**21/06/2024**

**Typescript Tech Lesson 2 – Functions and Testing!!**

We will be doing a demo from other people’s computers

**Implicit and Explicit Types:**

* Create an addTwoNumbers function

*export const addTwoNumbers = (num1:number, num2:number) => num1 + num2;*

*export function addTwoNumbers(num1:number, num2:number) (return num1 + num2);*

*= implicit return type*

*export function addTwoNumbers(num1:number, num2:number) : number(return num1 + num2);*

*= explicit return type*

*export –* Visibility modifier that will allow the function to be visible outside of the origin file.

export default- Visibility modifier for if your file only has one component (older way of doing this)

* More complex version of the addTwoNumbers function

*export function addNumbers2(num1:number, num2:number) {*

*return num1 === 7 ? ‘Hello’ : num1+num2;*

*}*

=== - Non-truthy version of equals in if statements in Typescript. Means the input has to be EXACTLY what is on the right of the sign!

**If statement => Ternary Operator:**

*if (num1 === 7){*

*return ‘Hello’;*

*} else {*

*return num1+ num2;}*

*becomes:*

*num1 === 7 ? ‘Hello’ : num1+num2;*

**Writing Tests**

These are good because it allows you to figure out if cutting out certain lines of code will actually break the code or not.

The steps are as follows:

1. **Create file in format ‘[name of corresponding file you’re testing].test.ts’**
2. **Import the file that you want to test**
3. **Write the test using the following syntax:**

*test(“Add two numbers together”, () => {*

*const result = addTwoNumbers(4,5);*

*expect (result).toEqual(9);*

*});*

1. **Write ‘npm test’ into terminal – this will run jest!**
2. **Write ‘npm test -- --coverage’ to get the test coverage!**