Tartalom

Változó típusok

A változók a typescript esetében „esősen típusosak, azaz meg kell adni nekik a típusokat, annak „kitalálását” nem a böngészőnkre bízzuk.

Változók létrehozása

A változók létrehozása a következ képpen történik

hatáskör valtozoNev : tipus = érték;

Hatáskör: Let, var, vagy const

A változóknak javascriptben 3 típusát megkülönböztetjük, de ez nem a bennük tárolandó értékek típusát határozza meg hanem a változók funkcióit.

let:

Hasonlóan működik, mint korábban JavaScript eseténben.

const:

Szintén hasonlóan működik, mint a korábban bemutatott JavaScript esetén.

var:

A TypeScript változói a var kulcsszóval is deklarálhatók, ugyanúgy, mint a JavaScriptben. A hatókör szabályai ugyanazok maradnak, mint a JavaScriptben. Újra deklarálás esetén, a rendszer külön figyelmeztet, ha más típusú változót szeretnénk létrehozni ugyanazon a néven!

valtozoNev: Névadási szabályok

A JavaScript nyelvben az alábbi dolgokra kell ügyelnünk:

- A **változó nevében lehet betű akár kicsi akár nagy**, ugyanakkor bár lehet **ékezetes** betűt használni azokat mégis próbáljuk meg kerülni.
- A változó nevében **NEM lehet space** karakter.
- A változó **NEM kezdődhet számmal**!
- A változó **kezdődhet „_”***(alulvonás)* és **„$”***(dollár jel)* karakterekkel bár ez utóbbi ritkább.
- A betűk mellett a „_” és „$” karakterek is bárhol lehetnek a változó nevében.
- Törekedjünk arra, hogy a **név utaljon a** benne eltárolt **változó feladatára**.
- A nyelv **case-sensivitve**, ami azt jelenti, hogy a kis és nagybetűk között különbséget tesz! tehát „alma” nem egyenlő az „Alma” változó névvel
- A legtöbbször a **camelCase** változó neveket használjuk, a változót több szóra bontjuk, az első szó kisbetűvel a többi szó nagybetűvel kezdődik és az összes szót összefűzzük.
  pl.: camelCaseValtozonev, ezIsEgyValtozo, ittEgyFontosValtozo stb…
  Esetleg használhatjuk az alulvonás karaktert is mint a szavakat elválaszó elemet, de ez is ritkább:
  pl.: camel_case_valtozonev, ez_is_egy_valtozo, itt_egy_fontos_valtozo stb…

Érdekesség: Egyes programozók a const-t típusú változók neveit csupa nagybetűvel írják, a könnyebb olvashatóság érdekében, de ez egyéni megszokás kérdése.

Változó típusa

Az első dolog, amiben igazán különbözik a TypeScript az eddigiektől.

: boolean
logikai változó típus, igaz hamis érték rendelhető hozzú

: number
szám típus, egész és valós számok rendelhetők hozzá

: string
szöveg (karakterlánc típus, karakterTömb), idézőjelek között adhatunk meg neki értéket

Ezeket a típusú változókat, ha más típusú értékkel próbáljuk feltölteni, akkor hibát fog dobni nekünk a TypeScript!

Speciális változó típusok

: any
any-hez rendelhetünk bármien típust mint ahogy javascriptnél már megszoktuk, de ugyanazokkal a veszélyekkel jár mint anno, ha rossz változótípust adunk meg egy művelethez az eredmény hibára futnat

: unknow
hasonló mint az any, viszont van egy nagy előnye, használat előtt(mielőtt csinálnánk vele valamint, módosításon kívül) meg kell határoznunk, épp aktuálisan milyen értéktípust vesz fel, azaz unknow típusú változóra nem hívhatjuk meg például a .toUpperCase típusú függvényt csak azután, hogy azt szöveggé alakítottuk, valamilyen módon!

Egyszerű tömbök

Tömbök esetében is, mint az előbb bemutatott primitív változók esetében, is megadhatók annak típusai az alábbi módokon.

```
var szovegesTomb: string[]=["alma", "körte", "barack", "szilva"];
//Csak szöveg típusú változókat fogad el
```

vagy az alábbi módon, esetleg akik C# ban programoztak azoknak ismerős lehet:

```
var szovegesTomb: Array<string>=["fps", "tps", "rpg","szimulátor"];
//Csak szöveg típusú változókat fogad el

var szamTomb: Array<number>=[3,7,13,21,42];
//Csak szám típusú változókat fogad el
```