



# TypeScript Learning - Session 2

Presented by Umang Bhatt



# Today's content is from:

- TypeScript Essentials
- Pro TypeScript
- TypeScript Revealed
- Learning TypeScript - 2<sup>nd</sup> edition
- Mozilla developer network

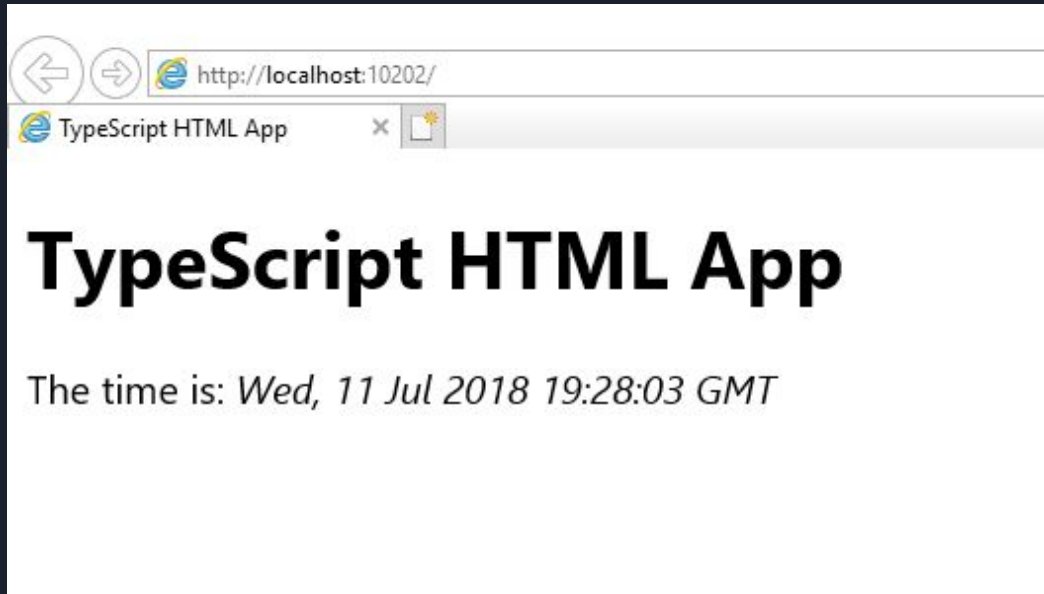
Let us get started



Start with visual studio



# Exercise 1 - Create html typescript app from template

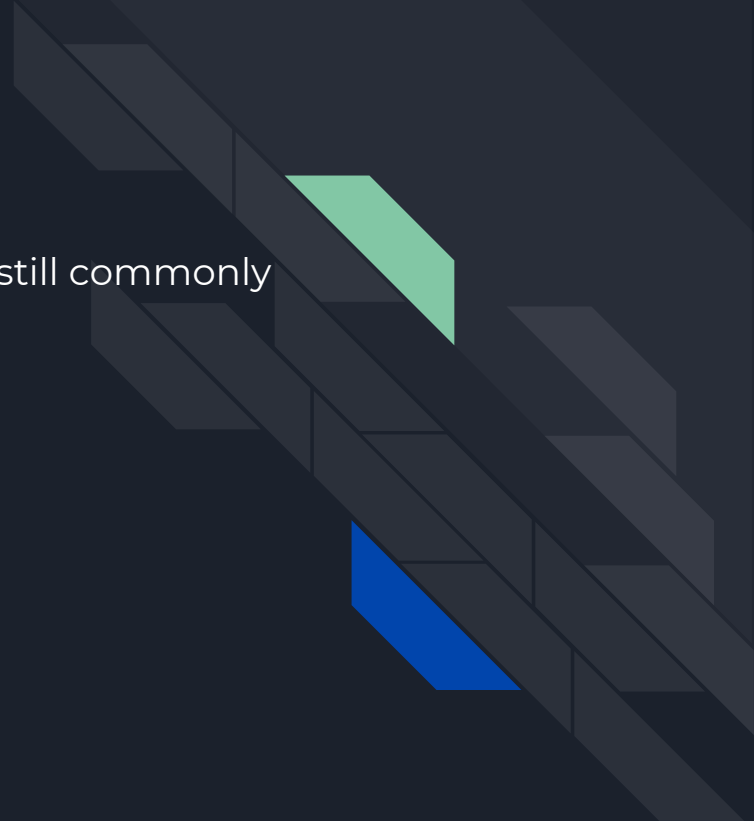




Who can find “do not emit on error”  
for this project ?

# Why is there no “JavaScript” on properties page ?

JavaScript is now formally known as ECMAScript but it is still commonly referred to as its original name.



# Home work

## Exercise 2 - Exception to “Every valid JavaScript file is a valid TypeScript file”

“With” and other vendor specific  
Implementations like Mozilla’s const

```
C:\js_learning\2nd day\Exercise 2\TS Example>tsc a.ts
a.ts(4,1): error TS2410: The 'with' statement is not supported. All symbols in a 'with' block will have type 'any'.
```



## Exercise 3

- Type inference
- Learn to mouse hover

Copy code from exercise file and paste into app.ts file in visual studio

```
28  f,  
29  
30  let counter; // unknown any type  
31  let counter1 = 0;  
32  let counter2: number;  
33  let counter3: number = 0;
```

[🔍] let counter3: number

# Exercise 4

## optional type inference

This is just previous exercise with same code,  
different understanding

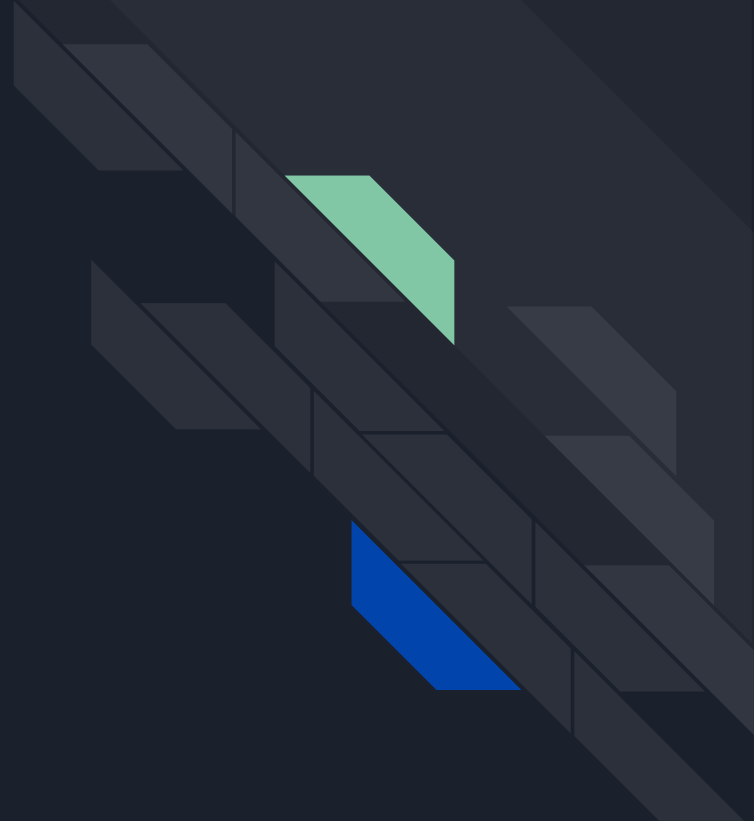


# Basic types



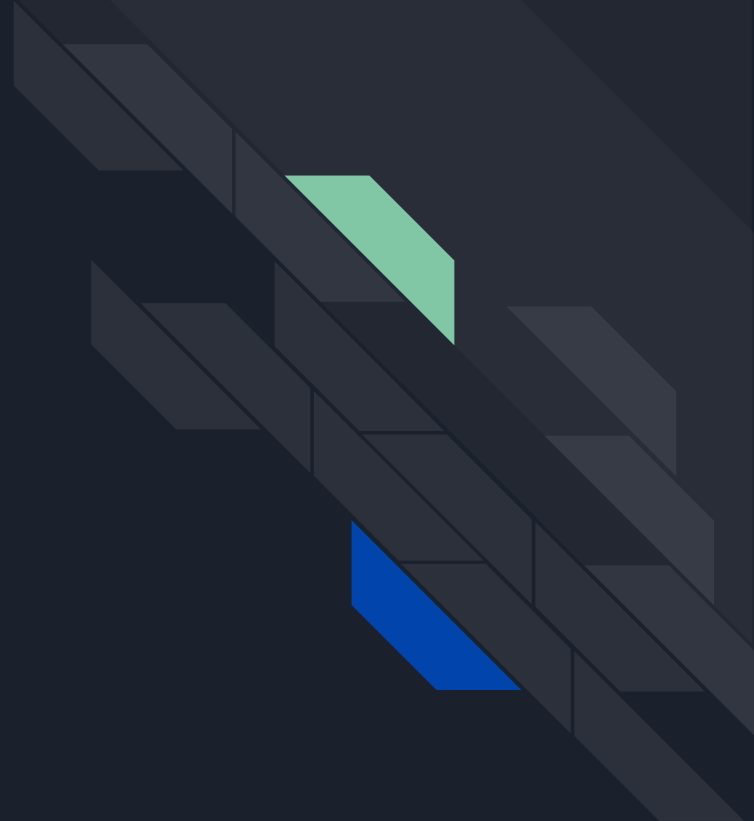
# Exercise 5 - boolean

Copy code from file to visual studio



# Exercise 6 - nullable boolean

Copy code from file to visual studio



# Exercise 7 - number Float

Copy paste the code from file to visual studio.

All numbers in EcmaScript are floats. Here is a [link](#).

Fix the compile time error.





# Exercise 8

## Special case in JavaScript - division by 0

Copy code from exercise folder to visual studio.

# Home work

Do following :

- Create a program checking if a variable is NaN.
- A program to check if a variable is positive infinity.
- A program to check if a variable is negative infinity.
- A program to check if a negative zero and positive zero are present and if a comparison can be made.

First person to share programs over email will be rewarded in next session in presence of everyone.





Strings



# Exercise 9 - Strings - single quote - double quote

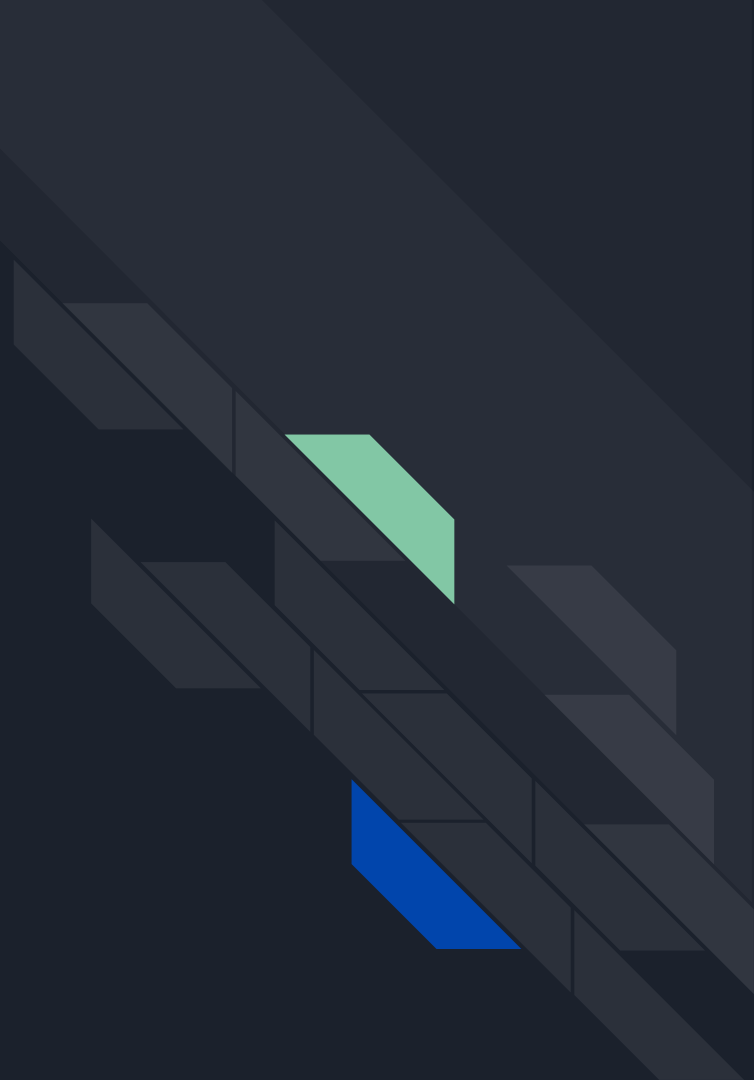
Copy code from exercise folder to visual studio.



# Home work

## Exercise 10 - no distinction between single and double quote

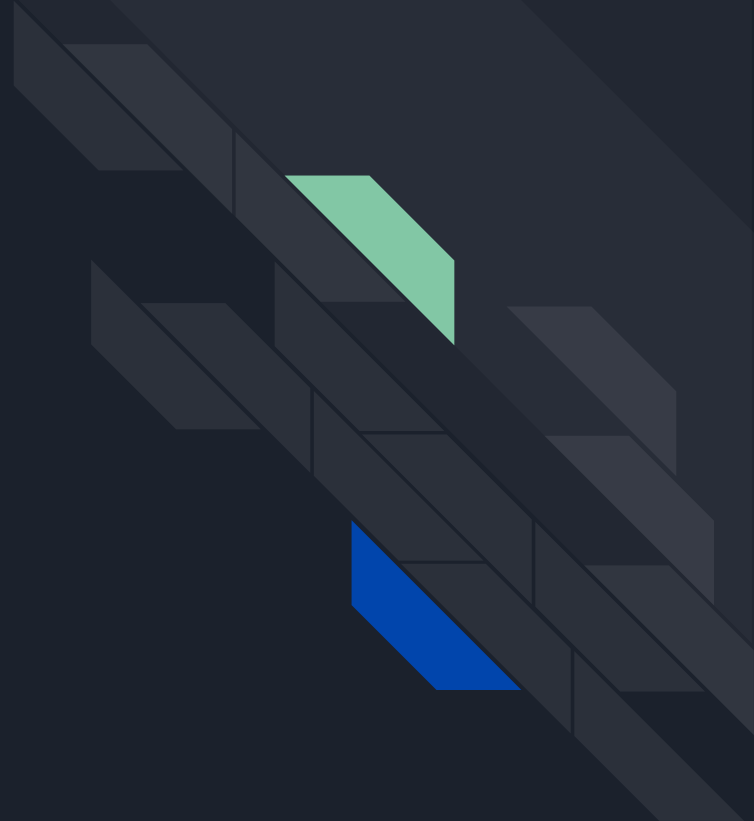
Make comparison using `==` for two similar string declared using  
single quote and double quote



# Home work

## Exercise 11 - long strings

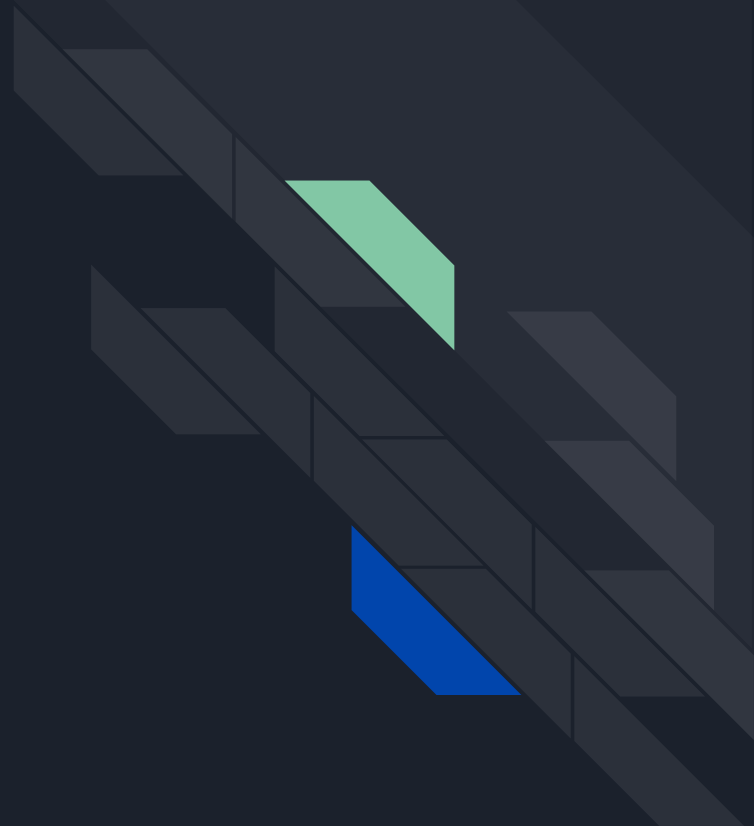
Copy code from exercise folder to visual studio.



# Home work

## Exercise 12 - string character access

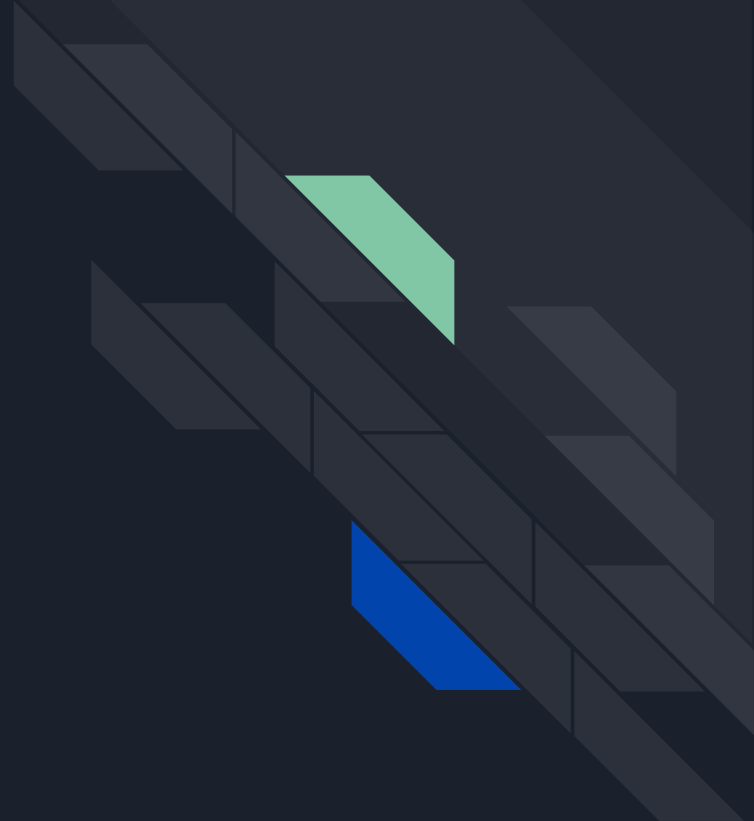
Mouse hover on piece variable in debugging and  
see variable type of it.



# Home work

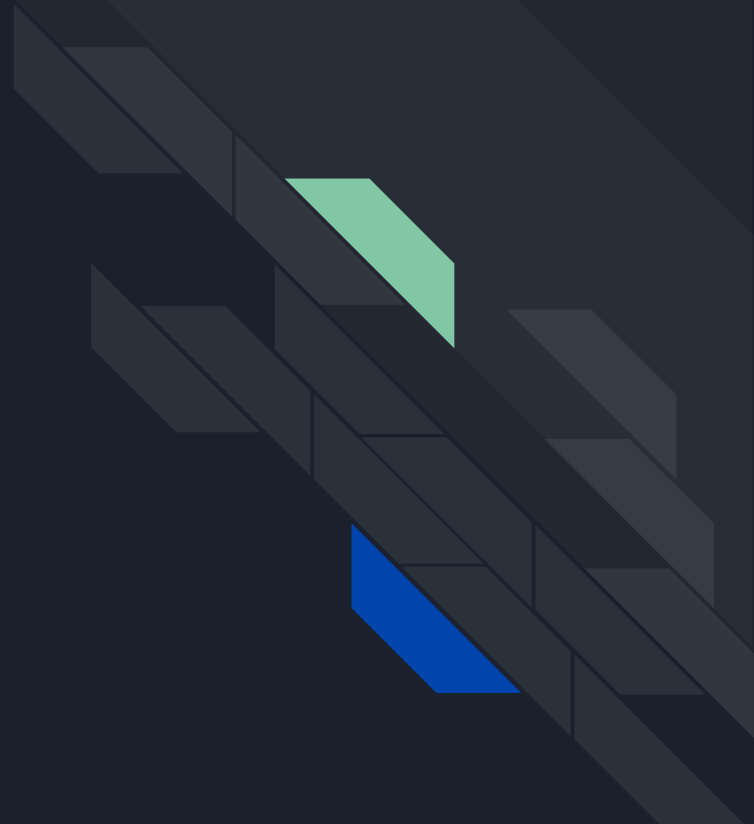
## Exercise 13 - strcmp

Strings can be compared using > and <  
Copy code from exercise folder to visual studio.



# Exercise 14 - arrays

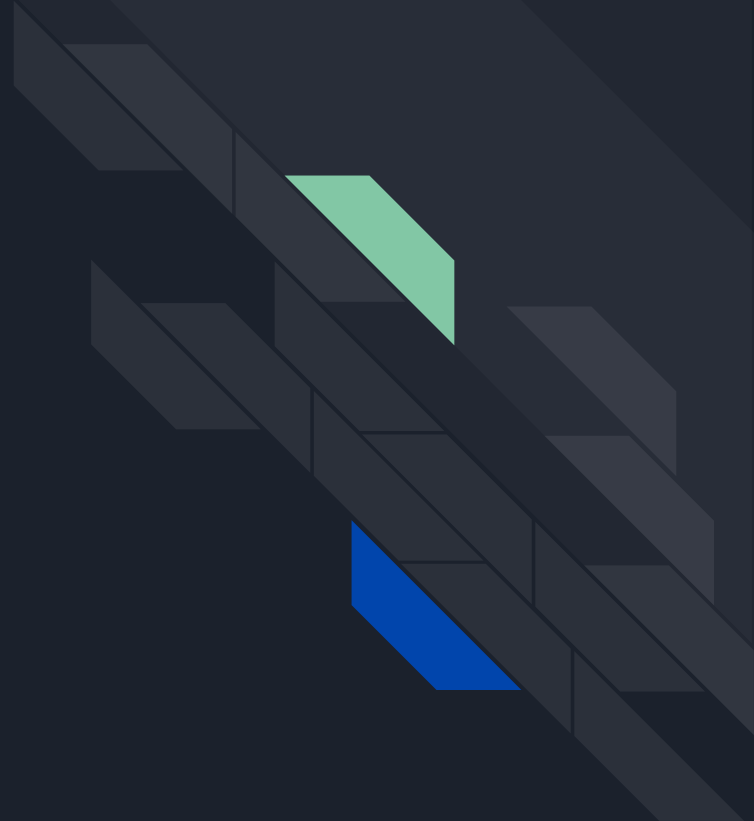
Two ways to declare arrays. Copy code from exercise folder to visual studio.



# Home work

## Exercise 15 - arrays

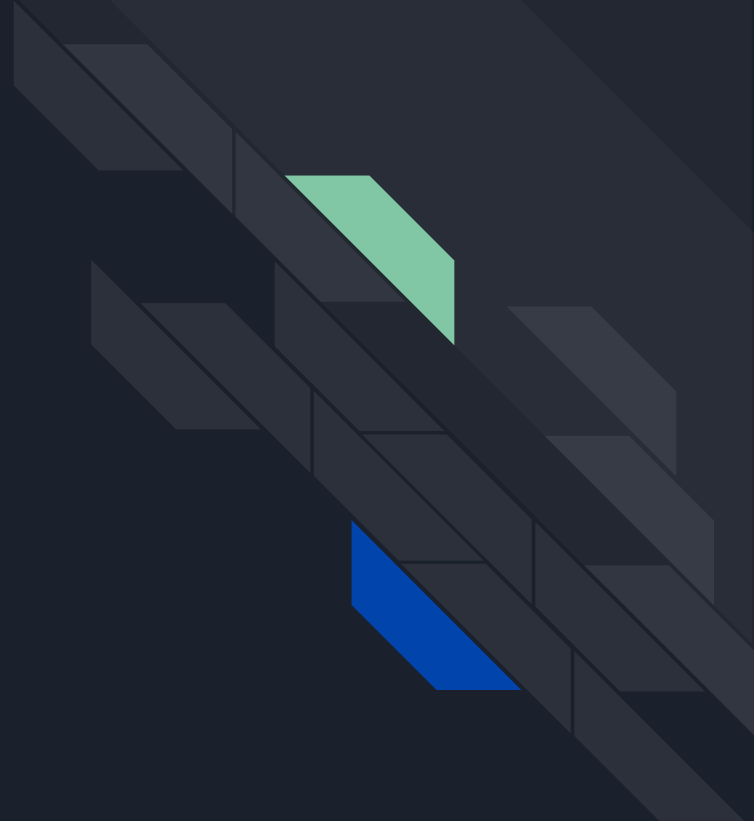
Look various examples of methods on array at  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)





# Exercise 16 - tuples

Copy code from exercise folder to visual studio.



# Exercise 17 - Enums

Copy code from exercise folder to visual studio.  
Remember that there is no enum in javascript.  
Let us look at the code in source.

First one to figure out how to see corresponding js file  
from visual studio gets reward.

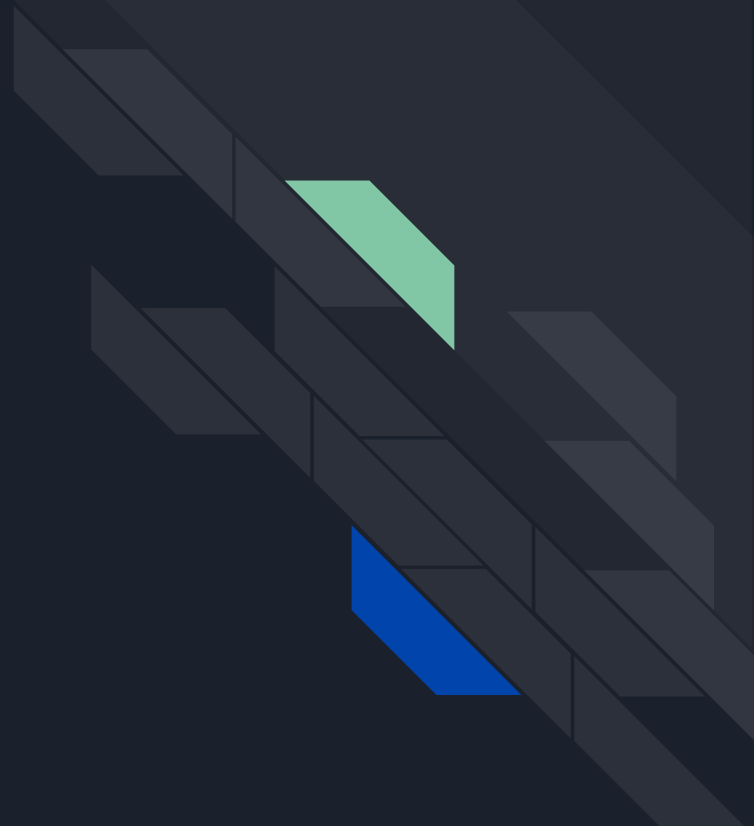


# Exercise 18 - any

All types in TypeScript are subtypes of a single top type called the any type.

The any type can be useful while migrating existing JavaScript code to TypeScript

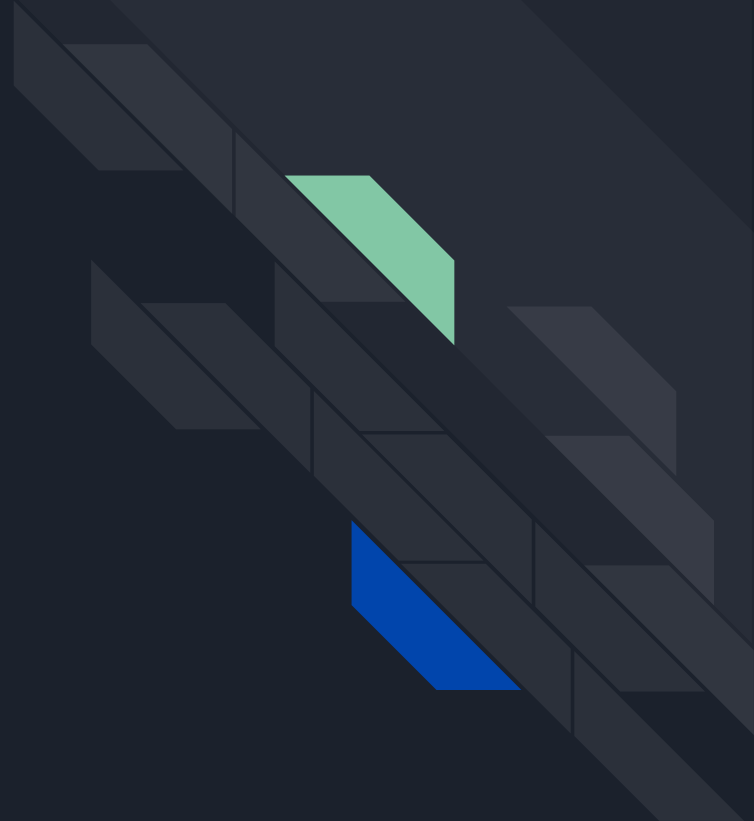
The any type eliminates most of the TypeScript type checks and represents all the possible types



# Exercise 19 - null and undefined

In TypeScript, both undefined and null are types. By default, null and undefined are subtypes of all other types. That means you can assign null and undefined to something like a number.

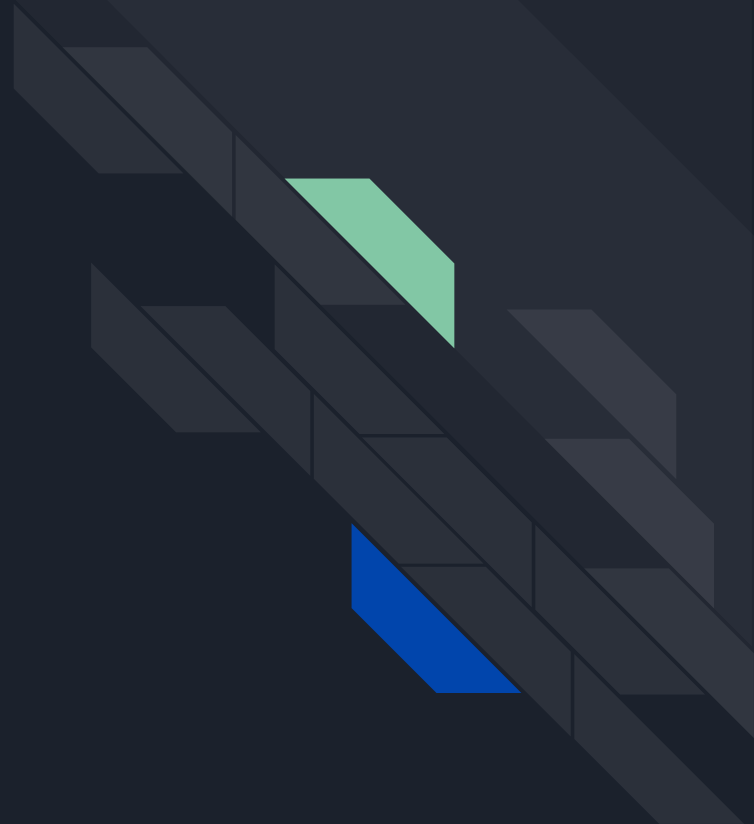
When you declare a variable by having a var a statement in a block, but haven't yet assigned a value to it, it is undefined



## Exercise 20 - void

In TypeScript, both undefined and null are types. By default, null and undefined are subtypes of all other types. That means you can assign null and undefined to something like a number.

When you declare a variable by having a var a statement in a block, but haven't yet assigned a value to it, it is undefined



# Exercise 21 - var, let and const

Variables declared with `var` are scoped to the nearest function block (or global, if outside a function block).

Variables declared with `let` are scoped to the nearest enclosing block (or global, if outside any block), which can be smaller than a function block.

The `const` keyword creates a constant that can be global or local to the block in which it is declared. This means that constants are block-scoped.

discussion?



Thank you.

