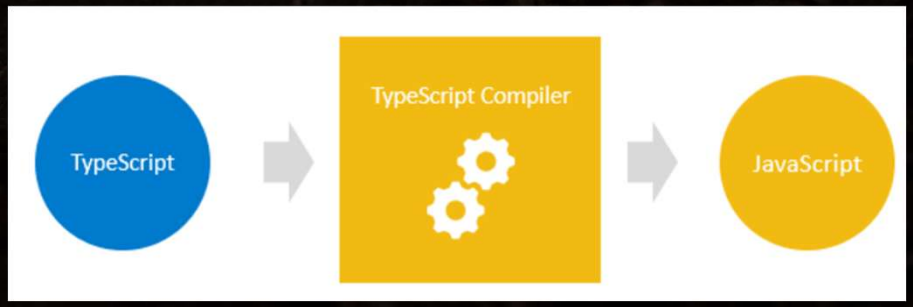
Typescript

Imelda Doharta Aritonang

Pengenalan Typescript

Typescript merupakan javascript yang diberi type.
File ekstensionnya adalah .ts, misalnya typescript.ts. Sedangkan ekstension file javascript adalah .js misalnya javascript.js

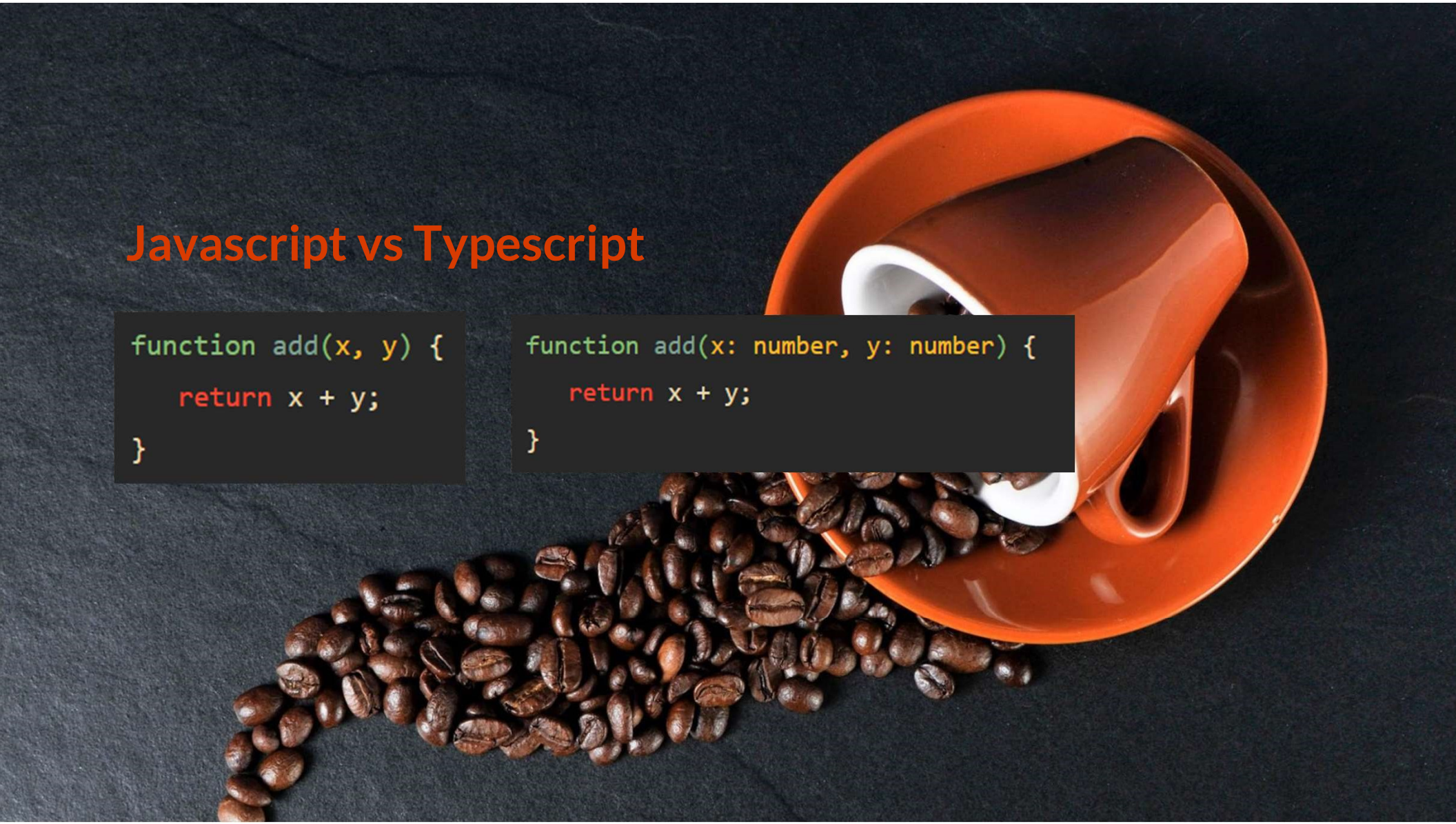
Typescript menggunakan sintaks javascript dan menambahkan sintaks tambahan untuk mendukung type.



Javascript vs Typescript

```
function add(x, y) {
    return x + y;
}
```

```
function add(x: number, y: number) {
    return x + y;
}
```



Setup Typescript

NodeJs, environment dimana Typescript akan dijalankan

Typescript Compiler, module node.js yang mengcompile Typescript menjadi Javascript

Visual Studio Code sebagai editor

Setup Typescript (Sambungan)

Install NodeJs

- Silahkan buka web NodeJs
- Download file installer NodeJs sesuai sistem operasi yang dimiliki
- Klik file installer untuk melanjutkan install nodejs
- Pastikan sudah terinstall dengan ketik node –v di command prompt
- Jika ada pesan bahwa node tidak dikenali tambahkan path instalasi di System Variables pada Envirenment System

Install Typescript Compiler

- Untuk versi nodejs versi 10 ke atas, npm akan terinstal otomatis.
- Gunakan npm untuk menginstal modul Typescript compiler

```
npm install -g typescript
tsc --v
```

- Run file typescript atau javascript bisa dengan ts-node.
- File typescript error run dengan node
- Compile file typescript ke javascript: tsc nama file typescript

Install Visual Studio Code

- Silahkan buka web Visual Studio Code
- Download file installer Visual Studio Code sesuai sistem operasi yang dimiliki
- Klik file installer untuk melanjutkan install visual studio code

Error When compile using tsc

```
PS D:\WorkSpace\typescript> tsc basics.ts tsc : File
C:\Users\mn\AppData\Roaming\npm\tsc.ps1 cannot be loaded because running scripts is
disabled on this system. For more information, see about_Execution_Policies at
https:/go.microsoft.com/fwlink/?LinkID=135170.
```

@Cerlancism. type the following commands in powershell

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned

or

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine
```

firebase - VSC PowerShell. After npm updating packages .ps1 cannot be loaded because running scripts is disabled on this system - Stack Overflow

atau delete tsc.ps1

TIPE DATA

Tipe data ada dua yaitu primitive type dan object type. Primitive type mewarisi yang di javascript.

Object type misalnya array, fungsi, class .

Fungsi tipe pada Typescript:

1. Mendeteksi ada error sebelum eksekusi program
2. Mengetahui nilai yang akan diberikan ke suatu variable

Sintaks Typescript `: type` setelah identifier

```
let person: {
   name: string;
   age: number
};

person = {
   name: 'John',
   age: 25
}; // valid
```

```
let greeting : (name: string) => string;
```

```
function increment(counter: number) {
    return counter++;
}
```

```
let variableName: type;
let variableName: type = value;
const constantName: type = value;
```

```
let counter: number;
```

```
counter = 1;
```

```
let name: string = 'John';
let age: number = 25;
let active: boolean = true;
```

```
let names: string[] = ['John', 'Jane', 'Peter', 'David', 'Mary'];
```

TIPE DATA (Sambungan)

Tipe data dapat diprediksi Typescrypt berdasarkan value yang diberikan pada variabel disebut **Type inference**. Contoh:

1. Inisialisasi variabel
2. Set nilai default parameter
3. Menentukan return type

Tipe data yang diberikan pada saat mendeklarasikan sebuah variable disebut Type annotation.

Type annotation digunakan ketika:

1. Mendeklarasikan variabel dan nilai diberi kemudian
2. Ketika variabel sudah pasti harus menerima nilai tertentu saja
3. Ketika fungsi sudah pasti harus memberi input tertentu

- `number`
- `bigint`
- `string`
- `boolean`
- `null`
- `undefined`
- `symbol`

TIPE DATA (Sambungan)

```
let employee = {
    firstName: 'John',
    lastName: 'Doe',
    age: 25,
    jobTitle: 'Web Developer'
};

console.log(employee);
```

```
let employee: {
    firstName: string;
    lastName: string;
    age: number;
    jobTitle: string;
};
```

```
let employee: {
    firstName: string;
    lastName: string;
    age: number;
    jobTitle: string;
} = {
    firstName: 'John',
    lastName: 'Doe',
    age: 25,
    jobTitle: 'Web Developer'
};
```

```
let scores : (string | number)[];
scores = ['Programming', 5, 'Software Design', 4];
```

```
employee = {
    firstName: 'John',
    lastName: 'Doe',
    age: 25,
    jobTitle: 'Web Developer'
};
```

TIPE DATA (Sambungan)

Tuple, mirip array bedanya tuple memiliki size dan type yang sudah fixed

```
let color: [number, number, number] = [255, 0, 0];
```

```
let bgColor, headerColor: [number, number, number, number?];
bgColor = [0, 255, 255, 0.5];
headerColor = [0, 255, 255];
```

enum, singkatan enumerated type, digunakan untuk menampung kumpulan variable yang nilainya konstan atau sudah pasti, dan masih berkaitan

```
enum Month {
    Jan,
    Feb,
    Mar,
    Apr,
    May,
    Jun,
    Jul,
    Aug,
    Sep,
    Oct,
    Nov,
    Dec
};
```

```
enum ApprovalStatus {
    draft,
    submitted,
    approved,
    rejected
};
```

Fungsi

```
let add: (x: number, y: number) => number;
```

```
add = function (x: number, y: number) {
    return x + y;
};
```

```
let add: (a: number, b: number) => number =
    function (x: number, y: number) {
        return x + y;
    };
```

```
add = function (x: string, y: string): number {
    return x.concat(y).length;
};
```

Fungsi (Sambungan)

Optional Parameter

```
function multiply(a: number, b: number, c?: number): number {

    if (typeof c !== 'undefined') {
        return a * b * c;
    }
    return a * b;
}
```

Default Parameter

```
function name(parameter1=defaultValue1,...) {
   // do something
}
```

```
function applyDiscount(price, discount = 0.05) {
    return price * (1 - discount);
}

console.log(applyDiscount(100)); // 95
```

```
function name(parameter1:type=defaultvalue1, parameter2:type=defaultvalue2,...)
   //
}
```

```
function applyDiscount(price: number, discount: number = 0.05): number {
    return price * (1 - discount);
}

console.log(applyDiscount(100)); // 95
```

```
let promotion: (price: number, discount: number = 0.05) => number;
```

Fungsi (Sambungan)

Rest Parameter

```
function getTotal(...numbers: number[]): number {
    let total = 0;
    numbers.forEach((num) => total += num);
    return total;
}
```

```
console.log(getTotal()); // 0
console.log(getTotal(10, 20)); // 30
console.log(getTotal(10, 20, 30)); // 60
```

Overloading Function

```
function addNumbers(a: number, b: number): number {
    return a + b;
}

function addStrings(a: string, b: string): string {
    return a + b;
}
```

```
function add(a: number | string, b: number | string): number | string {
    if (typeof a === 'number' && typeof b === 'number')
        return a + b;

    if (typeof a === 'string' && typeof b === 'string')
        return a + b;
}
```

```
function add(a: any, b: any): any {
    return a + b;
}
console.log(add('a','b'))
```

OOP atau Pemrograman Berbasis Objek

Pemrograman yang berorientasi objek yang bertujuan untuk mempermudah pengembangan program dengan mengikuti model yang sudah ada

Class suatu *blue print* untuk membuat suatu objek.

Objek merupakan instansi atau cetakan, wujud dari sebuah Class, yang akan mewarisi semua milik Class tersebut.

OOP atau Pemrograman Berbasis Objek (Sambungan)

Encapsulation, membungkus property atau atribut dan method dalam class menggunakan access modifier(public, protected, private)

Inheritance, mewariskan semua properti dan method sesuai access modifier kepada turunan class. Yang mewariskan disebut super class atau parent class, yang diwariskan disebut child class

Abstraction, sebuah class yang didalamnya ada abstract method yakni method yang belum diimplementasikan, implementasi beberapa atau semua method abstract pada subclassnya

Interface, mirip abstraction namun didalam interface semua method harus diimplementasikan

Polymorphism, suatu teknik pada OOP untuk membuat banyak bentuk suatu method, ada 2 Teknik yaitu override dan overload

THANK YOU!