek için kullanılan etikettir. **<a href=”#”> #**’yerine bir web sitesi adresi yazılabilir. **#** bir yere bağlanmayacağı anlamına gelir. **<body>** etiketlerinin içerisine yazılır. Örneğin;



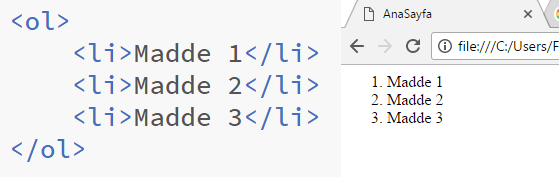
* **Görsel Ekleme - <img>**

Sayfamızda bir görsel yayınlamak istiyorsak kullanacağımız etikettir. Fotoğrafın kaynağını belirttiğimiz **src** ifadesi vardır. **<body>** etiketinin içerisine yazılır. Örneğin;



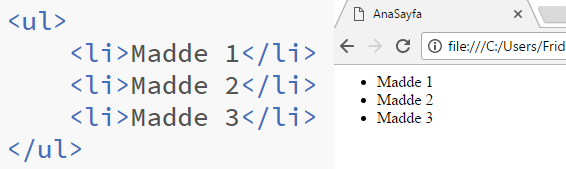
* **Sıralı Liste Oluşturma - <ol>**

1., 2., gibi belirli bir sıraya göre sıralanan maddelerin olduğu listeyi oluşturmamız için kullanılır. <body> etiketinin içerisine yazılır. Elemanları **<li>** etiketiyle oluşturulur. Örneğin;



* **Sırasız Liste Oluşturma - <ul>**

Herhangi bir alfabetik veya numaratik sıraya ihtiyacımız olmadığında kullanılır. **<body>** etiketinin içerisine yazılır. Elemanları **<li>** etiketiyle oluşturulur. Örneğin;



Web Tasarım Atölyesi

## CSS Temelleri

CSS, web sitemizde kullanacağımız stil şablonlarıdır. HTML sayfamızda kullandığımız elementleri görsel olarak düzenlememizi böylece daha düzenli ve şık bir web sitesi elde etmemizi sağlar.



Görselde görmüş olduğunuz bir CSS kodudur. CSS kodumuzu HTML sayfasına bağlamanın birden fazla yolu vardır. Biz bunlardan biri olan link yöntemiyle CSS dosyamızı belirtiyoruz. HTML sayfasında **<head>** etiketleri arasına **<link href=”cssdosyası” rel=”stylesheet” />** kodunu yazıyoruz. CSS yazarken bir element üzerinden gidebilir veya kendi class ve id’lerinizi oluşturup sitenizi geliştirebilirsiniz. Peki nedir bu class ve id?

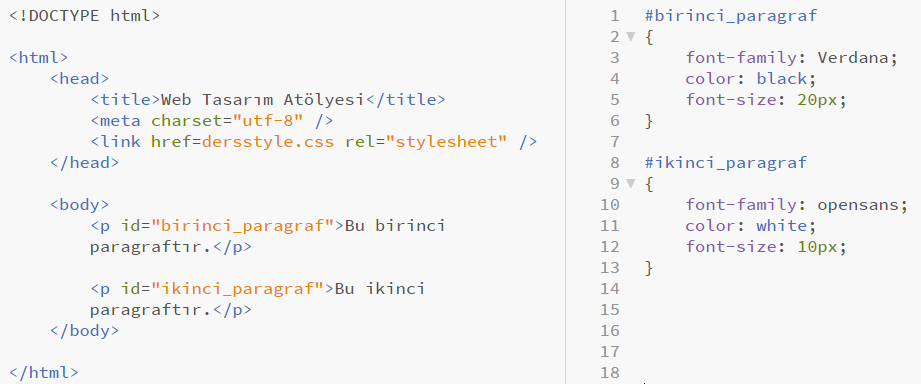
* **Class nedir?**

Bir sayfada birden fazla elemente aynı özellikleri vermek için kullanılır. CSS kodlarını yazarken vereceğimiz ismin önüne “**.**” koyarız. HTML sayfasına eklerken ise **class=”class\_adi”** şeklinde etiketin içerisinde kullanırız. Örnek verecek olursak;



* **ID nedir?**

Bir sayfada sadece bir elementte kullanılır. CSS kodlarını yazarken vereceğimiz ismin önüne “**#**” koyarız. HTML sayfasına eklerken ise **id=”id\_adi”** şeklinde etiketin içerisinde kullanırız. Örnek verecek olursak;



## Metin Biçimleme

* **Font Değiştirme**

**font-family: Verdana, monospace, sans-serif;**

Bu kodu yazdığımızda eğer font bulunamıyorsa soldan başlayarak bir sonrakini dener. Yani alternatif fontları belirtebiliriz. Fontları yazarken aralarına virgül koymayı unutmayın.

* **Font Boyutlandırma**

font-size: 20px;

Bu kodla fontumuzun büyüklüğünü belirtiyoruz.

* **Metnin Rengini Değiştirme**

**color: black;**

Bu kodu yazarken renk tipini üç şekilde belirleyebiliriz. En kolayı renklerin isimlerini yazarak bunu yapabiliriz. Örneğin **black, white, red, blue** gibi. HEX kodunu vererek rengi bulabiliriz. Örneğin; **#F54EF** gibi. RGB kodunu vererek yine rengini buluruz. Örneğin; **rgba(34, 13, 12, 45, 0.5)**

* **Metni Hizalama**

**text-align: center;**

Metni hizalamak için kullanılır. **Sağa, sola, ortaya** veya **her iki yana** da hizalamayı bu kodla yaparız. Örneğin; **left, right, center, justify**

* **Metine Gölge Efekti Verme**

**text-shadow: 1px 2px 10px black;**

Bu kodda gördüğünüz değerler sırasıyla; yatayda ne kadar uzakta, dikeyde ne kadar uzakta, bulanıklık ve renk değerleridir.

## Arka Plan Ayarları

Sayfamızda kullanacağımız elementlerin veya sayfanın tamamının arka planına müdahale etmek istersek aşağıdaki kodları kullanıyoruz.

* **Arka Plan Renklendirme**

**background-color: yellow;**

Arka plana istediğimiz rengi vermemizi sağlar. Rengi daha önce metin renklendirme kısmında gördüğümüz gibi adını, HEX kodunu veya RGB kodunu yazarak belirleyebiliriz.

* **Arka Plana Fotoğraf Ekleme**

**background-image: url(resmin uzantısı);**

Arka plana istediğimiz fotoğrafı eklemek için kullanılır. url() içerisine fotoğrafın bulunduğu uzantı yazılır.

* **Arka Planı Sabitleme**

**background-attachment: fixed;**

Bu kod sayfayı yukarı aşağı kaydırırken arka planın sabit içeriğin hareketli olmasını sağlar.

* **Arka Planın Uyumlu Olması**

**background-size: cover;**

Bu kod arka planın pencere büyüklüğüne uyumlu olarak büyüyüp küçülmesini sağlar.

* **Elemente Çerçeve Ekleme**

**border: 10px solid black;**

Bu kodda verilmiş olan değerler sırasıyla; kalınlık, çizgi tipi ve renktir.

* **Çerçevenin Köşeleri Yuvarlama**

**border-radius: 5px;**

Çizdiğimiz çerçevenin köşeleri yuvarlar.

* **Nesneye Dış Boşluk Verme**

**margin: 10px 40px 20px 80px;**

Bu kodda verilen değerler sırasıyla; yukarı, sağ, aşağı ve sol taraftan ne kadar boşluk verileceğini belirtiyor.

* **Nesneyi Ortalama**

**margin: auto;**

Bu kod nesnenin sayfanın ortasında yer almasını sağlar.

**Bootstrap (https://getbootstrap.com/)**

Web sitelerin ön yazı tasarımı için dünyada en çok kullanılan, mobil ve responsive tasarımı önceliklendiren kütüphane.

**Jekyll – Github Pages**