



Typescript

Diego Iván Oliveros Acosta

Agenda



¿Qué es?



Sintaxis



Ejemplos



Herramientas
necesarias



Herramientas
sugeridas



Referencias

¿Qué es?

Extensión de Javascript.

Verificación de tipos

Mantiene el comportamiento en runtime.

¿TS vs JS?

Chequeo de editor

Auto completado

Interfaces

Tipado estático

TypeScript 4.4

4.3.4 (17 de junio de 2021 (3 mes y 19 días))

- Análisis de flujo de control de condiciones con alias y discriminantes
- Firmas de índice de patrón de cadena de símbolo y plantilla
- Valor predeterminado del ***unknown*** tipo en las variables de captura (**`--useUnknownInCatchVariables`**)
- Tipos de propiedad opcionales exactos (**`--exactOptionalPropertyTypes`**)
- Static Blocks in Classes
- tsc –help Updates and Improvements
- Auto-Imports Show True Paths in Completion Lists
- Code suggestions#
-

Sintaxis más veloz

```
for (const talent of talento){  
    console.log(talent);  
}
```

```
for (var _i = 0, talento_1 = talento; _i < talento_1  
.length; _i++) {  
    var talent = talento_1[_i];  
    console.log(talent);  
}
```

Algunos tipos

```
var areatotal:number=8;
const uncolor:number=0xff00;
const binario: number=0b00110011;
var texto1: string="El área de la figura es: ";
var estavivo:boolean=true; //no se puede usar 1.
let talento:string=["java", "python", "html", "css", "javascript", "typescript"];
```


No siempre es más simple

```
console.log("Hola Mundo!");  
function add(number1:number, number2:number): number  
{  
    return number1+number2;  
}  
console.log("Hola Mundo!");  
function add(number1, number2) {  
    return number1 + number2;  
}
```

enum

```
• enum ROLE {Employee, Manager, Admin, Developer };
• var ROLE;
• (function (ROLE) {
•     ROLE[ROLE["Employee"] = 0] = "Employee";
•     ROLE[ROLE["Manager"] = 1] = "Manager";
•     ROLE[ROLE["Admin"] = 2] = "Admin";
•     ROLE[ROLE["Developer"] = 3] = "Developer";
• })(ROLE || (ROLE = {}));
• ;
```


Herramientas necesarias

1. Contar con un ID(visual)
2. Instalar **Node**
3. Habilitar la ejecución de scripts
4. Instalar TypeScript compiler:

`npm install -g typescript`



TypeScript



Visual Studio Code



Recomendaciones.

- <https://www.typescriptlang.org/play?>





Ejercicios

JavaScript / TypeScript

JavaScript / TypeScript

```
var nombre;  
nombre = "Miguelo";  
var edad;  
edad = 30;
```

```
var PERSONAJE = {  
  nombre: nombre,  
  edad: edad  
};
```

```
let nombre:string;  
nombre = "Miguelo" ;  
let edad:number;  
edad = 30 ;
```

```
const PERSONAJE = {  
  nombre:nombre ,  
  edad:edad  
}
```

Interface

```
var spiderman = {  
    nombre: "Peter parket",  
    poderes: ["trepar", "fuerza", "agilidad", "telas de araña"]  
};
```

```
interface superHero {  
    nombre:string;  
    poderes:string[];  
}
```


Ahora en TypeScript vamos a crear la clase Rombo, la cual debe tener dos propiedades:

- DiagonalVertical y DiagonalHorizontal.
- Le añadiremos un constructor al que le pasaremos los valores anteriores cuando instanciamos el objeto.
- Y también debe de tener un método que calcule el área, que será la multiplicación de DiagonalVertical * DiagonalHorizontal. Este método devolverá un número.

```
var Rombo = /** @class */ (function () {  
    function Rombo(diagonalVertical, diagonalHorizontal) {  
        this.diagonalVertical = diagonalVertical;  
        this.diagonalHorizontal = diagonalHorizontal;  
    }  
    Rombo.prototype.calcularArea = function () {  
        return this.diagonalHorizontal * this.diagonalVertical;  
    };  
    ;  
    return Rombo;  
})();
```

Var function flecha Suma

```
function suma( a, b ){  
    return (a + b)  
}
```

```
let varfunctionflechaSuma  
= (a:number, b:number) =>  
  (a+b);
```

Referencias

- <https://www.typescriptlang.org/>
- <https://nodejs.org/en/>
- <https://www.w3schools.com/js/default.asp>
- <https://www.typescriptlang.org/docs/handbook/release-notes/typescript-4-4.html>