**Typescript**

**Setup**

C:\ > **npm install -g typescript >** Typescript son sürüm global kurulum

C:\ > **npm install -g typescript@4.4.2 >** Typescript belli bir sürüm global kurulum

C:\ > **npm uninstall -g typescript >** Typescript’i sistemden kaldırmak

C:\ > **tsc -v >** Typescript versiyon kontrol

**Genel**

🡪 JS’den daha güzel/modern bir söz dizimi sunar. (Superset of Javascript)

🡪 Derleme aşamasında anlık kod hatalarını bildirir. (Compile-time Error)

🡪 JS’de bir değişkene sonradan farklı bir değer atanabilir ama TS buna izin vermez.

C:\ > **tsc main.ts >** Typescript kodlarını main.js dosyasına dönüştürür.

**Değişkenler**

let a: number = 5;

let b: string = 'a';

let c: boolean = true;

let d: any;

let e: number[] = [1,2,3];

let f: Array<number> = [1,2,3];

let g: any[] = [1,'a',true];

let h: [string,number,boolean] = ['a', 5 , false]; // tuple

🡪 Değişken ile birlikte tür ve değer de atanabilir.

🡪 e değişkenininde diziye number türü dışında eleman eklenemez.

🡪 e ve f değişkenleri yazımları farklı olsa da benzer özellik taşır.

🡪 **tuple** olarak da bilinen h değişkeninde tür ve değer sıralaması aynı olmalıdır.

**Enum Kullanımı**

const krediPayment = 0;

const havalePayment = 1;

const eftPayment = 2;

// Enum

enum Payment {kredi, havale, eft};

let kredi = Payment.kredi;

let havale = Payment.havale;

let eft = Payment.eft;

🡪 Enum kullanımında değer atanmadığı takdirde index değerleri çağırılır.

🡪 Ancak **enum Payment {kredi=0, havale=1, eft=5}** şeklinde değer de atanabilir.