**Swap tutorial**

**Design**

In the tutorial section, we developed a module that introduces the basic idea of swapping two elements in a programming context. As a newcomer of programming would have little knowledge about this operation swap, we designed this page to provide basic view of how swap operates. This page helps users prepare for understanding sorting algorithms which will use swap.

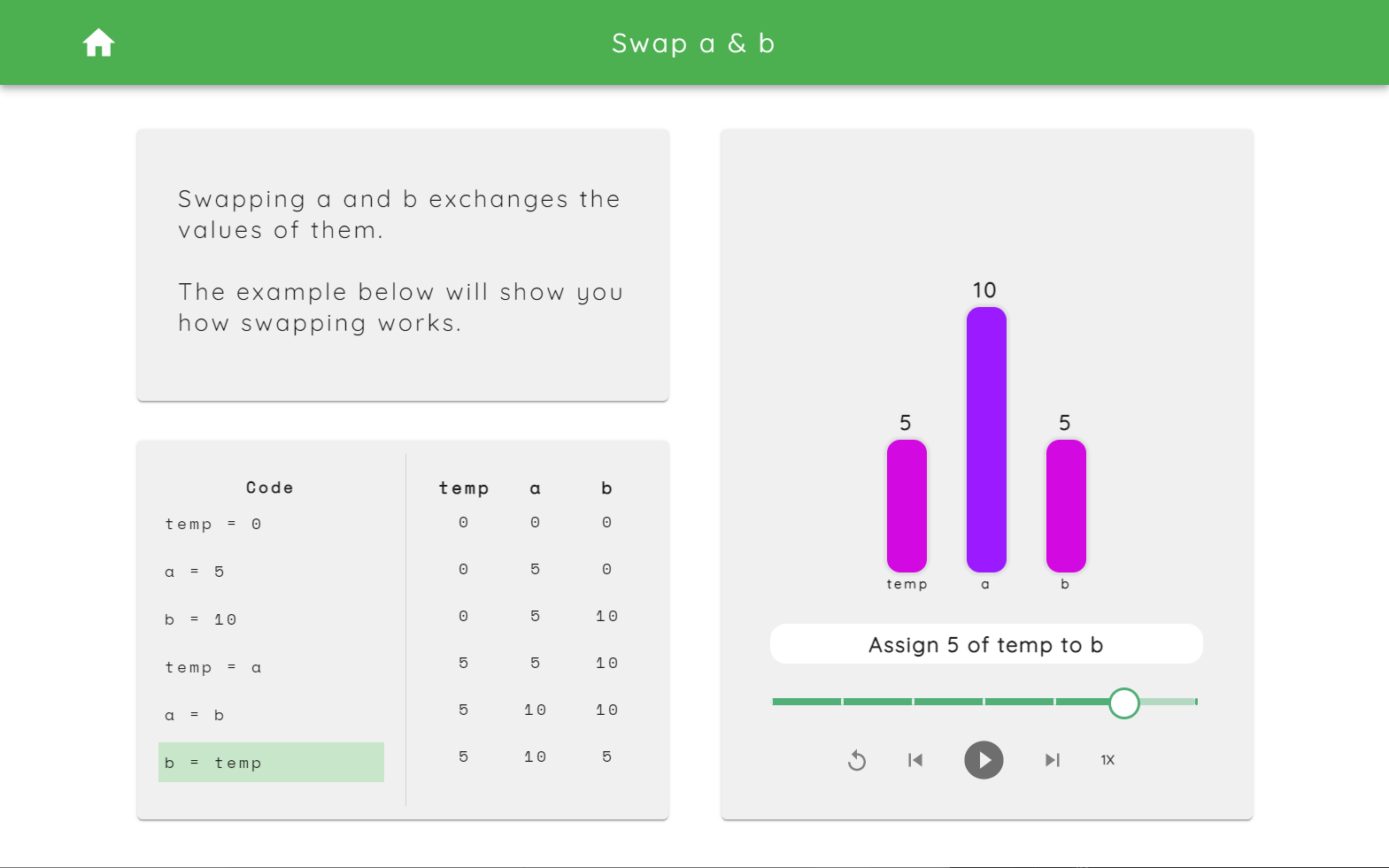


Figure 1. Swap a & b

There are three cards, an introduction, a code & variable table and a bar visualisation of the process in the page. By pressing the play button, it will demonstrate each step of swapping two variables automatically. The code part at the bottom-left corner highlights the current executing step with a background of green. The vertical displayed table will update the real-time variable value step by step.

**Implementation**

Automated animation is realised by the setTimeout function, which would execute particular instruction after specified periods.

// Set a timeout for each item in the trace

subTrace.forEach((item, i) => {

    let timeoutId = setTimeout(

        (item) => {

            // update the current step

            setCurrentStep((prevStep) =>

                i === trace.length - 1 ? prevStep : prevStep + 1

            );

            // update bars to be animated

            setBars(item);

            i === subTrace.length - 1

                ? setIsPlaying(false)

                : setIsPlaying(true);

        },

        i \* timer, //time interval

        item

    );

    timeoutIds.push(timeoutId);

});

Code 1

Code trace is realised by changing the class name of each line’s style.

<pre className={blockNum===1 ? classes.background : classes.noBackground}>

    {`temp = 0`}

</pre>

Code 2

The table is updated according to a current step variable which indicates the current progress. Function tables takes current step to generate a table with desired rows.

function tables() {

    let tables=[];

    let i = 1;

    for(var index=0; index<currentStep; index++){

        i = index + 1;

        tables.push(table[i]);

    }

    return tables;

}

Code 3