
Visit the Wondrous
FOREST OF FUNCTION EXPRESSIONS

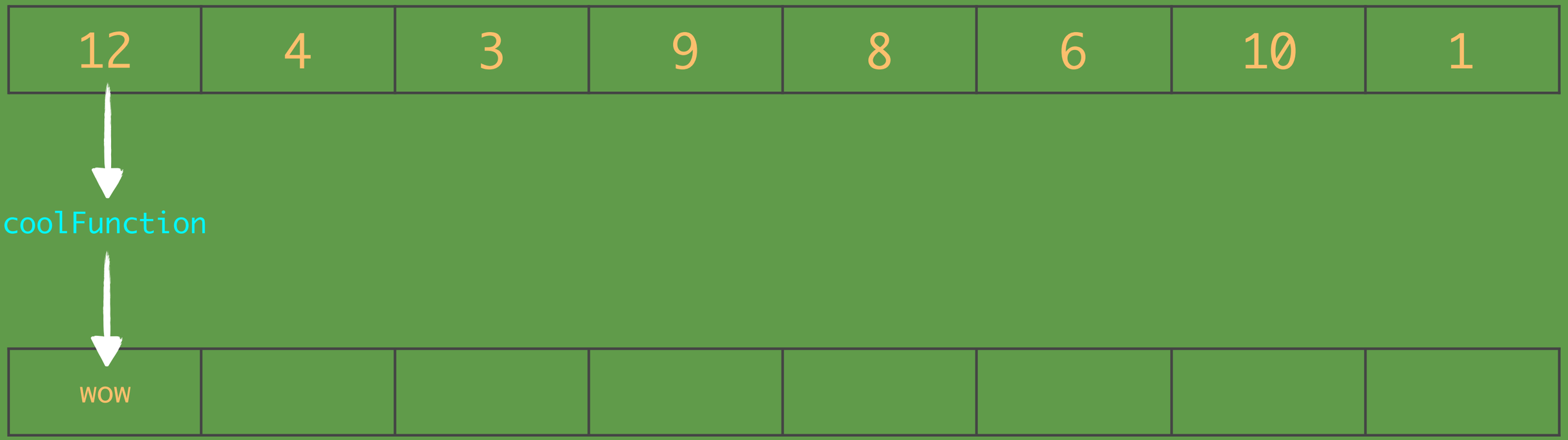
USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

The map( ) method will always take in a function as a parameter, and return a new array with the results.

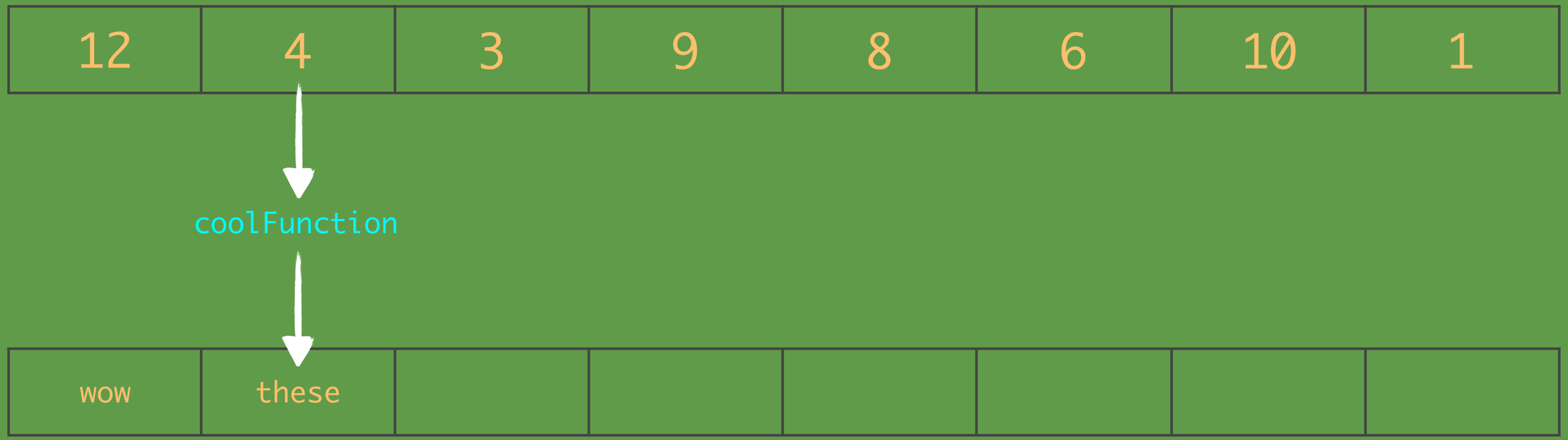


USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

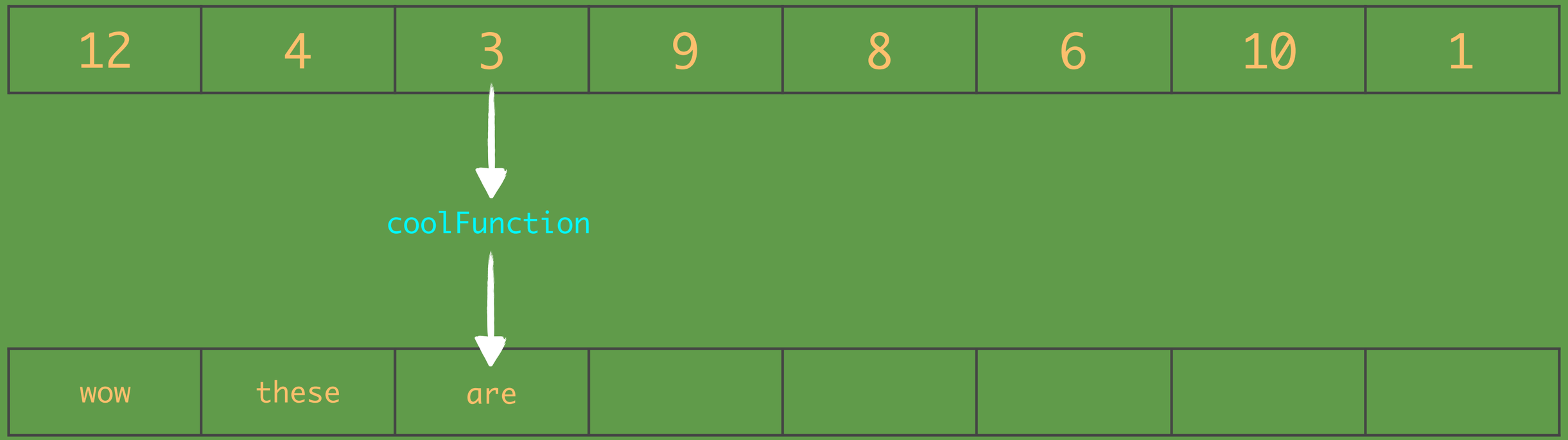


USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

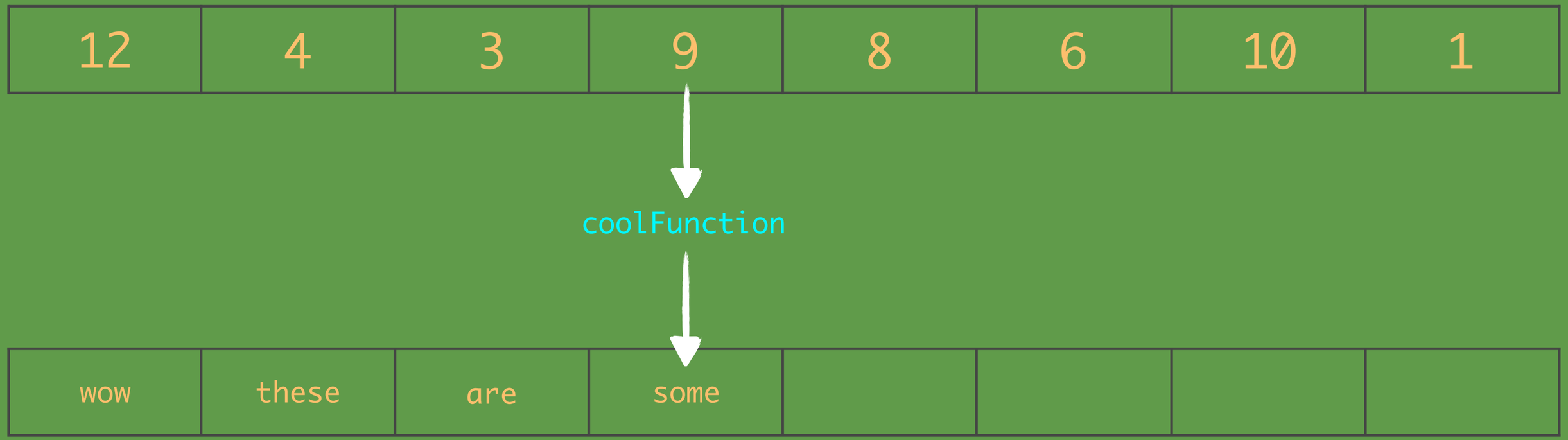


USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

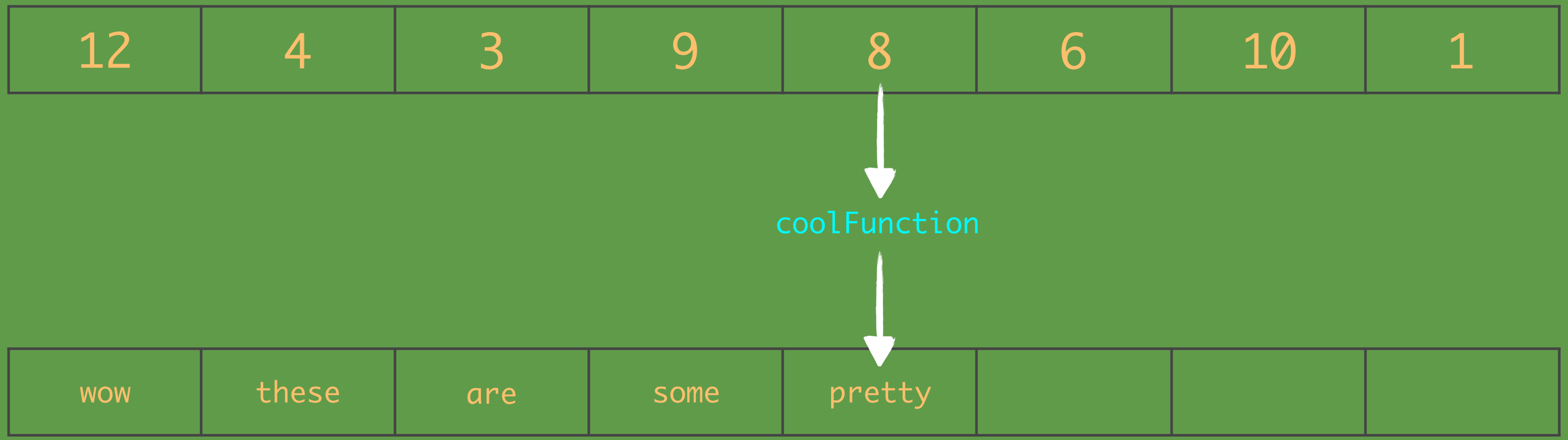


USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```



USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

USING FE'S WITH ARRAYS AND MAP()

A function expression is just that...an expression. We can pass them without variables!

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

USING FE'S WITH ARRAYS AND MAP()

Map works like a loop that applies a function to each array index

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

```
var results = [ ];
for(var i = 0; i < numbers.length; i++){

    results[i] =

}
```

USING FE'S WITH ARRAYS AND MAP()

Map works like a loop that applies a function to each array index

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( *some coolFunction goes here* );
```

```
var results = [ ];
for(var i = 0; i < numbers.length; i++){

    results[i] = coolFunction(numbers[i]);

}
```

The array's map conveniently takes this entire loop format and consolidates it to one nice line of code.

USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

We build an anonymous function for map's parameter, which takes in the contents of each cell of numbers and returns a doubled value to results.

12	4	3	9	8	6	10	1

USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                                return arrayCell * 2;
                            }
);
```

Don't forget to close both your anonymous function with a } and the map method with a), while also adding a semicolon in order to execute the map.

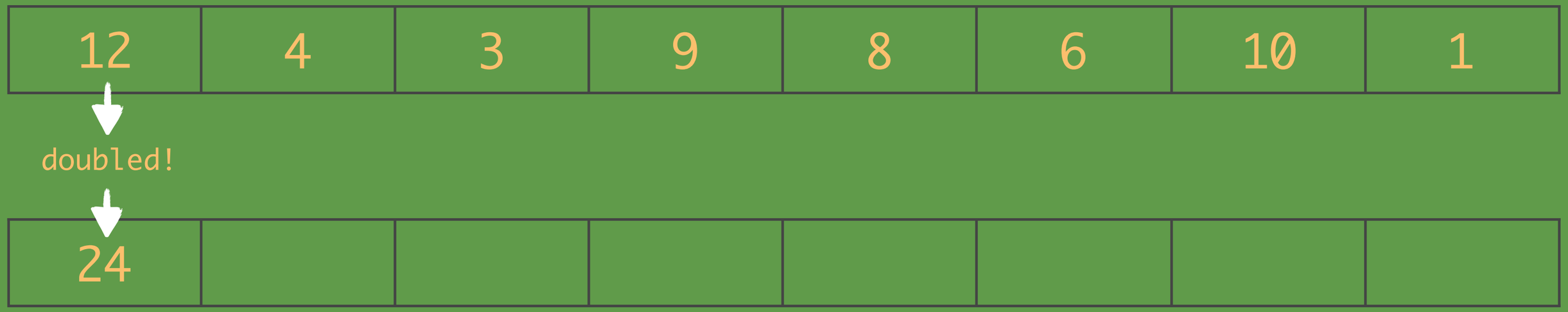
12	4	3	9	8	6	10	1

USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

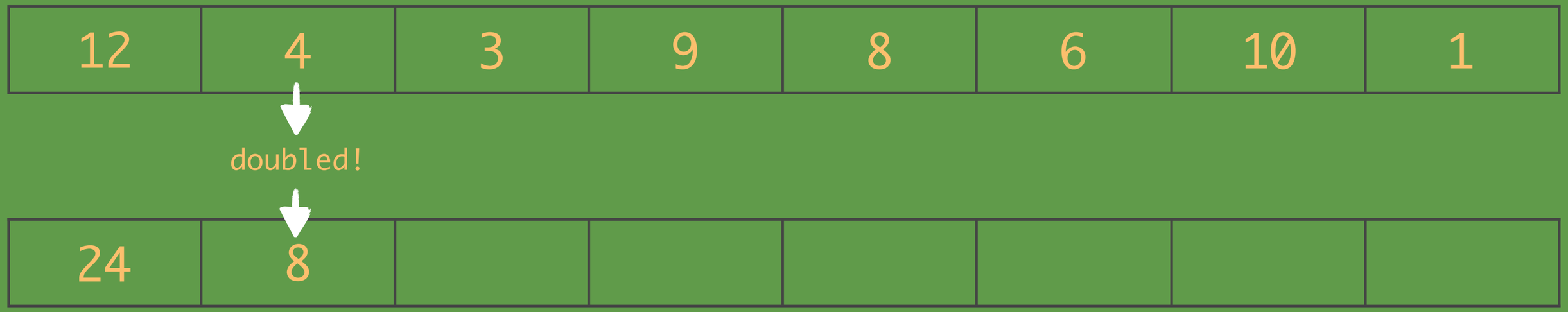


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

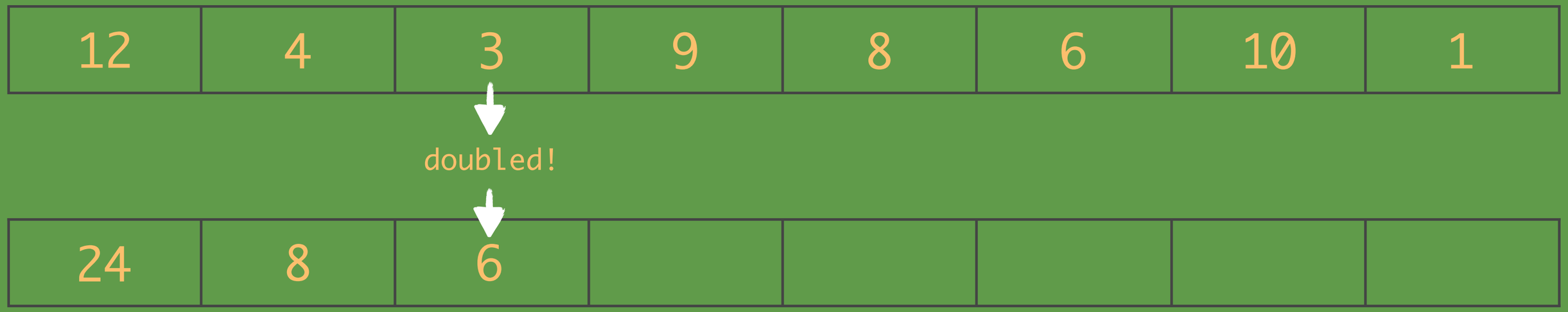


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

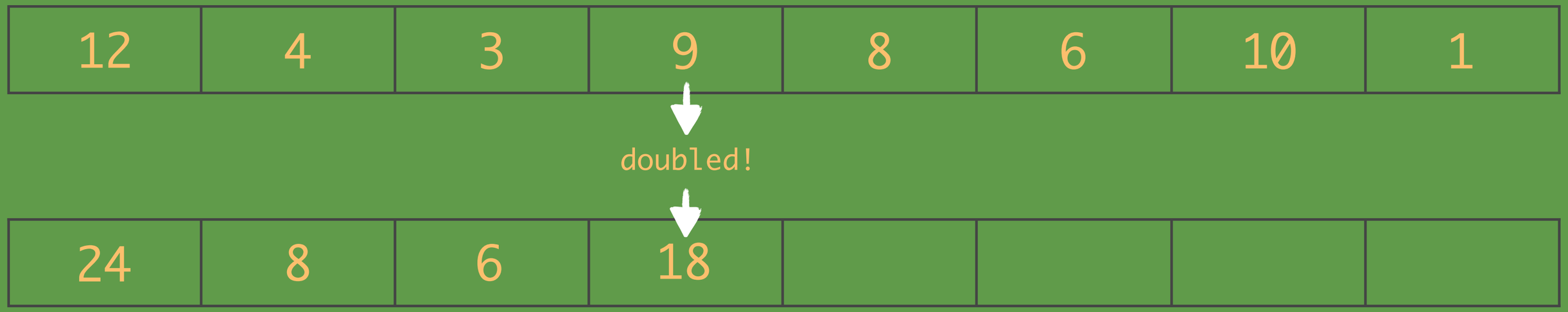


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

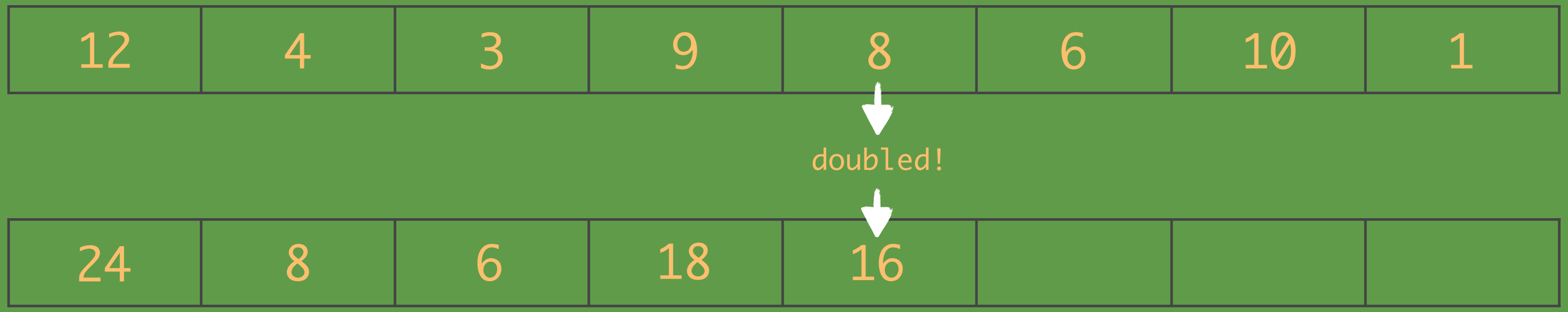


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

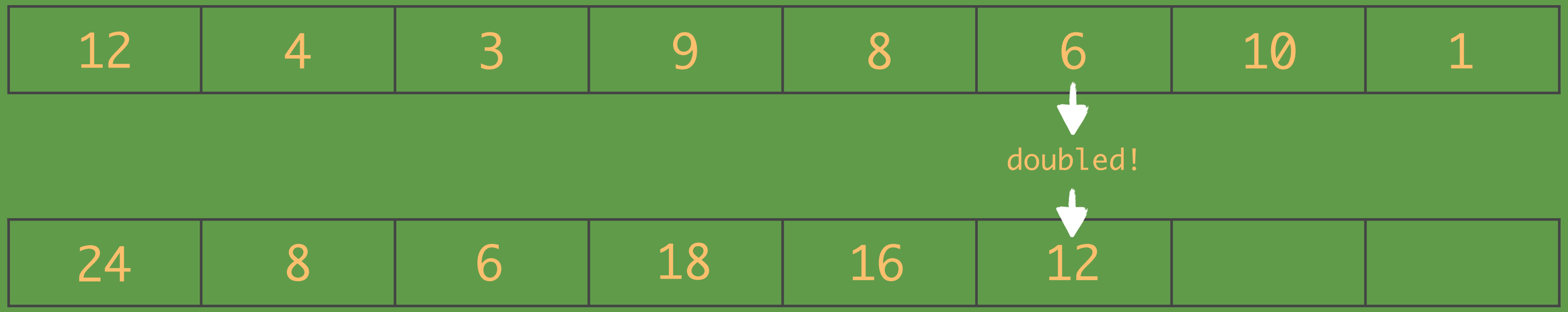


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

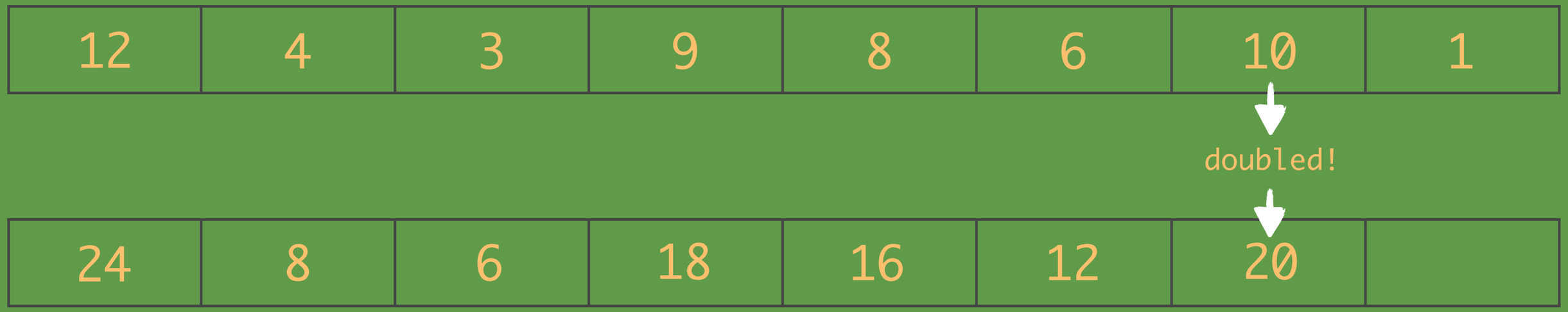


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

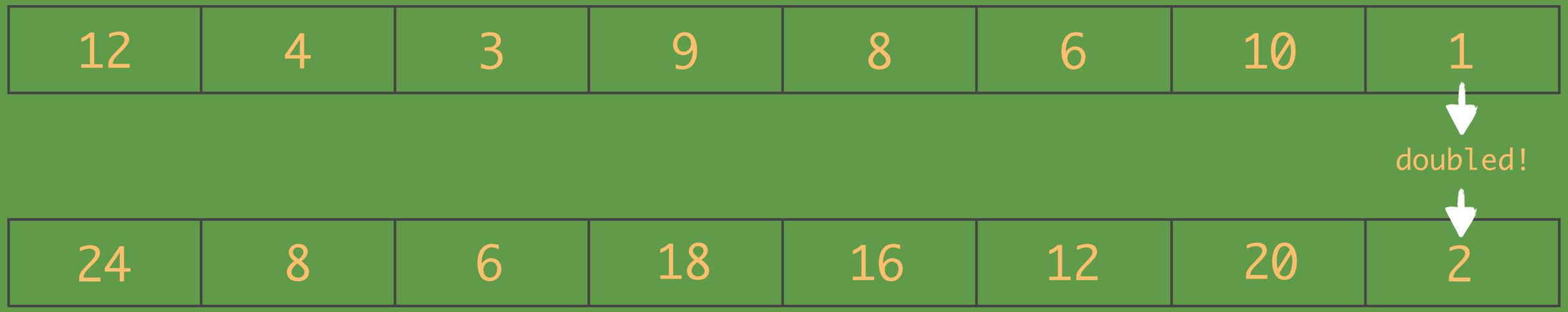


USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```



USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) {
                              return arrayCell * 2;
                           }
);
```

```
console.log(results);
```

→ `[24, 8, 6, 18, 16, 12, 20, 2]`

USING FE'S WITH ARRAYS AND MAP()

Let's pass in function that will double each cell's value in our numbers array.

```
var numbers = [12, 4, 3, 9, 8, 6, 10, 1];
```

```
var results = numbers.map( function (arrayCell) { return arrayCell * 2; } );
```

Short functions are often built in one line for clarity and simplicity.

```
console.log(results);
```

→ [24, 8, 6, 18, 16, 12, 20, 2]


Visit the Wondrous
FOREST OF FUNCTION EXPRESSIONS