001 TypeScript - declare variables

TypeScript Kata List – Blog Page

TypeScript Kata List – Github Page

# Duration

5 minutes

# Brief

Declaring variables

# Instructions

Get Github folder or the entire katas-typescript repo.

Open the [before/\*.sln] file.

# Github

* Before (start kata with this)
  + <https://github.com/robertdunaway/katas-typescript/tree/master/001%20TypeScript%20-%20declare%20variables>
* After
  + <https://github.com/robertdunaway/katas-typescript/tree/master/001%20TypeScript%20-%20declare%20variables/after>

# Kata

Using the [before/src/app.ts] file do the following:

* Create variables “variableA”, “variableB”, and “variableC” of type number.
* Now set each of the variables to some default values. Possibly 2, 3, and 4.
* Now save the file.
  + If you have the “Task Runner Explorer” tab open then you’ll notice Gulp processing the file just as you save it.
  + If you look at app.ts and app.js in the wwwroot folder you’ll notice that you’re changes suddenly appear there. This is the power of a build tool like Gulp.
  + Not only has Gulp moved your changes to the wwwroot folder but it also created a minified app.js file and an app.js.map file which links your TS file to your JS file within the browser. This enables you to set debug breakpoints on your TS file even though your browser has no idea what a TS file is.

At this point your code might look like the following:

// Let's get started!

console.log("Let's get started!");

var variableA: number = 2;

var variableB: number = 3;

var variableC: number = 4;

* You can also create new variables where its type must be inferred.

var inferredvariable = 5;

* You can also create new variables where the type is defined but with no initial value.

var noInitialValue: number;

* Write out your values to the console.

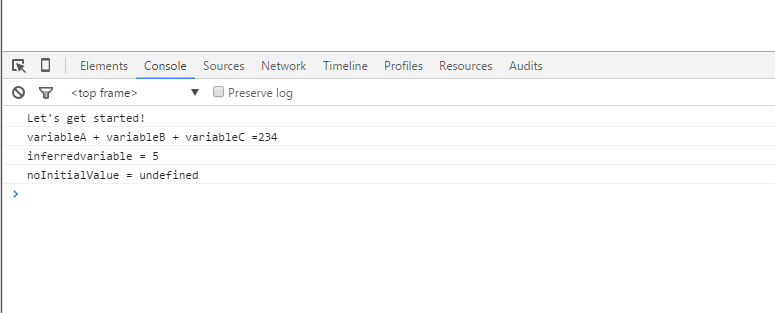
console.log("variableA + variableB + variableC =" + variableA + variableB + variableC);

console.log("inferredvariable = " + inferredvariable);

console.log("noInitialValue = " + noInitialValue);

* Run your project and when the web page loads press F12.
* Select the Console window and you should see the following, in Chrome.

Console Window



Select the “Sources” tab to view any file the browser has downloaded. In this case the TypeScript file we’ve been using is selected with a breakpoint at line 15. The browser has no idea what TypeScript is but because we have also provided a “\*.map” file the browser is able to break in the TypeScript file.

TypeScript file with a breakpoint

