Introduction to TypeScript: Should we migrate to TypeScript?

Perkenalan

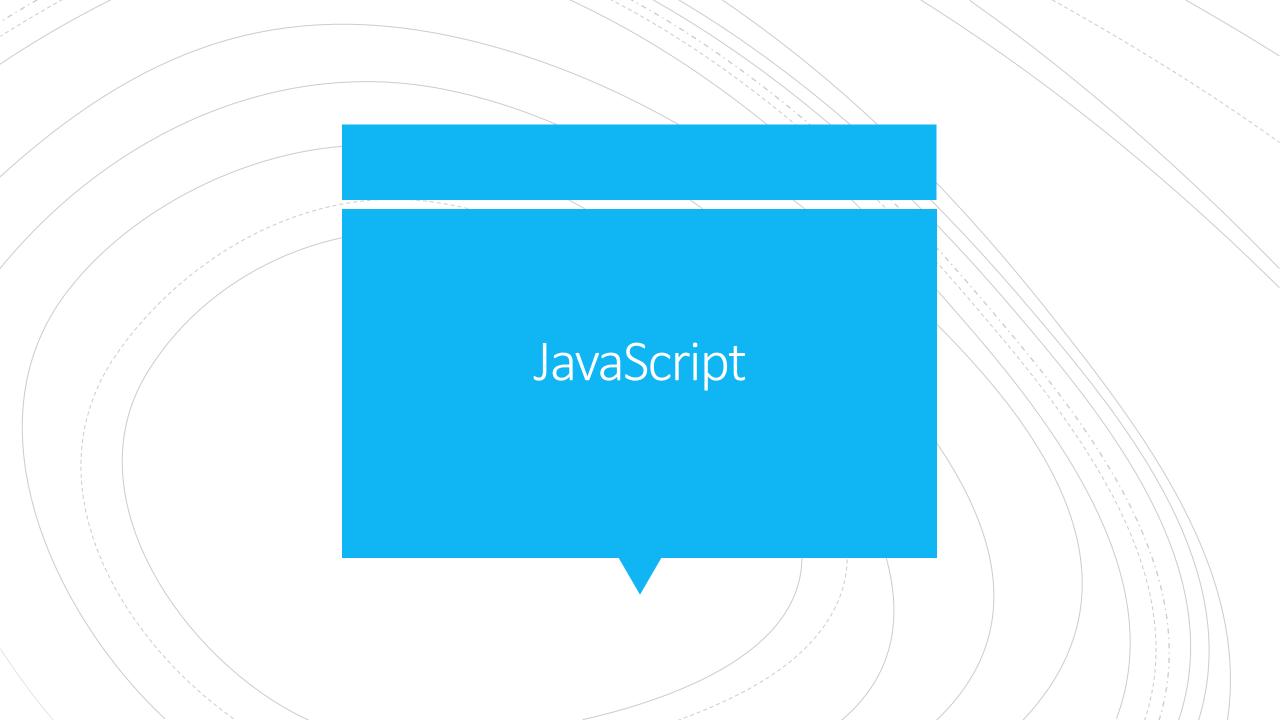
Nama: Kurniawan

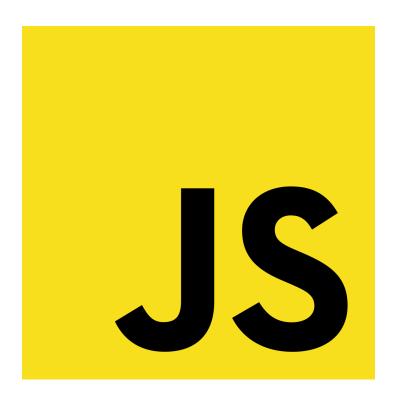
TTL: Bangkalan, 9 Mei 1993

Alamat: Kedungwonokerto, Prambon, Sidoarjo

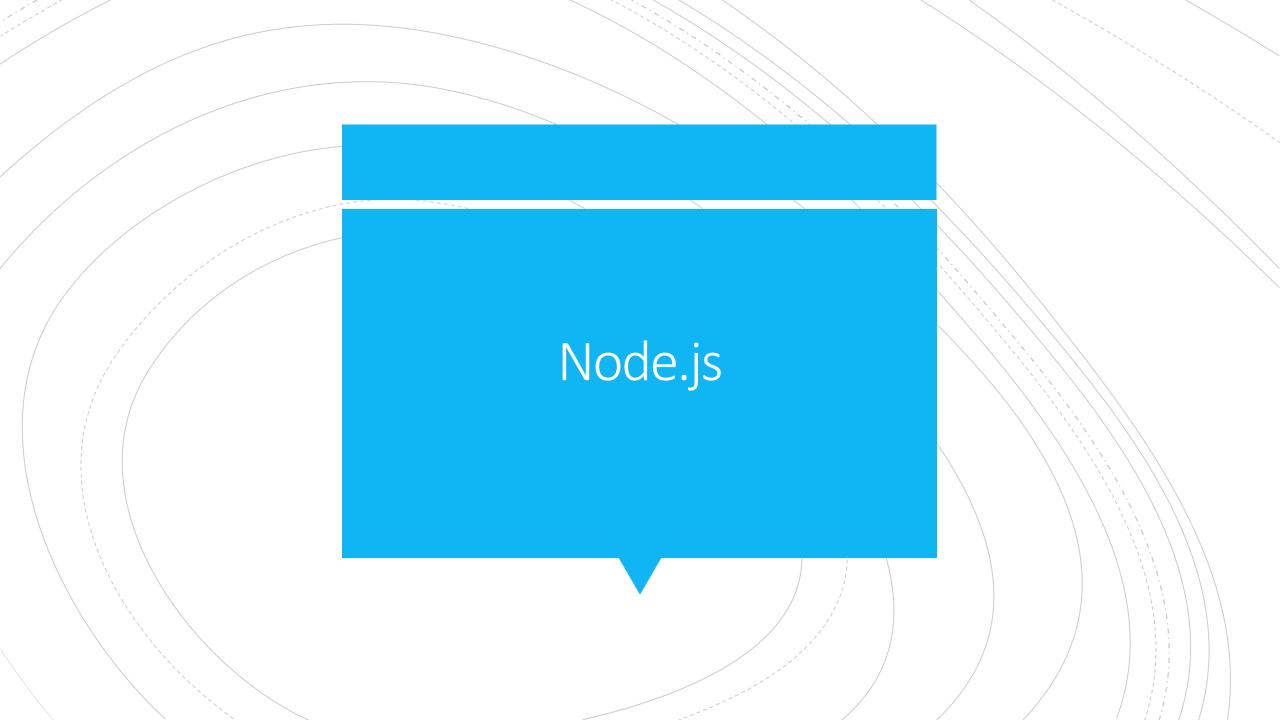
Aktifitas: Bekerja di salah satu startup bernama Lumpang sebagai CTO

Fokus: Saat ini senang mengulik dunia Backend dan Mobile developer



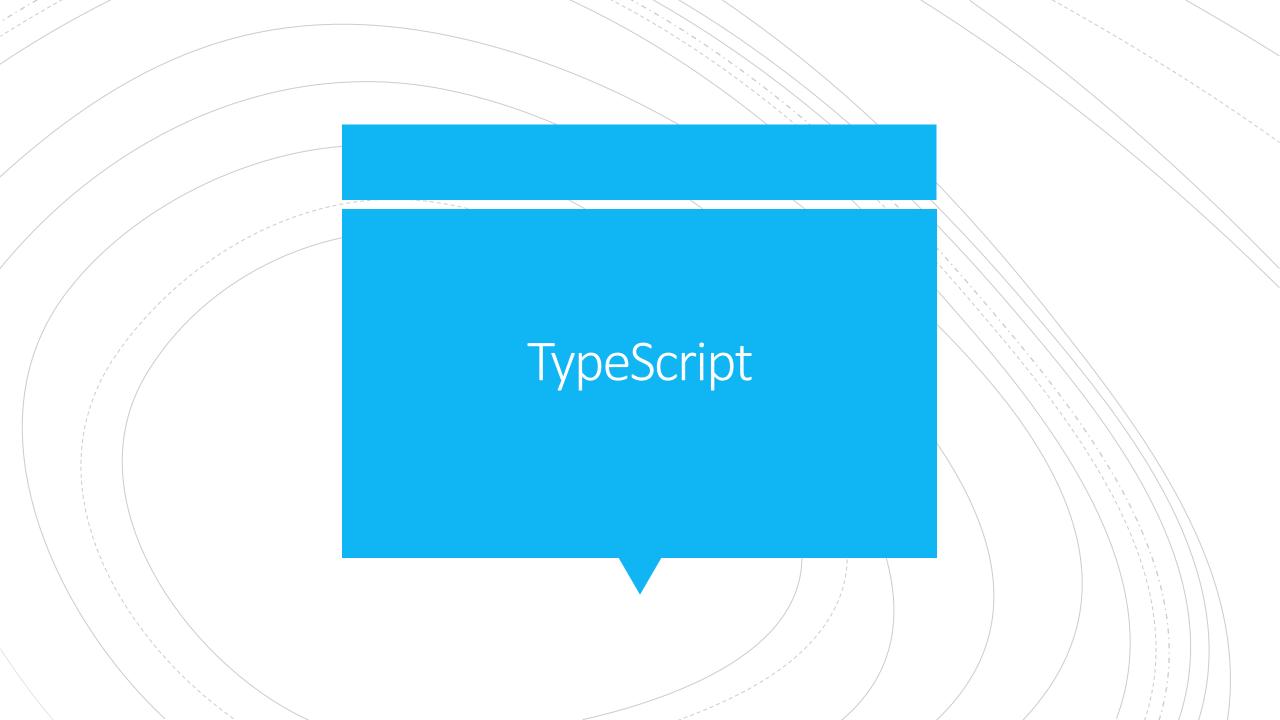


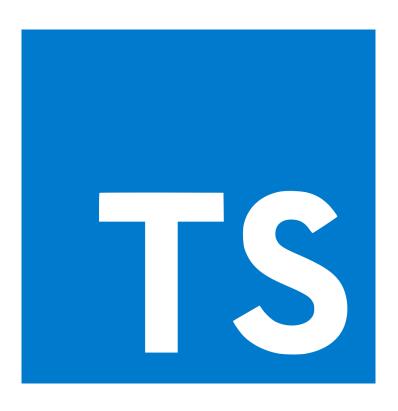
 JavaScript is the programming language of HTML and the Web.



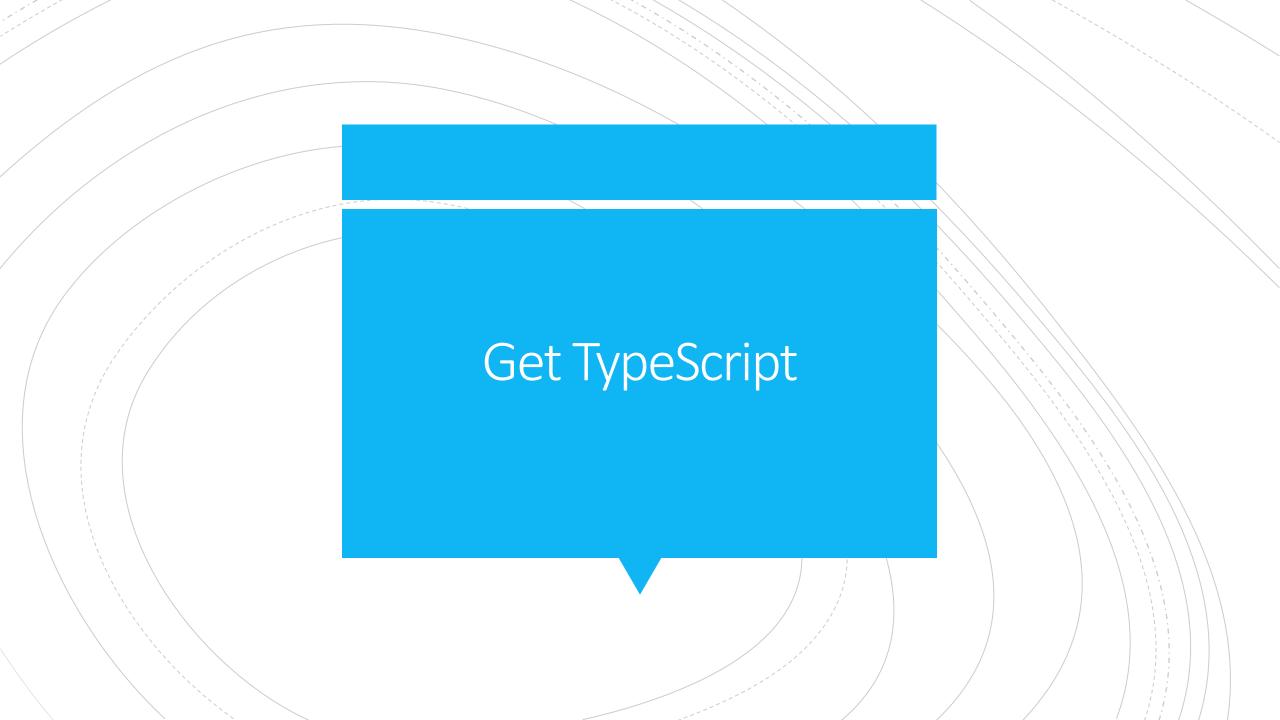


- Node.js is an open source server environment.
- Node.js allows you to run JavaScript on the server.





- is an open source programming language developed and maintained by Microsoft.
- It is a strict syntactical superset of JavaScript,
 and adds optional static typing to the language.
- TypeScript is a typed superset of JavaScript that compiles to plain JavaScript



Get TypeScript

Node.js

The command-line TypeScript compiler can be installed as a Node.js package.

INSTALL

npm install -g typescript

COMPILE

tsc helloworld.ts

Visual Studio



Visual Studio 2019



Visual Studio Code



Visual Studio 2017

And More...



Sublime Text



Emacs



WebStorm

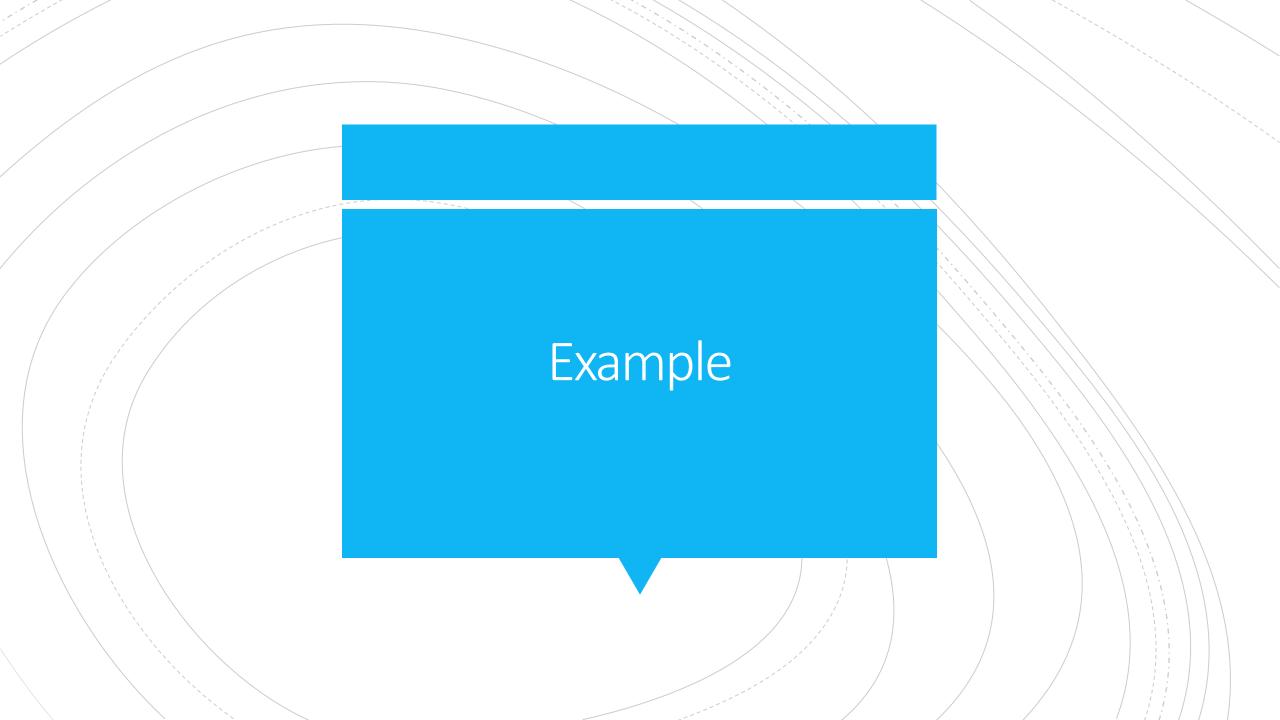


Eclipse

Atom



Vim







surabaya.js • Following



surabaya.js Typescript Basic

Pada Typescript kita bisa menentukan tipe tertentu pada sebuah variable, variable yang sudah memiliki tipe ini tidak bisa diisi dengan sembarang data, hal ini membuat kode yang kita tulis jauh lebih aman.

#ts #typescript #javascript #js #surabaya #programming #coding #nodejs

16w



minehp js ato ts...? 😂 😂 😂

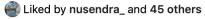


16w 2 likes Reply



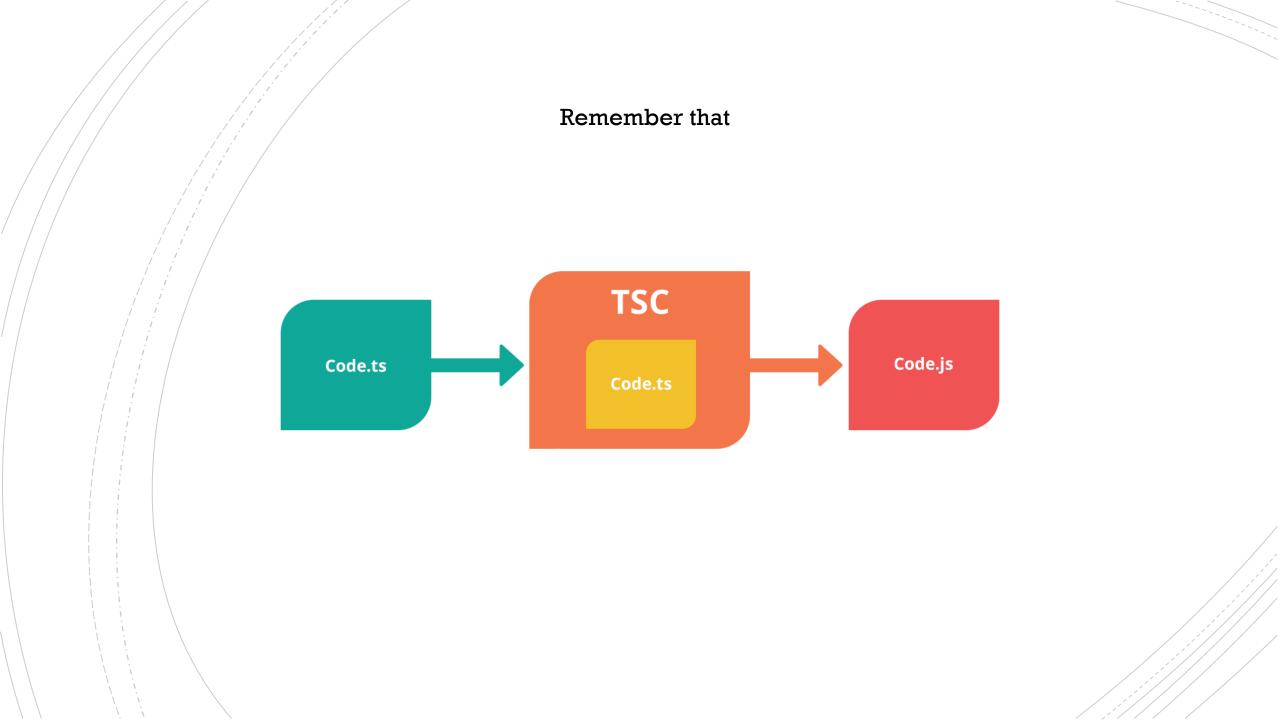




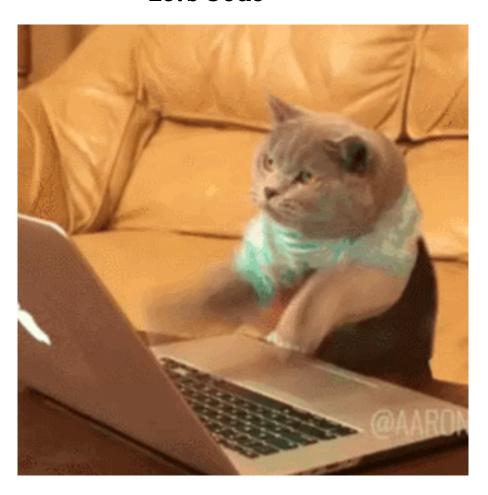


FEBRUARY 19

Add a comment...



Let's Code



Hello World

```
console.log('hello world')
const nama: string = 'surabaya js'
const sayHello = () => {
 console.log(`Halo ${nama}`)
sayHello()
```

var, const, let

```
• • •
const firstName = 'Thor'
var animal = 'dog'
let canFly: boolean = false
function changeAnimal(kind: string) {
  const firstName = 'Marcopolo'
  animal = kind
  if (kind === 'bird') {
    canFly = true
changeAnimal('bird')
animal = 'cat'
```

Callback & Promise

```
function isGenap(num: number, cb: any) {
  if (num % 2 === 0) {
    cb(true)
  } else {
    cb(false)
isGenap(4, function(result: boolean) {
  if (result) {
    console.log('Genap')
  } else {
    console.log('Ganjil')
})
```

```
• • •
function isGanjil(num: number) {
  return new Promise(function(resolve, reject) {
    if (num % 2 !== 0) {
      resolve(true)
    } else {
      resolve(false)
   reject()
isGanjil(8).then(function(result) {
  if (result) {
    console.log('Ganjil')
  } else {
    console.log('Genap')
.catch(function(error) {
  console.log('Gak tau')
})
```

Class

```
class Bank {
  private _balance: number
  constructor(balance: number) {
    this._balance = balance
    console.log(`constructor balance = ${this._balance}`)
  public withdraw(amount: number): any {
   if (this._balance > amount) {
      this._balance -= amount
      return {
       message: 'Berhasil',
         newBalance: this._balance
    return {
      message: 'Gagal saldo tidak cukup'
  public get balance() : number {
    return this._balance
  public set balance(amount: number) {
    this._balance = amount;
    console.log('balance', this._balance)
const budi = new Bank(1000)
budi.balance = 2000
console.log('withdraw', budi.withdraw(200))
console.log('balance', budi.balance)
```

Generic

```
const identity = <T, P>(name: T, age: P): T => {
  return name
}

const person = identity<string, number>('Stark', 67)
console.log(person)
```

Optional Parameter

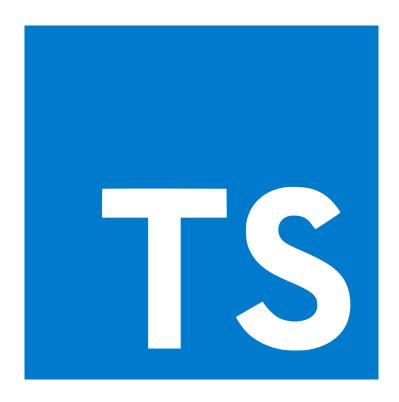
```
const optionalParamFunc = (name: string, gender?: string, age: number = 23) => {
  const person = {
    name: name,
    age: age,
    gender: gender
  }
  console.log(person)
}

optionalParamFunc('parto', 'm')
```

Interface

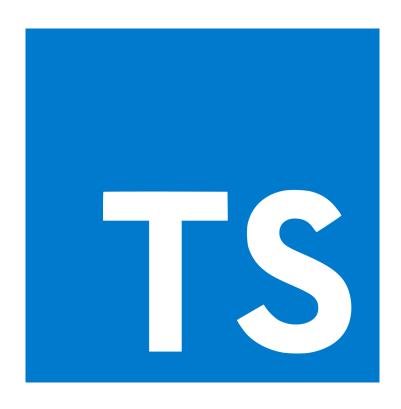
```
interface IArgs {
  name: string
  age: number
function initPerson(args: IArgs) {
  const person = {
    age: args.age,
    name: args.name
  console.log(person)
const argsPerson: IArgs = {
  age: 12,
  name: 'parto'
initPerson(argsPerson)
```





PROS

- Making it easier to read and debug.
- Highly productive development like static checking FAST DEBUGGING.
- TypeScript gives us all the benefits of ES6 (ECMAScript 6)
- TypeScript code can be compiled as per ES5 and ES6 standards to support the latest browser.
- Superset of JavaScript
- Used in popular frameworks
- Typescript is just JavaScript (with some additional features)
- Portable



CONS

- Learning curve
- Type annotations everywhere in their code Wasting time
- Produce bigger size better than origin js file (it added roughly 30% to the total code size)



- https://www.w3schools.com/js/
- https://www.w3schools.com/nodejs/
- https://www.typescriptlang.org/
- https://en.wikipedia.org/wiki/Microsoft_TypeScript
- https://www.instagram.com/p/BuEX_ZIA-_T/
- https://dzone.com/articles/what-is-typescript-andwhy-use-it
- https://engineering.carsguide.com.au/es5-vs-es6syntax-6c8350fa6998