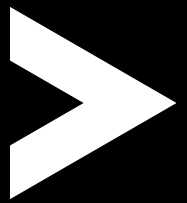




TODAY'S LEARNING

STOP WRITING LOOPS



(**IMPERATIVE** vs. **DECLARATIVE**)

SWIPE TO SEE THE CODE





THE CONCEPT

THE "TAXI" ANALOGY 

IMPERATIVE

YOU FOCUS ON THE PROCESS

TURN LEFT. DRIVE 200M. TURN RIGHT.
STOP AT THE LIGHT. DRIVE 5KM

LINKEDIN.COM/IN/MRSYEDMUZAMMIL
GITHUB.COM/MRSYEDMUZAMMIL



THE CONCEPT

THE "TAXI" ANALOGY 

DECLARATIVE

YOU FOCUS ON THE RESULT

TAKE ME TO THE AIRPORT.

BOTTOM NOTE: DECLARATIVE CODE HIDES THE MESSY DETAILS

LINKEDIN.COM/IN/MRSYEDMUZAMMIL
GITHUB.COM/MRSYEDMUZAMMIL



KEY ASPECTS

READABILITY

Code reads like English.

NOSIDE EFFECTS


Imperative relies on mutating variables. Declarative returns new data (Immutability).

REACT STANDARD

Modern UI is $UI = f(state)$. You declare what the UI looks like; you don't manually update the DOM.

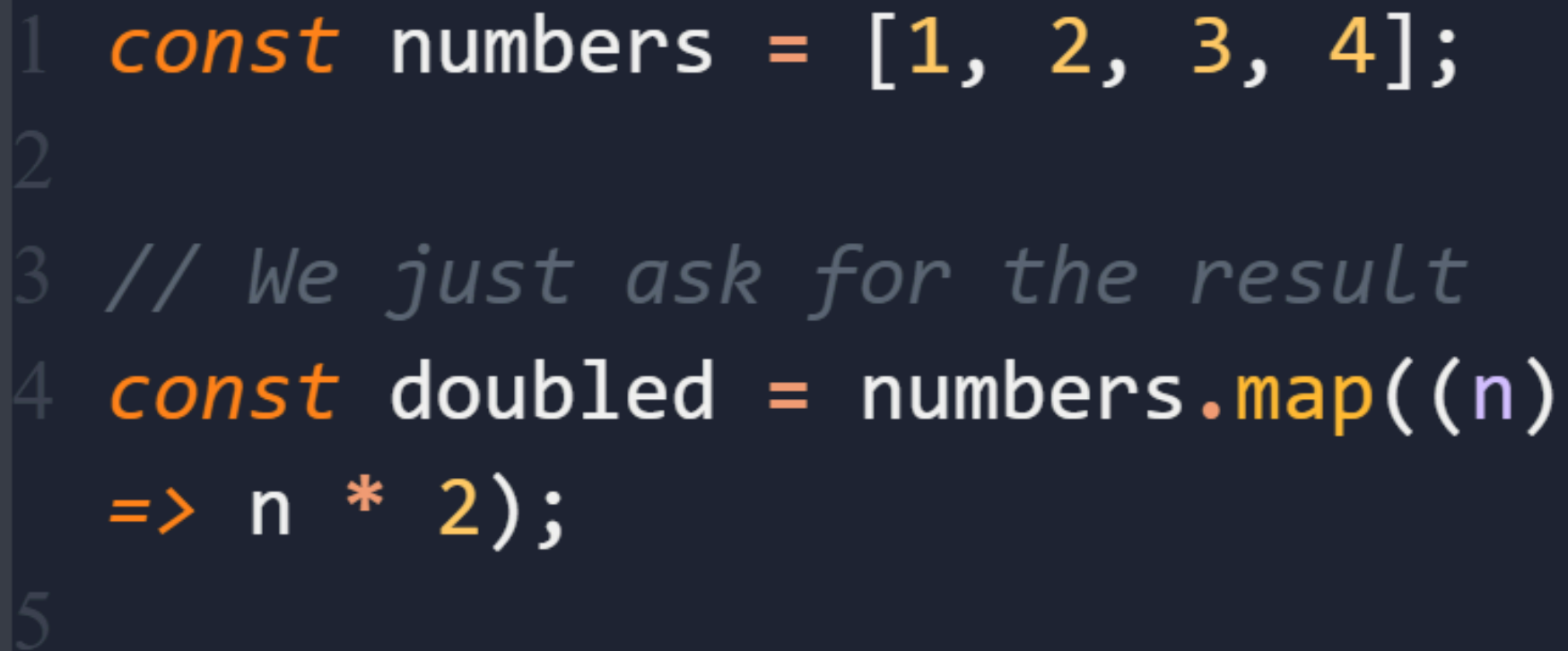
[LINKEDIN.COM/IN/MRSYEDMUZAMMIL](https://www.linkedin.com/in/mrsyedmuzammil)
[GITHUB.COM/MRSYEDMUZAMMIL](https://github.com/mrsyedmuzammil)

IN CODE



```
1  const numbers = [1, 2, 3, 4];
2  let doubled = [];
   // External State
3
4  // We manage the loop manually
5  for (let i = 0; i < numbers.
   length; i++) {
6      doubled.push(numbers[i] * 2);
7  }
8
```

IN CODE



```
1  const numbers = [1, 2, 3, 4];  
2  
3  // We just ask for the result  
4  const doubled = numbers.map((n)  
    => n * 2);  
5
```

[LINKEDIN.COM/IN/MRSYEDMUZAMMIL](https://www.linkedin.com/in/mrsyedmuzammil)
[GITHUB.COM/MRSYEDMUZAMMIL](https://github.com/mrsyedmuzammil)