JS

var	global + changeable
let	local + changeable
const	local + unchangeable

Functions

∴ react : fat arrow functions

- Named function.

```
function    func_name() {

};

func_name();
```

∴ invoke

- Anonymas function

```
         ↙no name
function   () {

};
```

how to invoke?

```
let   variable_name  =  anonymous function

variable_name() ;
```

∴ invoke

- Fat Arrow function (recommended)

assigning function to a variable is called function expression

```
↙no keyword + name
() => {

};
```

How to invoke?

```
let   variable_name  =  fat arrow function

variable_name() ;
```

∴ invoke

Template String.

∴ formatted string.

```
` message ___ ${variable_name} `
```

Operators

All operators + one extra.

Ternary Operator ?

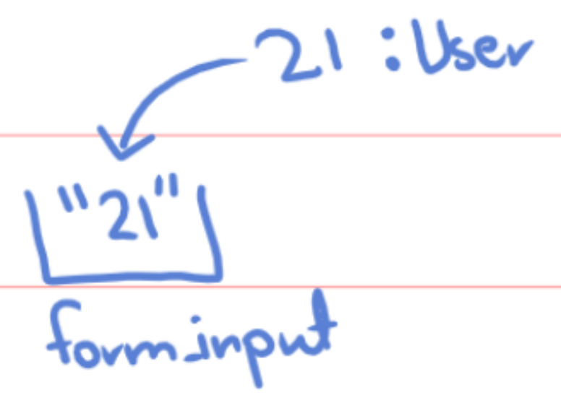
```
let  variable_name  =  (condition) ? ____ : ____ ;
                                     true   false
                                   ↓
                          Ternary Operator
```

Example:

```
let  num = 4;
let  check =  num==4 ? "it is 4" : "it's not 4";
console.log(check)
```

- Assign =
- Compare ==
- ? ===

Form input func take each value as string.
So if user enter number 21 it is stored as "21"



```
Let val = form_input

If (val == 21) : False
If (val === 21) : True
```
↳ this auto checks

- Arrays

```
let array_name = [];
```
∴ single array can hold all types of data
int, float, string, etc

∴ 0 base indexing

- Array Objects

```
let array_name = [
        { key : value , key : value },
        { key : value , key : value }
    ]
```
∴ to access particular key of object we use dot operator

- For loop:

```
for ( start point ; condition ; increment )
{

}
```

For loop & For Each are exactly same

For Each returns undefined
Map returns array.

- For Each Loop: Recommended react

```
array_name .forEach()

array_name . forEach( fat arrow func )

array_name.forEach( () => {  })

                   ↓ iterator
array_name . for Each ( (std) => { console. log(std) })

Let variable = array_name . for Each ( (std) => { console. log(std) })
console. log ( variable )
```
∴ return undefine_value

- Map [Transformer]

array_name • map ()

array_name • map (fat arrow func)

array_name • map (() => { })

iterator

array_name • map ((std) => { console.log(std) })

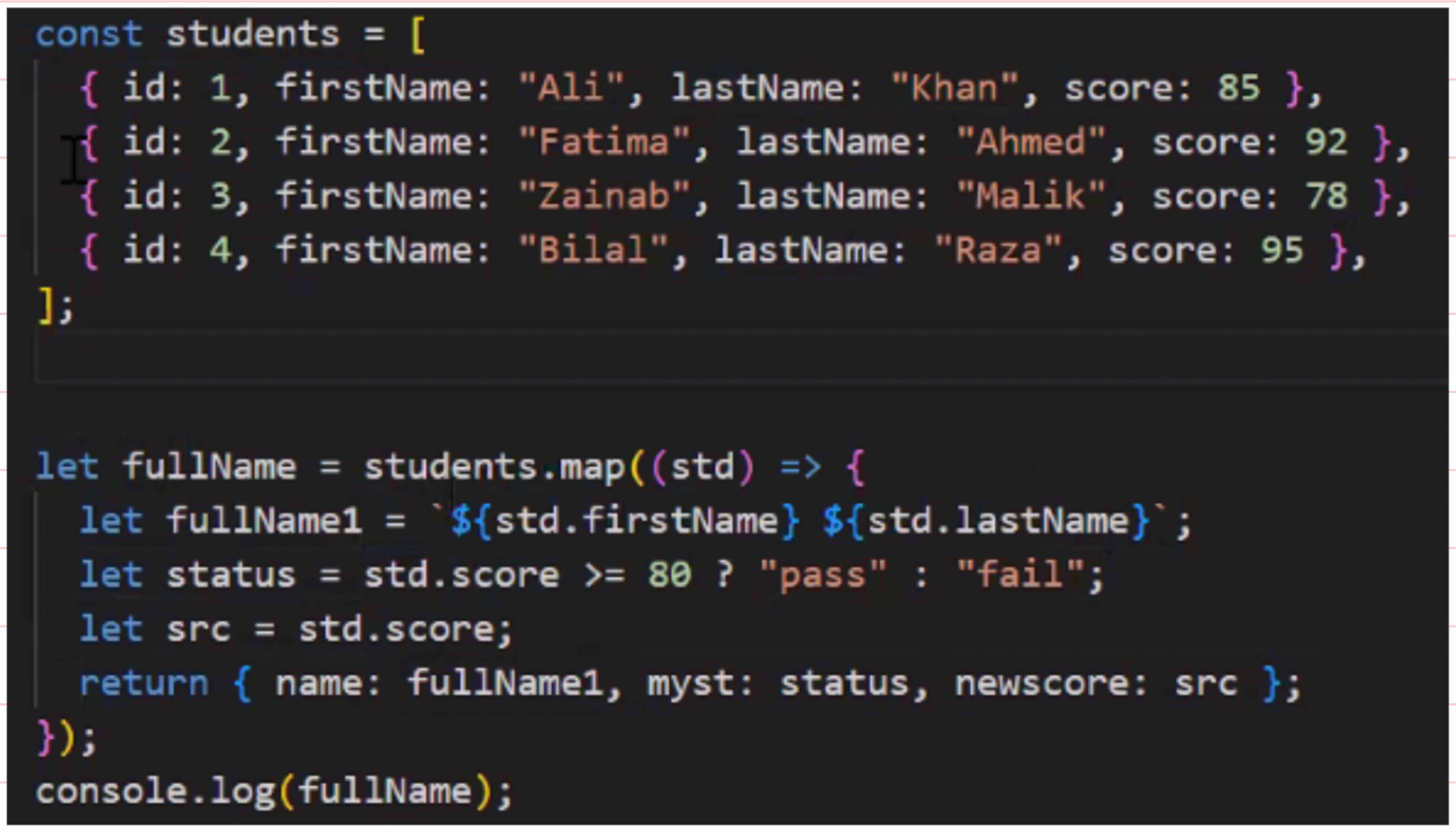
let variable = array_name • map ((std) => { console.log(std) })

console.log (variable)

∴ return undefined_value_array or new_transformed_array.

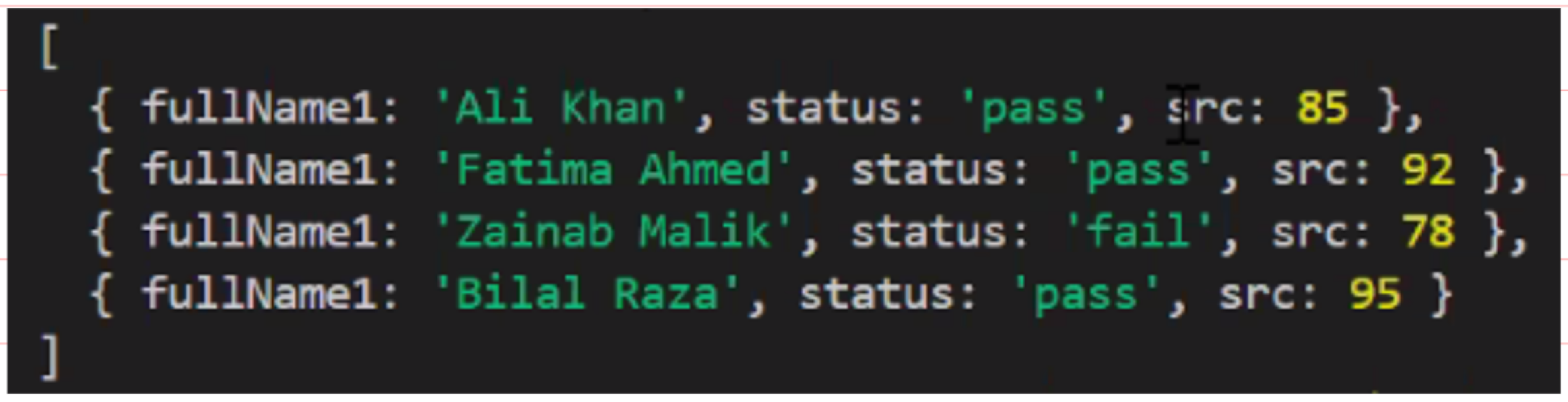
- ForEach returns undefine.
- Map returns array

∴ also use to display data on web we use map().



```
const students = [
  { id: 1, firstName: "Ali", lastName: "Khan", score: 85 },
  { id: 2, firstName: "Fatima", lastName: "Ahmed", score: 92 },
  { id: 3, firstName: "Zainab", lastName: "Malik", score: 78 },
  { id: 4, firstName: "Bilal", lastName: "Raza", score: 95 },
];


let fullName = students.map((std) => {
  let fullName1 = `${std.firstName} ${std.lastName}`;
  let status = std.score >= 80 ? "pass" : "fail";
  let src = std.score;
  return { name: fullName1, myst: status, newscore: src };
});
console.log(fullName);
```



```
[
  { fullName1: 'Ali Khan', status: 'pass', src: 85 },
  { fullName1: 'Fatima Ahmed', status: 'pass', src: 92 },
  { fullName1: 'Zainab Malik', status: 'fail', src: 78 },
  { fullName1: 'Bilal Raza', status: 'pass', src: 95 }
]
```