



**Microsoft** Partner  
Silver Learning



# TypeScript

Модулі та простір імен

# TypeScript

## Introduction



Igor Михайличенко  
Frontend Engineer

 iMykhailchenko

 Ihor Mykhailchenko



# TypeScript

## Тема уроку

# Модулі та простір імен



# TypeScript

## План уроку

1. Використання просторів імен
2. Робота з модулями
3. Налаштування webpack



# TypeScript

## Простір імен

Простір імен - механізм для групування коду та визначення окремих областей видимості.

```
namespace Sample {  
    export class MyClass1 {  
        public message() {  
            console.log("Sample2.MyClass1.message");  
        }  
    }  
}
```

```
let temp = new Sample.MyClass1();  
temp.message();
```



# TypeScript

## Triple Slash Reference

```
///
```

- Директива для TS компілятора
- Визначає залежність для даного файлу
- Вказується на початку файлу

```
tsc --outFile app.js main.ts
```

Компілюємо main.ts і всі пов'язані файли в один файл app.js



# TypeScript

## Модулі

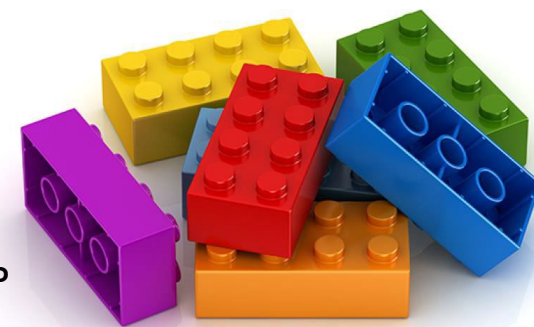
**Модуль** - окремий файл зі своєю областю видимості. Змінні, функції, класи та інші мовні конструкції, визначені в модулі, не доступні за його межами.

Файл, у якому на верхньому рівні перебуває `import` чи `export`, є модулем.

Залежності між модулями визначаються за допомогою ключового слова `import`.

Щоб імпортувати модулі, використовуються завантажувачі модулів, які відповідають за знаходження модуля та його завантаження перед виконанням програми.

Завантажувачі модулів - CommonJS, SystemJS, require.js



# TypeScript

## Виконання import

```
export interface Validator {  
    validate(value: string): boolean;  
}
```

```
import { Validator } from "01-validators";
```

JavaScript modules via script tag  - LS

Usage

% of all users

Global

68.08% + 1.11% = 69.19%

Loading JavaScript module scripts using `<script type="module">`

Includes support for the `nomodule` attribute.

Current aligned

Usage relative

Date relative

Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			64		4 10.3				
	16	2 59	65	11	11.2				4
11	17	60	66	11.1	11.3	all	66	11.8	6.2
	18	61	67	TP					
		62	68						
			69						

TypeScript



# TypeScript

## Робота з модулями

Основні способи визначення залежностей між модулями :

- AMD Modules
- CommonJS
- ES6 modules

### AMD modules

```
// add.js
define([], function () {
    return {
        add: function () { }
    });
});

// main.js
define([
    './path/to/add',
], function (math) {
    // using math
});
```

### CommonJS

```
// add.js
module.exports = {
    add: function () { }
}

// main.js
var math = require('./path/to/add.js');
```

### ES6

```
// add.js
export function add() { }
```

```
// main.js
import { add } from "../path/to/add.js"
```

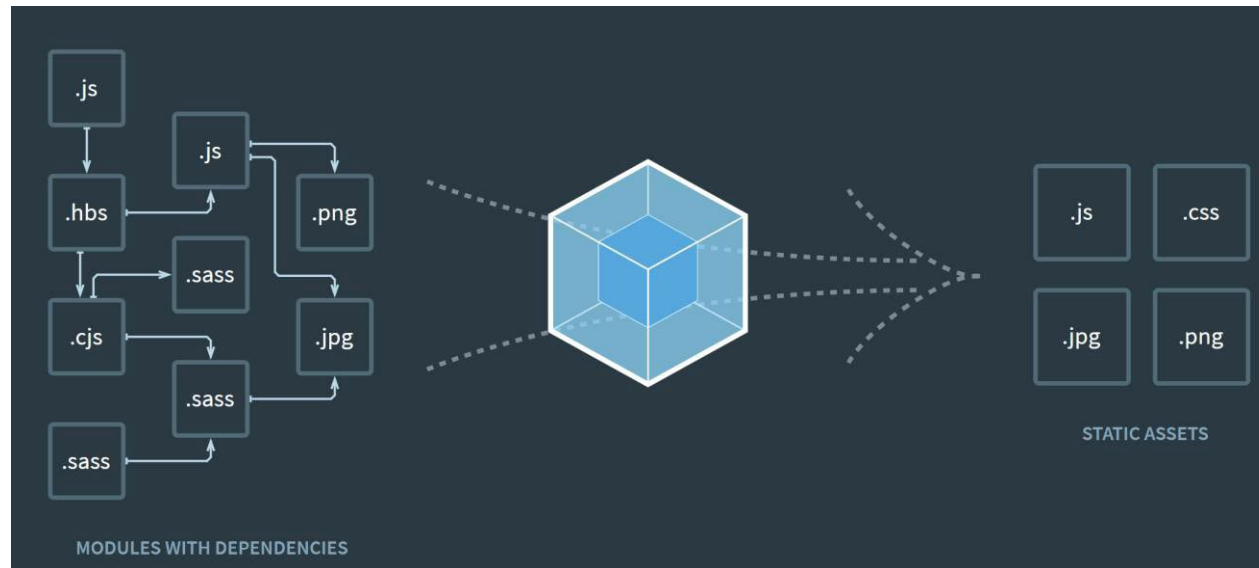


# TypeScript

## Webpack

**Webpack** – інструмент для збирання JS файлів у пакети з дотриманням правильного порядку використання файлів.

<https://webpack.js.org/>



# Інформаційний відеосервіс для розробників програмного забезпечення



# Перевірка знань

TestProvider.com



Перевірте, як ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. За його допомогою ви можете оцінити свій рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Успішне проходження фінального тестування дозволить отримати відповідний Сертифікат.

# TypeScript

Дякую за увагу! До нової зустрічі!



Ігор Михайличенко  
Frontend Engineer

