Calculator Application

Assigning Click Event Listeners on a Loop

Working with Math Operator Buttons

This Calculator application is a simplified version of a calculator.

Here's what the calculator can do:

- Add, subtract, multiply or divide 2 numbers

Here's what the calculator cannot do:

- Perform any other math operations
- Calculate with more than 2 numbers
- Store answers (all is reset when you click "clear")

The user interface consists of buttons that call functions. The functions are called in order, as follows

I. onNumberClick()

- The user clicks a digit (0-9) or the decimal point to call the **onNumberClick()** function, which:
  - concatenates the digit (or decimal point) as **numStr**
  - updates the display in the ouput box.

II. onOperatorClick()

- The user clicks an operator (+,-,*,/) to call the **onOperatorClick()** function, which:
  - converts **numStr** to **num**, an actual number
  - pushes **num** into the **nums** array
  - saves the operator to a variable, **oper**
  - updates the ouput box to include the operator

III. onNumberClick() again

- The user inputs the second number, which again calls **onNumberClick()**

IV. calculateAnswer()

- After entering the second number, the user clicks the equal sign (**=**) to call the **calculateAnswer()** function, which:
  - pushes the second number into the **nums** array

    - runs a series of if-statements, to determine what the operator is and to perform the correct calculation:
        - If operator is **'+'**, add the two numbers
        - If operator is **'-'**, subtract the second number from the first
        - If operator is **'*'**, multiply the two numbers
        - If operator is **'/'**, divide the first number by the second
    - updates the ouput box to include the answer

V. clearBox()

- Clicking **clr** (clear) calls **clearBox()** function, which:
    - empties the box
    - resets the variables

output box NOT directly editable
To reduce the amount of code required for this application, the output box has been intentionally *not* been made editable. Since the user cannot click inside the box and type, the only way to enter content into the box is by clicking the buttons.

numbers cannot be erased / deteted
Also to reduce the amount of code required for this application, there is no delete button to remove content or change the inputted numbers.

1. Open the file **05.07-Calculator.html**, and take it for a spin in the browser:

- enter a number (as many digits as you like)
- click one of the four operators: *+, -, , /
- enter another number
- click the **=** (equals) sign to get the answer
- click **clear** to reset

2. Have a look at the tags in the html file. Note that the "buttons" are actually divs:

```
    <section>
        <div id="0" class="num-btns">0</div>
        <div id="1" class="num-btns">1</div>
        <div id="2" class="num-btns">2</div>
        <div id="3" class="num-btns">3</div>
        <div id="4" class="num-btns">4</div>
        <div id="5" class="num-btns">5</div>
        <br>
    </section>

    <section>
        <div id="6" class="num-btns">6</div>
        <div id="7" class="num-btns">7</div>
        <div id="8" class="num-btns">8</div>
        <div id="9" class="num-btns">9</div>
        <div id="." class="num-btns">.</div>
        <div id="=" class="equals-btn">=</div>
    </section>
```

```
    <section>
        <div id="+" class="oper-btns">+</div>
        <div id="-" class="oper-btns">-</div>
        <div id="*" class="oper-btns">*</div>
        <div id="/" class="oper-btns">/</div>
        <div id="c" class="clear-btn">clear</div>
    </section>

    <div id="num-box">
        <!-- output appears here -->
    </div>
```

- The **num-btns** class occurs 10 times, once for each digit.
- The **oper-btns** class occurs 4 times, once for each operator.
- The **delete-btn**, **clear-btn** and **equals-btn** class occurs once each.

3. In the script tags, comment out the **FINAL.js** file and reactivate **START.js**.

4. Our task is to recreate **FINAL.js** from scratch, so open **05.07-Calculator-START.js**

We start by bringing in the "buttons". We will use **document.querySelectorAll()** to bring in multiple elements at once. These come in as an array of objects, known as a **Node List**.

5. Get the divs that have a class of **num-btns**. These are the digits from 0-9:

```
    const numBtnsArr = document.querySelectorAll('.num-btns');
```

6. Loop through the **numBtnsArr** array and assign event listeners to the objects:

```
    for(let i = 0; i < numBtnsArr.length; i++) {
        numBtnsArr[i].addEventListener('click', onNumberClick);
    }
```

7. Get the operator divs (+,-,*,/), which have a class of **oper-btns**:

```
    const operBtnsArr = document.querySelectorAll('.oper-btns');
```

8. Loop through the **operBtnsArr** arrays, adding listeners to each item:

```
    for(let i = 0; i < operBtnsArr.length; i++) {
        operBtnsArr[i].addEventListener('click', onOperatorClick);
    }
```

9. Get the other three "buttons" which call functions; these are the **=** (equals) and **clear** buttons:

```
const equalsBtn = document.querySelector('.equals-btn');
equalsBtn.addEventListener('click', calculateAnswer);

const deleteBtn = document.querySelector('.delete-btn');
deleteBtn.addEventListener('click', delete);

const clearBtn = document.querySelector('.clear-btn');
clearBtn.addEventListener('click', clearBox);
```

10. Get the **num-box**, which is the div for the output:

```
const numBox = document.getElementById('num-box');
numBox.textContent = "";
```

11. Next, declare several global variables for use by the functions:

- **numStr**: for storing the inputted digit(s) as a string
- **nums**: an array for storing the two **numStr** values
- **num**: the numeric version of **numStr**, for doing the math
- **oper**: a string for storing the operator symbol
- **answer**: for storing the calculated result

```
let numStr = "";
let num = 0;
let nums = [];
let oper = '';
let answer = 0;
```

12. Write the **onNumberClick()** function, which runs on click of any digit from 0-9 or the decimal point.

The **onNumberClick()** function concatenates the clicked digit onto **numStr**.

- The keyword **this** in a function always refers to the object which called the function, so **this.id** is the digit
- If the user clicks 1, 2, 3 in order, then **numStr** equals "123"
- The value of **numStr** is displayed in the box

```
function onNumberClick() {
    numStr += this.id; // concatenate the clicked digit
    numBox.textContent += this.id; // updates the output box
}
```

13. Write the **onOperatorClick()** function, which runs when an operator "button" is clicked:

- saves **this.id** (the clicked object id) to the **oper** variable
- pushes **numStr**, the second "number-like string", into the **nums** array
- resets **numStr** to make way for the second number
- outputs the operator, surrounded by spaces, to the box

```
function onOperatorClick() {
    oper = this.id;
    nums.push(numStr);
    numStr = '';
    numBox.textContent += ' ' + oper + ' ';
}
```

The user inputs the second of the two numbers. With each digit or decimal click, the **onNumberClick()** function is called, which concatenate the second "number-like string", saves it to **nums** and displays it in the box.

When the user is done inputting the second number, they click the equal sign, which calls the **calculateAnswer()** function.

14. Write the **calculateAnswer()** function, which:

- pushes the second "number-like string" into the **nums** array
- converts the two array items to actual numbers, using the **Number()** method and saves the results as variables, **num1** and **num2**
- runs a series of if-statements to identify the operator and does the correct mathematical calculation based on the operator
- runs an if-else-statement, which passes the **answer** to the **Number.isInteger()** method, which returns true if the answer is an integer. Else, the answer is a float, which gets passed to the **toFixed()** method, which rounds it to 7 decimal places
- The **answer** is outputted, preceded by **' = '**

```
function calculateAnswer() {
    nums.push(numStr);
    let num1 = Number(nums[0]);
    let num2 = Number(nums[1]);
    if(oper == '+') answer = num1 + num2;
    if(oper == '-') answer = num1 - num2;
    if(oper == '*') answer = num1 * num2;
    if(oper == '/') answer = num1 / num2;
    if(Number.isInteger(answer)) {
        numBox.textContent += ' = ' + answer;
    } else { // not an integer
        numBox.textContent += ' = ' + answer.toFixed(7);
    }
}
```

15. Write the **clearBox()** function, which empties the output box and resets the global variables. The calculator is ready for fresh input.

```
function clearBox() {
    numBox.textContent = '';
    nums = [];
    num = 0;
    oper = '';
    total = 0;
    numStr = '';
}
```

END: Lesson 05.07
NEXT: Lesson 05.08