

Javascript ve Typescript farkı

- JavaScript geliştirildiğinde, JavaScript geliştirme ekibi JavaScript'i istemci tarafı programlama dili olarak tanıttı. Ancak insanlar JavaScript'i kullanırken, geliştiriciler JavaScript'in sunucu tarafı programlama dili olarak da kullanılabileceğini öğrendiler. Ancak JavaScript büyürken JavaScript kodu karmaşık ve ağır hale geldi. Bu nedenle JavaScript, Nesne yönelimli bir programlama dilinin gereksinimlerini tam olarak karşılayamadı.
- TS nesne yönelimli programlama dilidir, JS betik dilidir.
- TS statik veri tiplerine sahiptir, JS dinamik olarak tanımlanır.
- TS parametrelili fonksiyonları destekler.
- TypeScript kodu tarayıcılar tarafından anlaşılmıyor. Bu nedenle, eğer kod TypeScript'te yazılmışsa, derlenir JavaScript'e çevirir.
- JavaScript'te yazılan kod ne olursa olsun, uzantı .js'den .ts'ye değiştirilerek TypeScript'e dönüştürülebilir
- TypeScript kodu herhangi bir tarayıcıda, cihazda veya herhangi bir işletim sisteminde çalıştırılabilir.
- TypeScript, Statik yazma olarak bilinen bir özelliğe sahiptir, ancak JavaScript'te bu özellik yoktur.
- TypeScript modülleri desteklerken JavaScript modülleri desteklemez.
- TypeScript'de Interface vardır, ancak JavaScript'in bir Interface yoktur.
- TypeScript, derleme zamanında hataları vurgular, oysa JavaScript çalışma zamanında hataları gösterir.
- JS doğrudan tarayıcıda çalıştırılır. TS çalıştırılmaz.

Özetle; JS istemci tarafı kodlama dili olması, TS Nesne yönelimli derlenmiş dil olmasıdır.

Örnek

1 class Hello {	1 var Hello = (function () {
2 WriteHelloWorld() {	2 function Hello() {
3 alert("Hello World");	3 }
4 }	4 var Hello.prototype.WriteHelloWorld = function () {
5 }	5 alert("Hello World");
6	6 };
7 var hello = new Hello();	7 return Hello;
8 hello.WriteHelloWorld();	8 })();
	9 var hello = new Hello();
	10 hello.WriteHelloWorld();

Solda TS kodunda OOP dillerinde olduğu gibi classlar kullanabiliyoruz. TS tarafında bu kod sağdaki gibi JS koduna çevriliyor.

NodeJs ile sunucu ayağa kaldırma

```
1  var http=require("http");
2  var server=http.createServer(function(request,response){
3    response.end("hello world");
4
5  });
6  var port=3000;
7  server.listen(port,function(){
8    |   console.log("server ayağa kalktı. port:"+port);
9  })
```

1.satırda http modülünü oluşturuyoruz.

2. satırda sunucumuzu oluşturuyoruz. Gelen bir http isteğini işlemek ve bir http yanıtını döndürmek içindir. Bu işlevin iki argümanı olmalıdır. Bir istek nesnesi ve yanıt nesnesi tanımlandı.

İstek nesnesi gelen http isteğinin tüm verilerini yakalar , yanıt nesnesi sunucu için http yanıtları döndürür.

3.satırda sunucumuza erişildiğinde “hello world” mesajı döndürüyoruz.

6 ve 7. Satırda ulaşacağımız sunucumuzu host ve porta bağlıyoruz.

8.satırda ise program doğru şekilde çalıştığında konsol ekranına mesaj yazdırıyoruz.

Bu örnekte web sunucusunu ayağa kaldırdık. Sunucumuza metin değeri gönderdik. Farklı içerik türleri de döndürülebilir. Örneğin json, html, csv vs.