Declaring Types in Untyped Code

This lesson delves into how to declare types on a JavaScript code that does not have a type.

The keyword `declare` can be used before one of the previous three declaration types (`var` , `let` , `const`). As the name suggests, it declares to TypeScript that the variable is somewhere but not saying where. This is not used frequently but can be useful if you need to tell the transpiler that the variable is present, just not in the current project (or loaded module), and may not be visible.

```
1  declare let variableDefinedSomewhereElse: number;
2  let newVariable = variableDefinedSomewhereElse + 1;
```

The official jargon for this is *ambient declaration.* If you are working with JavaScript and there is no definition file, you may want to use `declare` since you know that you will be importing the code with the variable present. Unfortunately, you will lose all Intellisense since `declare` doesn't have a type provided.

Ambient declarations can be defined anywhere. However, a good practice is to have a file with the extension (d.ts). The file does not produce any JavaScript but indicates that the variables or function exists. An example for people who use jQuery is to declare an ambient declaration, for jQuery. jQuery will be added along with the code. Thus, you should assume it's present.

```
1  declare let $: JQuery;
2  export default $
```

It is possible to declare more than just variables. You can define a function as well. In fact, you can declare anything that exists in JavaScript.

```
declare function ambientFunction(i: number):void;

function myFunction(i: number){
    ambientFunction(1);
}
```

A common use case is a library that already exists in JavaScript (but has not migrated to TypeScript), which can be used in TypeScript with a good definition by having a *.d.ts*. It is common to migrate the definition file along with the JavaScript file. If this is not the case, it is possible to download the definition type after you install the JavaScript library.

```
npm install --save apex
npm install --save-dev @types/apex
```

I invite you to take a look at the main repository of definition files on Github.

Back
Declaring a Variable

Next
Hoisting Variables

Completed

Report an Issue