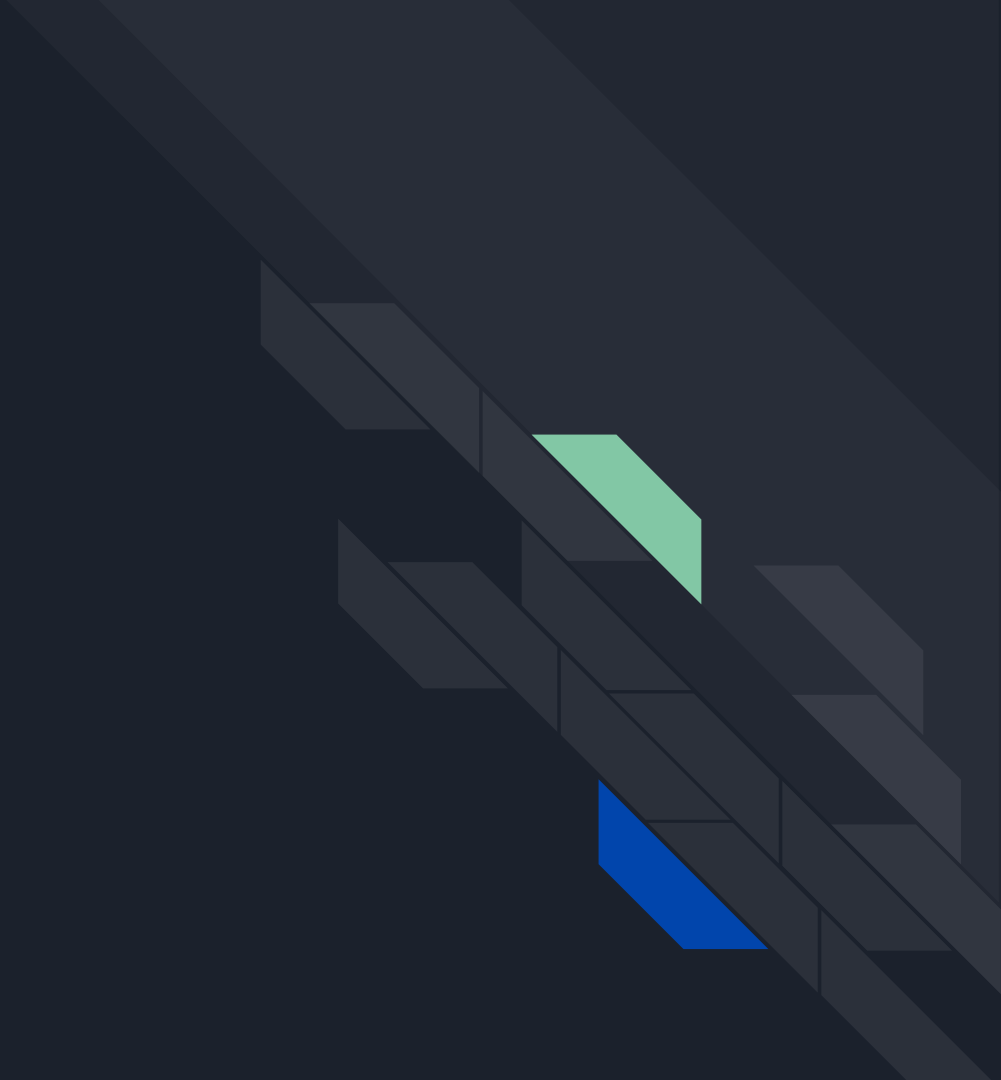


# Template literals

Manasseh



# Template literals

Description: Template literals are string literals allowing embedded expressions . You can use multi-line strings and string interpolation features with them. (They were called string templates in prior editions of ES2015)

Template literals uses the backtick (``` -located underneath the esc key) to do different things when formulating strings.

E.g Multiple line capabilities

In es5 `\n` would be used to make multiple lines:

Let name = "Manasseh works at \n The National Archives"

> "Manasseh works at  
The National Archives"

In es6 Template literals uses the backtick to allow for multiple lines to be made :

Let name = `Manasseh works at  
The National  
Archives`

> "Manasseh works at  
The National  
Archives"



# No more Concatenation...

In es5 you would use the following method of concatenation to add strings and variables together:

```
Var firstName = "Manasseh"  
Var lastName = "Boyd"  
console.log ("My name is " + firstName + " " + lastName)  
> "My name is Manasseh Boyd"
```

In es6 we'll use let instead of var and use place holders which are indicated by a dollar sign `$` and curly braces `{ }` instead of the plus sign :

```
let firstName = `Manasseh`  
let lastName = `Boyd`  
console.log ("My name is " + ${ firstName } + ${ lastName})  
> "My name is Manasseh Boyd"
```



# Objects with template literals

Example with Objects in es6:

```
Let person = {  
  firstname: 'Manasseh',  
  lastName: 'Boyd',  
  SayName () {  
    return `My name is ${this.firstName} ${this.lastName}`;  
  }  
}
```

```
Let name = person.sayName ();
```

```
Console.log (name);
```

"My name is Manasseh Boyd"