Exercise on Destructuring

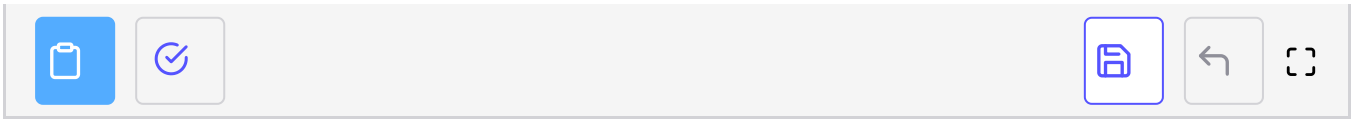
Let's try writing a few destructuring assignments and see how they simplify our code. Good luck!

Exercise 1:

Swap two variables using one destructuring assignment.

```
let text1 = 'swap';
let text2= 'me';

//Write Code here
```



Explanation

The `text1 = text2` and the `text2 = text1` assignments take place in parallel from the perspective of the whole expression. The expression on the right is evaluated, and becomes `[ 'me', 'swap' ]`. This evaluation happens before interpreting the expression on the left.

Exercise 2:

Complete the function below that calculates the nth fibonacci number in the sequence with one destructuring assignment! The definition of Fibonacci numbers is the following:

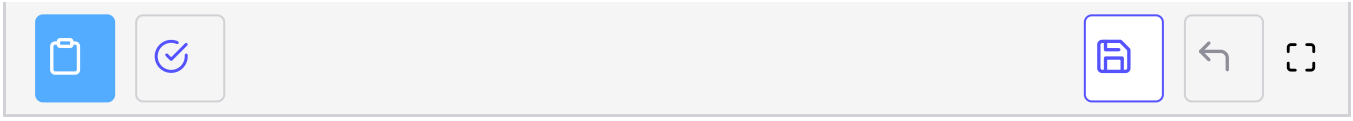
- `fib( 0 ) = 0`
- `fib( 1 ) = 1`
- `fib( n ) = fib( n-1 ) + fib( n-2 )`

```
function fib( n ) {
    let fibCurrent = 1;
    let fibLast = 0;
```

```
    if ( n < 0 ) return NaN;
    if ( n <= 1 ) return n;


    for ( let fibIndex = 1; fibIndex < n; ++fibIndex ) {
        // Insert one destructuring expression here
    }

    return fibCurrent;
}
```

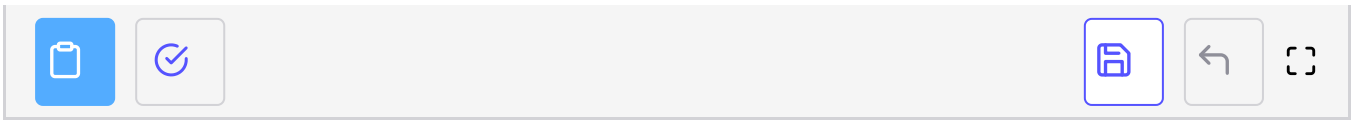


Exercise 3:

Create one destructuring expression that declares exactly one variable to retrieve `x.A[2]`. Return the value in a new variable called `A_2`.

```
let x = { A: [ 't', 'e', 's', 't' ] };

//Write your Code here
let A_2 = "";
```



Explanation

You don't have to provide variable names to match `A[0]`, `A[1]`, or `A[3]`. For `A[3]`, you don't even need to create a comma, symbolizing that `A[3]` exists. Similarly, adding two commas after `A_2` does not make a difference either, as in JavaScript, indexing outside the bounds of an array gives us undefined. Note that `A` was not created as a variable in the expression. You cannot assign the name of a variable and destructure its contents at the same time.

Exercise 4:

Suppose the following configuration object of a financial chart is given:

```
let config = {
    chartType : 0,
    bullColor : 'green',
    bearColor : 'red',
    days      : 30
};
```

Complete the function signature below such that the function may be called with any `config` objects (`null` and `undefined` are not allowed as inputs). If any of the four keys are missing, substitute their default values. The default values are the same as in the example configuration object.

```
function drawChart( data, /* Write your code here */ ) {
  // do not implement chart drawing functionality or anything
        // return {chartType, bullColor, bearColor, days};
};
```