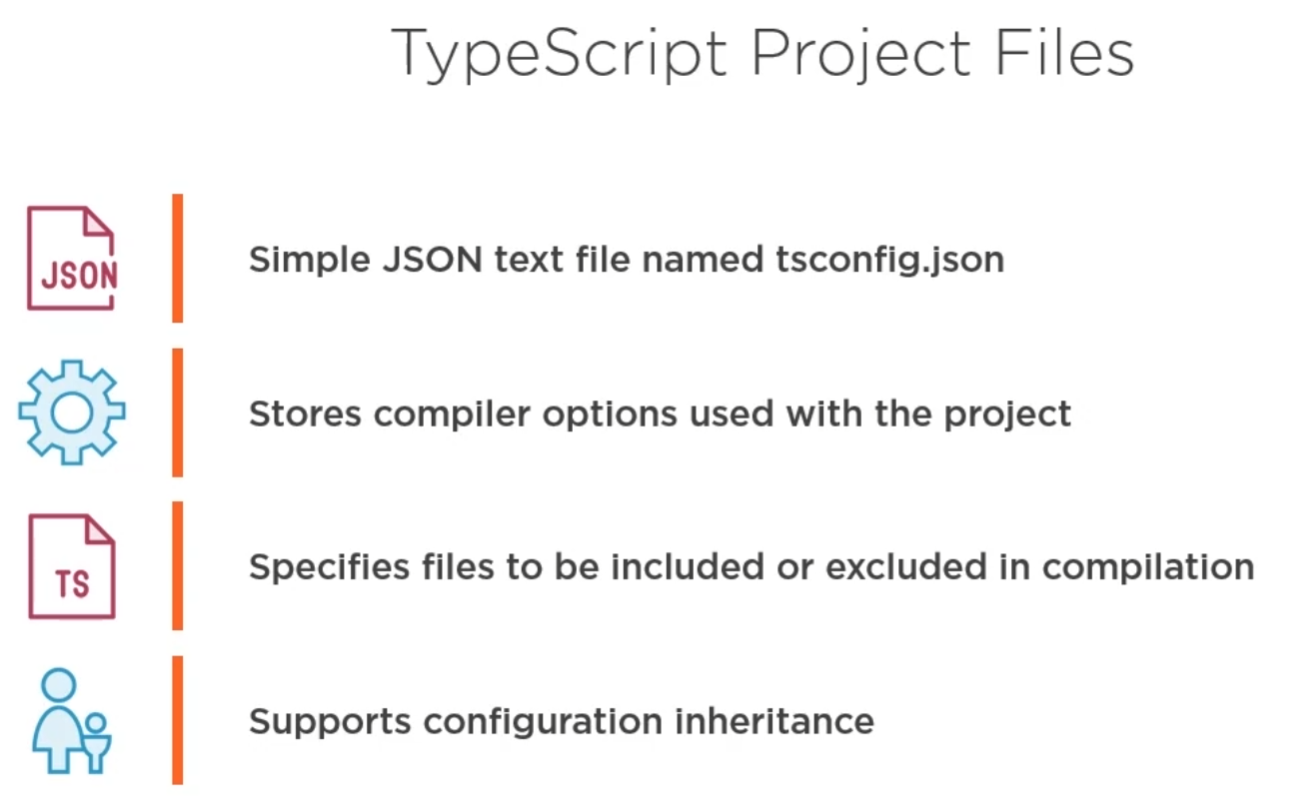
* Ts config file intro



* Tsconfig.json looks like:

一張含有 文字 的圖片

自動產生的描述

* **When starting a new project:**

**use tsc - - init to create default tsconfig.json**

* 母檔的tsconfig基本上許多功能都沒有開，只有幾個例如target預設轉成**es5**
* **\*\* 有用options:**

1. 把 module拿掉 (用es6 的 import 跟 export就好)
2. 設 sourceMap to true >> 讓你可以在browser裡面看到source file，如此可以在browser裡面debug
3. 指定outDir 為 “js”，這是說ts compile之後的去處，這樣會在tsconfig那一層自動建立js資料夾，並將compile好的js丟進去
4. Strict 預設就是true，但其底下的各種checking可以去看一下各自做了些什麼
5. 在最底下加入 “watch” option， set to true。這個option 會讓compiler持續rinning，所以每次source file存檔時會自動recompile
6. 除了wcompilerOptions之外，在其下面可宣告files的區塊，手動指定每次compile時想要compile的ts檔(原本是要compile時手動輸入檔名)

一張含有 文字 的圖片

自動產生的描述

手動輸入要compile的檔案跟輸出路徑

一張含有 文字 的圖片

自動產生的描述

>> 一個proj常有多個 tsconfig(根據不同ts檔做不同compile option)

>> 這邊在最外層的tsconfig.base.json是base檔，用 tsc –init做出

>> 在app folder中的tsconfig.json則是子檔，在這邊看到可用extends來inheritance母檔。

>> 重點1: option中的 “removeComments” 會讓在ts中你打的comment在compile之後自動被刪除

>> 重點2: include跟之前的files很像，這邊宣告 “/\*\*/\*” 代表在子tsconfig這一層或內部所有資料夾中(/\*\*)的所有ts檔(/\*)，在執行tsc 時都會被compile

一張含有 文字 的圖片

自動產生的描述

Note1: 加上 ! 讓ts開心 (他覺得可能潛在會是null的東西，這邊因為是bind在html上，他不確定到底有沒有這個元件，所以如果自己確定的話就在這邊加上驚嘆號)

Lesson 1:

一張含有 文字 的圖片

自動產生的描述

Review: Do not use var

一張含有 文字 的圖片

自動產生的描述

Js 引擎會將var 宣告變數的指令至頂，所以才讓他是global

但在指定他之前就使用他在閱讀code上會製造混亂

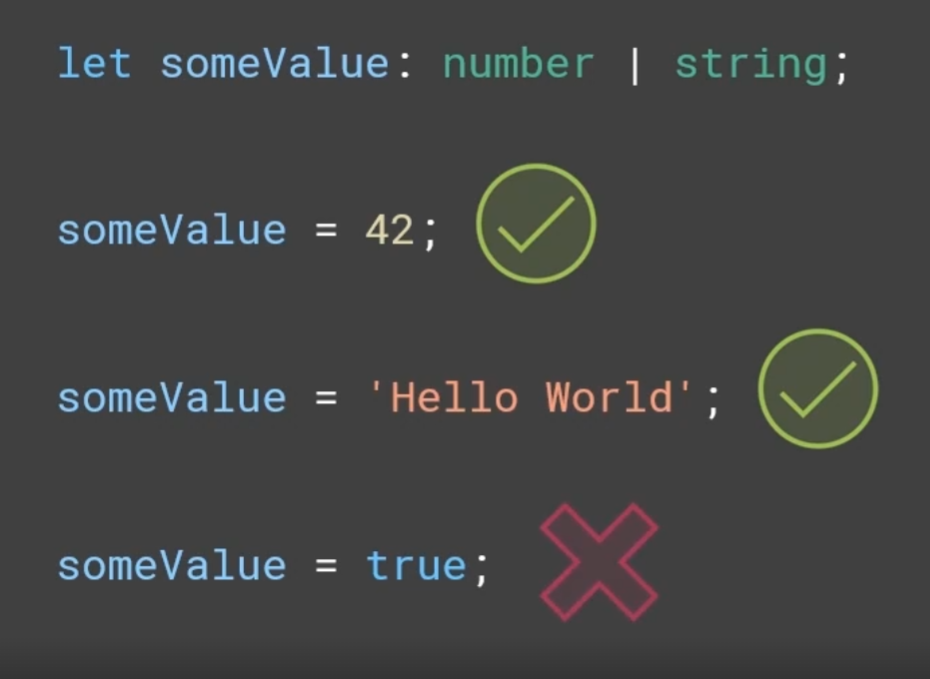
>> 如果用let或 const則不然，使用變數必須在其宣告之後

* Type annotation and Type interface

一張含有 文字 的圖片

自動產生的描述

* Union type: 複合型別 (可能是string 或 number)



\*\* By default, Null and Undefined 是所有型別的union type

>> means 所有type的東西給他null 或 undefined 都是可行的(不需要特別寫在annotation裡面)

但，如果在compilerOptions裡面宣告



To true的話，此時就必須在宣告變數時說他可能是null 或undefined(用union type)，如此只要有可能是null的狀況就會報錯，除非你一開始有宣告

一張含有 文字 的圖片

自動產生的描述

* Type assertions (generic) >> 泛型

用 <> 或 “as” 來做 type的assertion

一張含有 文字 的圖片

自動產生的描述

* Control Flow Type Analysis

一張含有 文字 的圖片

自動產生的描述

Message 預設是 HTMLElement (因為是從getElementById來的)

但也可以在宣告時用union type讓他可以是string

接下來在if block 宣告 if(typeof message === ‘string’)，則此時在block內部對於message型別的判斷會自動修正為只有string，同理在else block裡面會自動將型別修正為只有HTMLElement

* Type annotation to functions

一張含有 文字 的圖片

自動產生的描述

參數型別宣告跟變數一樣

注意這邊message後面的問號，是指這個參數是optional

>> 在js所有的變數都是optional的(有夠混亂)，甚至可以傳入超過function宣告數量的參數

>> 但在ts立面所有的變數都是required，如果要讓他變optional就要加上問號

* noImplicityAny compilerOptions

>> 預設在function中不須宣告型別，他會是Any type，但如果使用這個option，就必須在function中宣告參數跟回傳type (防呆裝置)

一張含有 文字 的圖片

自動產生的描述

* 參數defalut >> 跟 js 一樣

\*\* 注意! 給default 參數自帶給參數optional (前面的?)，如果沒給default參數又傳undefined或null進來的話會報錯，但如果有給就safe

一張含有 文字 的圖片

自動產生的描述

* Function type: 像是abstract function 先不寫funciton實際內容而是把他的架構(type)先寫出來

一張含有 文字 的圖片

自動產生的描述

>> 注意宣告方式 const func: (parameter: paraType) => returnType

* Interfaces and classes >> 基本上跟C# 一模模一樣樣

一張含有 桌 的圖片

自動產生的描述

* Create interface >> shape of a object

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 桌 的圖片

自動產生的描述

注意 access modifier:

1. Default是public，可不用寫出來
2. Private代表該property只能在該class中做使用
3. Protected代表該property只能在該class跟任何繼承該class的class中做使用

* Class creation

一張含有 文字 的圖片

自動產生的描述

1. 終於搞懂了…. C# 起初會先給private然後在用getter 跟 setter，是要讓外人只能接觸到處理好的型式的property，真的存在物件中的row property不讓人動

>> so 這邊一樣，title 先設private，然後用getter 跟 setter 來控制property的進出

1. 注意在class定義中function就直接從function名開始寫即可

Class 繼承 Class >> use extends keyword

* 注意: 繼承class不用重打母class的property，但可新增新的property
* 建立出來的實體可以觸及母及子class的property

一張含有 文字 的圖片

自動產生的描述

Class 繼承 interface

* 注意: 使用 implements keyword
* 因為interface只是合約，所以在class中要把interface中宣告的property寫出來

一張含有 文字 的圖片

自動產生的描述

* File configuration (c# 的using)

一張含有 文字 的圖片

自動產生的描述

>> 神奇的宣告方式

>> its just a single line comment that appear at the top of a file, then the compiler will

>> telling compiler the dependencies, so whenever the app.ts is compiled, compiler will automatically compile player.ts too!

* outFile compilerOptions >> 將輸出的js檔照妥當的順序集中在一個js中

一張含有 文字 的圖片

自動產生的描述



原本interface 跟 class等等的ts file 會各自被compile成多個js，如此就必須在html那邊加上所有的js檔(且順序不能搞錯)，如果宣告outFile路徑的話就可以解決這個問題

* static(靜態) member >> 可直接觸及

一張含有 文字 的圖片

自動產生的描述

>> 注意: 一樣要先說是哪個class(不一定要做出實體)，然後再接觸該static property

Constructor:

一張含有 文字 的圖片

自動產生的描述

* 注意:

1. 宣告constructor() 即可，readonly 的property只可以透過constructor來aaign
2. Super() 是去call 母class(Super class)的constructor! >> 因為Developer的ctor不用pass參數，因此這邊的super() 一樣是空function

* Use parameter property to quickly define a class!

一張含有 文字 的圖片

自動產生的描述