Introduction to TypeScript

This week will introduce you to TypeScript. In this week you are going to create a console application with TypeScript which will cover Types, Interfaces and Generic types.

**Note!**Before doing any assignments, install Node.JS by following [these instructions](https://moodle.lut.fi/mod/page/view.php?id=1484546).

**Note!**Half of the points are granted if the tasks are done correctly, and half of the points if TypeScript is being used correctly.

**Requirements and Scoring**

**1. TypeScript environment and usage**

After you have successfully installed node, initialize TypeScript. Make sure you have following modifications in your tsconfig.json file:  
"target": "ESNext",  
"outDir": "./dist",  
"noEmitOnError": true,  
  
After you have successfully initialized TypeScript add a file named app.ts  
File should be able to run with command "tsc && node dist/app.js" and the application should print "Hello World!" to the console.

**2. Creating type TVehicle**

In your app.ts file create a type "TVehicle". It should have attributes model (string), color (string), year (number) and power (number).

Create a Vehicle with following attributes:

{

    model: "Boring generic vehicle",

    color: "Red",

    year: 1993,

    power: 60

}

Console log the created vehicle.

**3. Extending interface**

Create interface "IVehicle" which should have same attributes as the "TVehicle" type. Extend the created IVehicle interface, and create following extended interfaces: "ICar", "IBoat" and "IPlane".  
In addition to the attributes in the "IVehicle" interface, Car should have bodyType (string) and wheelCount(number), Boat should have draft (number) and Plane should have wingspan(number).  
  
Create instances of all interfaces with the given values:

Car: {   
    model: "Ford focus",   
    color: "Green",   
    year: 2016,   
    power: 150,   
    bodyType: "Hatchback",   
    wheelCount: 4   
}

Plane: {   
    model: "Boeing 777",   
    color: "White",   
    year: 2020,   
    power: 170000,   
    wingspan: 65   
}

Boat: {   
    model: "Bella",   
    color: "Black",   
    year: 2022,   
    power: 100,   
    draft: 0.42   
}

And lastly, console log the created vehicles.

**4. Generic vehicle service**

Create a generic class "VehicleService" which should have following attribute:

* private list with generic type for storing vehicles named "items"

and functions:

* add()
* list()

User should be able to add vehicles to the list, and list all vehicles in the list.  
Create two variables "cars" and "boats" which are both implementation of the generic class with "ICar" and "IBoat" as parameters.  
  
Add the previously created car and boat from the task 3 to the corresponding listings and finally, console log the output when calling the list() function on both of the class implementations.  
  
**Note:**Use "T" as generic type. I.e. coolFunction<T> (niceParam: T) {...}

**Useful documents and links**

* [TypeScript Tooling in 5 minutes](https://www.typescriptlang.org/docs/handbook/typescript-tooling-in-5-minutes.html#handbook-content)
* [TypeScript Interfaces](https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#interfaces)
* [TypeScript Classes](https://www.typescriptlang.org/docs/handbook/2/classes.html#handbook-content)
* [TypeScript Generics](https://www.typescriptlang.org/docs/handbook/2/generics.html#hello-world-of-generics)

Last modified: Friday, 8 November 2024, 12:26 PM