



**UNIVERSIDAD
DE ANTIOQUIA**

Introducción y Aplicaciones a TypeScript

Carlos Andrés Mesa Roldán
Mateo Mesa Restrepo

Facultad de Ingeniería
Universidad de Antioquia
2024

TypeScript es...



TypeScript es...

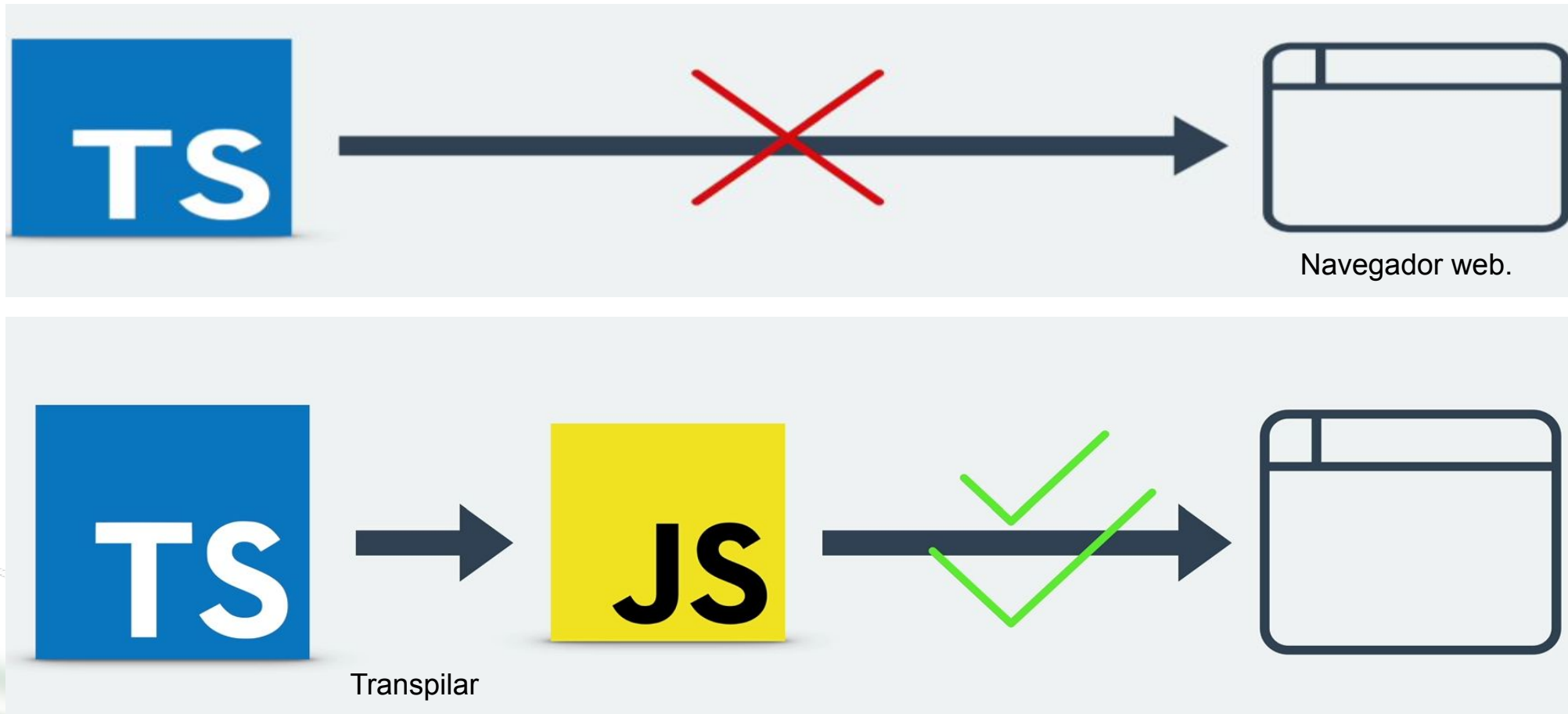


TypeScript es...

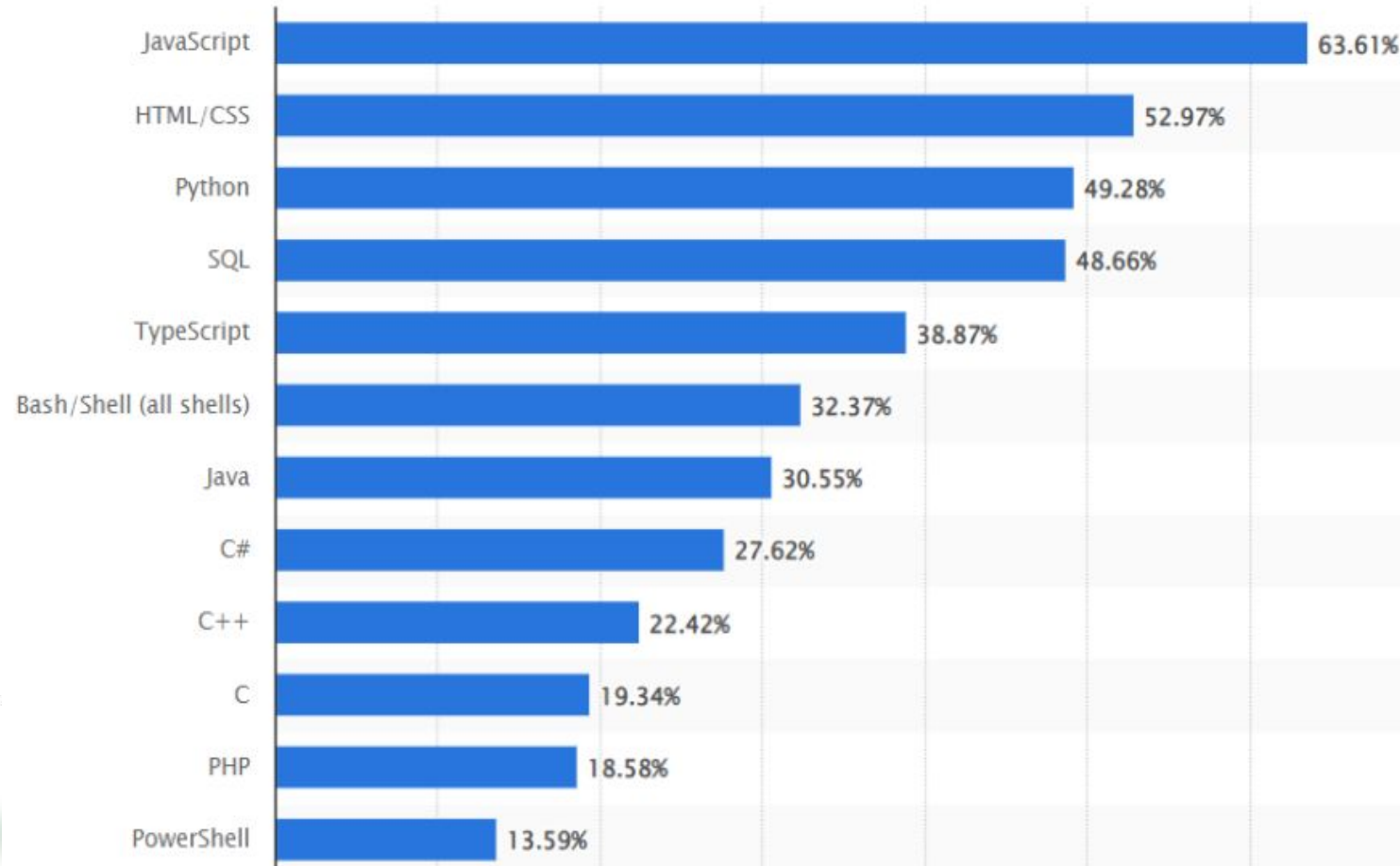


- Tipados estrictos.
- Otro tipo de manejo a las clases.
- Autocompletados que incluso pasan desapercibidos por los IDE.

Una sutil pero importante diferencia...



Crecimiento de JavaScript



El artículo en Pixlogix proporciona un análisis completo sobre el uso de JavaScript en 2024, destacando su integración en áreas como **IoT**, **WebAssembly**, y su crecimiento continuo en el desarrollo de aplicaciones web y móviles

Se basa en datos agregados del ecosistema JavaScript, incluyendo descargas de npm, estadísticas de GitHub, y adopción por parte de empresas globales como Google y Apple

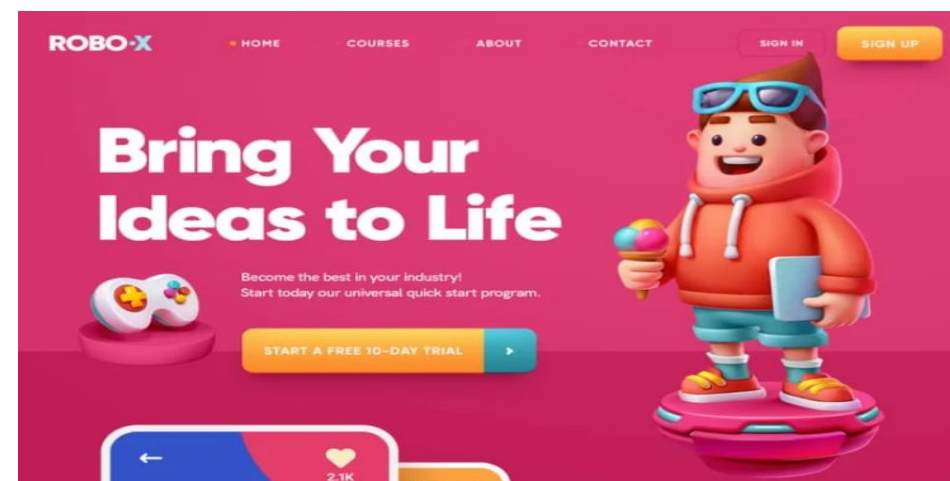
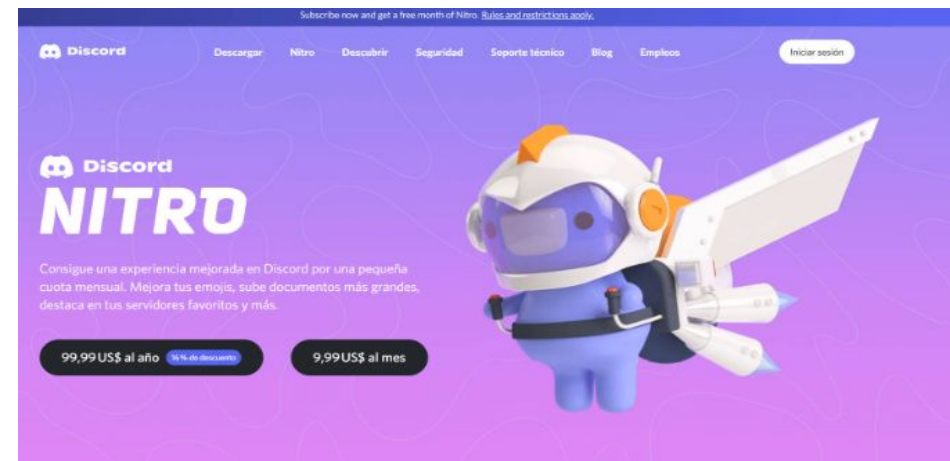
<https://www.pixlogix.com/javascript-usage-statistics/>



En los años 90...



Hoy...



¿Por que TypeScript?

Falta de Tipado Estático

JavaScript es un lenguaje dinámico, lo cual significa que no hay control sobre los tipos de datos (como strings, números, etc.)



Mantenibilidad en Proyectos Grandes

TypeScript introduce tipado estático y clases, lo cual ayuda a organizar el código y facilita el trabajo en equipo.



Compatibilidad con Funcionalidades Modernas

TypeScript permitió el uso de estas características modernas al compilar el código a una versión de JavaScript compatible con navegadores



Mejor Experiencia de Desarrollo

TypeScript proporciona autocompletado, detección de errores en tiempo de desarrollo y otras mejoras



```
1 + "1" // Devuelve 2
```

```
$(document).ready(function() {
    // ...
    $(this).on('click', function() {
        // ...
    });
    // ...
});
```

```
function showLog(text: string): void;
function showLog(number: number): void;
```

```
export class MyCtrl {
    constructor($scope: IMyscope) {
        $scope.person.name = "Lucifer";
        $scope.persons = [];
    }
    // ...
}

angular.module('myApp', []).controller('MyCtrl', function($scope) {
    // ...
});
```


¿Por qué no JavaScript?



Mejor Experiencia de Desarrollo

```
1 ▼ const sum = (a:number, b:number) => {  
2   return a + b  
3 }  
4
```

```
5 sum(3, 'EDteam')
```

Argument of type 'string' is not assignable to parameter of type 'number'.
typescript



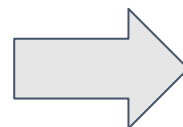
Hoy en día es fácil superar miles de líneas de código en una aplicación de JavaScript

Y trabajar en equipo en un proyecto de JavaScript puede ser retador

- **Tipado de variables**
- **Errores en tiempo de escritura (linter)**
- **Auto completado dependiendo de las variables**
- **Clases y módulos (ES6)**
- **Validación de objetos dentro de objetos**
- **Tipado de respuestas http**



```
const calculateAge = (birthday) => {  
  var ageDifMs = Date.now() - birthday.getTime();  
  var ageDate = new Date(ageDifMs);  
  
  return Math.abs(ageDate.getUTCFullYear() - 1970);  
}
```



```
const calculateAge = (birthday: Date): number => {  
  const ageDifMs: number = Date.now() - birthday.getTime();  
  const ageDate: Date = new Date(ageDifMs);  
  
  return Math.abs(ageDate.getUTCFullYear() - 1970);  
};
```

Declaración de variables, arreglos y funciones

```
let nombre: string = "Juan"; // Tipo explícito string
let edad: number = 25;      // Tipo explícito number
let esEstudiante: boolean = true; // Tipo explícito boolean
```

```
let numeros: number[] = [1, 2, 3, 4, 5];
let nombres: string[] = ["Juan", "Ana", "Luis"];
```

```
const sumar: (a: number, b: number) => number = (a, b) => {
    return a + b;
};
```

Compatibilidad de TypeScript con herramientas modernas

