Cheat Sheet – Debugging

Console.log

A very quick and easy way to start with debugging is by using console.log(). Whilst this doesn't replace "real debugging", it makes it easy to see if some variables are set as expected, certain parts in the code are reached or which status/ value a certain property has at a certain place during code execution.
It may be used like this:

```
let myExampleVariable = 123;

console.log(myExampleVariable);
```

Note, that console.log may not only log variables but also objects or arrays.

Browser Debugging & Source-Maps

Using console.log is great, to quickly check on some values. However, sometimes debugging with it very cumbersome or not possible at all. The great thing is, that you can use sourcemaps in your compilation step. What are sourcemaps? Sourcemaps allow the browser (which runs JS, not TypeScript!) to re-build the original TypeScript code/ find the original code files.
Therefore, you're able to place breakpoints in the TypeScript code and evaluate your TypeScript code at runtime.
Where can you do this? For example, the developer tools Chrome offers allow you to dig into your source/ TypeScript code ("sources" tab).

Augury

Augury (https://github.com/rangle/augury) is a Chrome extension you may install. It allows you to not only dig into your code (as with sourcemaps) but instead to get a more "understandable" representation of your components, their state etc. Definitely check out the official documentation to learn more about it!