3주차 과제2 (계산기) 기능 명세

기능 번호	세부 기능 설명	추가 기능 여부
FR1	화면에는 숫자 키패드(0-9)와 사칙연산 기호(+, -, *, /) 버튼이 있습니다.	N
FR2	입력된 숫자와 연산 기호를 화면의 상단에 표시하여 사용자가 현재 입력 상황을 확인할 수 있습니다.	N
FR3	'=' 버튼을 누르면, 현재 까지의 연산 결과가 화면에 표시됩니다.	N
FR4	'C' 또는 'CE' 버튼을 통해 입력된 내용을 초기화할 수 있습니다.	N
FR5	'⌫' 버튼을 통해 수식을 하나씩 삭제할 수 있습니다.	Y
FR6	'⏱' 버튼을 통해 연산 기록을 확인할 수 있습니다.	Y

구현

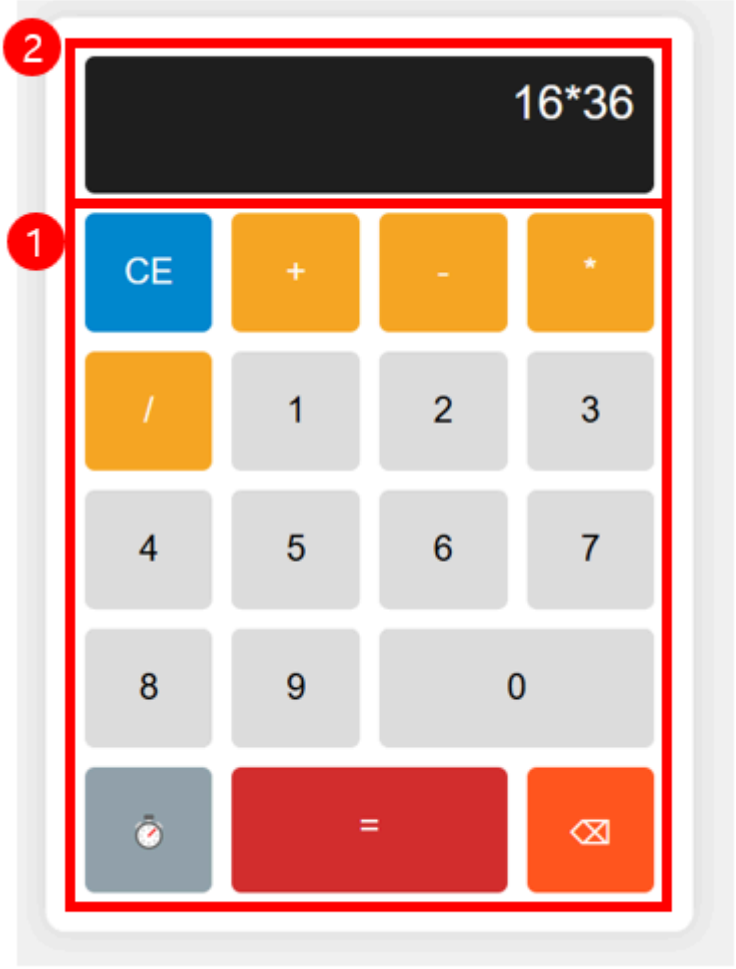
구현 사항

기능 번호	완료여부 (O,X)	세부사항
FR1	O	<진행사항> 숫자 키패드와 사칙연산 기호 버튼 표시
FR2	O	<진행사항> 각 버튼 클릭 시 상단에 있는 화면을 통해 현재 입력 상황 확인 가능

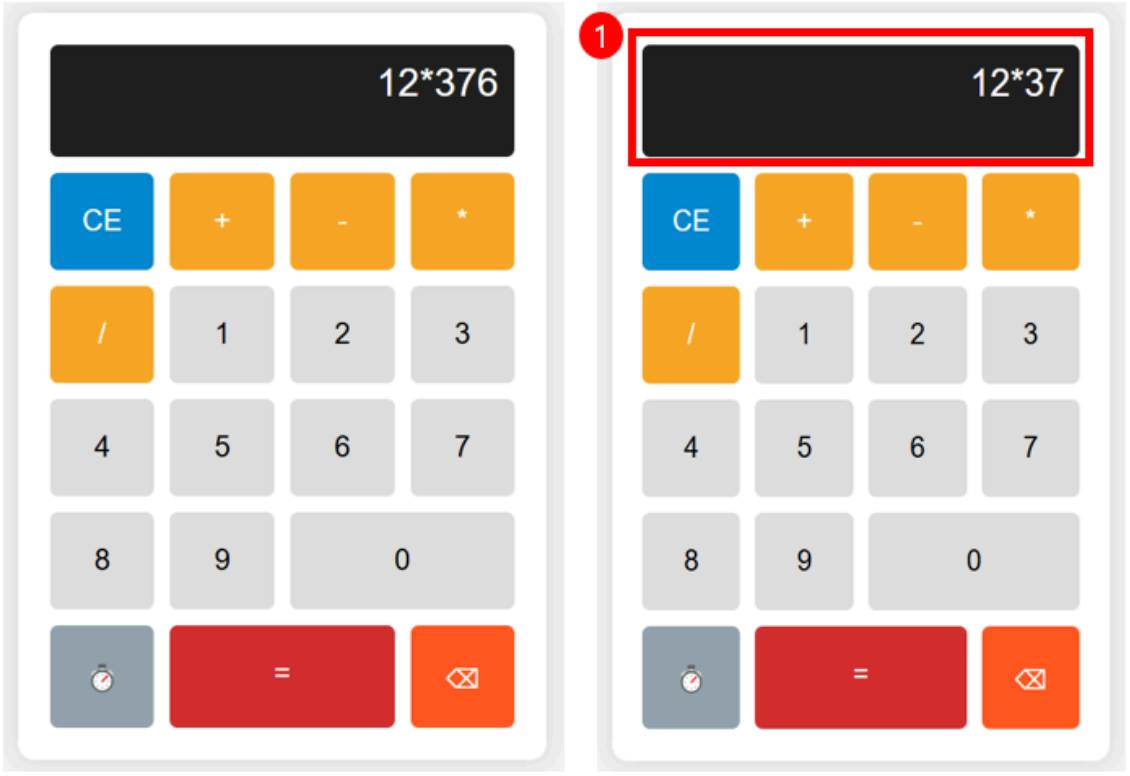
FR3	O	<진행사항> ‘=’버튼을 누르면 연산 결과가 화면에 표시 적절한 수식이 입력되지 않은 상태에서 ‘=’버튼을 클릭하면 에러를 표시
FR4	O	<진행사항> ‘CE’버튼을 통해 입력된 내용 초기화
FR5	O	<진행사항> ‘⌫’ 버튼을 클릭하면 입력된 내용을 하나씩 삭제
FR6	O	<진행사항> ‘ ⏱ ’ 버튼을 통해 연산 기록을 확인

웹 화면 구성

제목 : 계산기

화면	
	
설명	① 0-9의 숫자 및 연산자(+,-,*,/) 버튼이 표시되어 있습니다. 화면을 초기화하는 'CE' 버튼과 입력된 내용을 하나씩 삭제하는 '⌫'버튼, 연산기록을 확인하는 '⏱'버튼도 표시되어 있습니다. ② 입력된 결과 값이 다음과 같이 표시됩니다.

제목 : '⌫'버튼을 한 번 누른 상태의 계산기 화면

화면	
	
설명	① '⌫'버튼을 한 번 누르면 가장 뒤에 입력된 문자가 지워집니다.

제목 : '⏱'버튼을 눌렀을 때의 계산기 화면

화면	
	
설명	① 아무것도 누르지 않았을 때의 계산기 화면입니다. ② '⏱'버튼을 눌렀을 때 다음과 같은 화면이 표시됩니다.

구현 내용

HTML

파일명	구현 내용
index.html	계산기 화면을 구성하고 css와 js 파일을 적용

CSS

파일명	구현 내용
styles.css	화면에 적용할 스타일을 정의 - 기본 스타일 - 계산기 스타일 - 버튼 스타일 - 기록 스타일

JavaScript

파일 명 : script.js

함수명	구현 내용
appendNumber(number)	함수 인자 설명 number: 클릭된 숫자 버튼의 값 리턴값: 없음 동작 설명: currentInput에 전달된 숫자를 추가한다. 화면의 디스플레이에 현재 입력 값을 표시한다.
appendOperator(operator)	함수 인자 설명 operator: 클릭된 연산자 버튼의 값 리턴값: 없음 동작 설명: 현재 입력이 비어있거나 마지막 문자가 연산자인 경우 연산자를 추가하지 않는다. 연산자를 currentInput에 추가하고, 화면에 업데이트된 입력값을 표시한다.
clearDisplay()	함수 인자 설명: 없음 리턴값: 없음

	동작 설명: currentInput을 비우고, 화면의 디스플레이도 초기화한다.
deleteLast()	함수 인자 설명: 없음 리턴값: 없음 동작 설명: currentInput에서 마지막 문자를 제거한다. 화면의 디스플레이에 업데이트된 입력값을 표시한다.
infixToPostfix(infix)	함수 인자 설명 infix: 중위 표기법의 수식 리턴값: 후위 표기법의 수식 동작 설명: 중위 표기법을 후위 표기법으로 변환한다. - 수식을 후위 표기법으로 변환한다. 연산자의 우선순위에 따라 스택에서 연산자를 팝하여 결과를 출력 배열에 추가한다.
evaluatePostfix(postfix)	함수 인자 설명 postfix: 후위 표기법의 수식 리턴값: 계산 결과 동작 설명: 후위 표기법의 수식을 계산한다. - 후위 표기법 배열을 순회하며 숫자는 스택에 푸시하고, 연산자는 스택에서 두 개의 숫자를 팝하여 연산 후 결과를 다시 스택에 푸시한다..
calculate()	함수 인자 설명: 없음 리턴값: 없음 동작 설명: infixToPostfix를 호출하여 중위 표기법 수식을 후위 표기법으로 변환한다. evaluatePostfix를 호출하여 후위 표기법 수식을 계산한다. 결과가 유효한 경우 디스플레이에 결과를 표시하고 기록에 추가한다. 에러가 발생하면 'Error'를 표시한다.
updateHistory()	함수 인자 설명: 없음 리턴값: 없음 동작 설명: history 배열의 내용을 기반으로 기록 목록을 업데이트 한다. 기록 항목을 화면에 표시한다.
toggleHistory()	함수 인자 설명: 없음 리턴값: 없음

	동작 설명: historyElement의 display 속성을 토글하여 연산 기록을 보여주거나 숨긴다.

테스트 결과

번호	테스트 제목	테스트 절차	관련 FR	테스트 결과(O,X)
1	덧셈 연산 검증	1. 2를 입력한다. 2. ‘+’버튼을 클릭한다. 3. 8을 입력한다. 4. ‘=’버튼을 클릭한다.	FR 1 FR 2 FR 3	O
2	0으로 나누기 연산의 예외 처리	1. 7을 입력한다. 2. ‘/’버튼을 클릭한다. 3. 0을 입력한다 4. ‘=’버튼을 클릭한다.	FR 1 FR 2 FR 3	O
3	뺄셈 연산 검증	1. 9를 입력한다. 2. ‘-’버튼을 클릭한다. 3. 4를 입력한다. 4. ‘=’버튼을 클릭한다.	FR 1 FR 2 FR 3	O
4	곱셈 연산 검증	1. 6을 입력한다. 2. ‘*’버튼을 클릭한다. 3. 5를 입력한다. 4. ‘=’버튼을 클릭한다.	FR 1 FR 2 FR 3	O
5	‘CE’버튼 기능 검증	1. 3을 입력한다. 2. ‘+’버튼을 클릭한다. 3. ‘CE’버튼을 클릭한다.	FR 1 FR 2 FR 4	O
6	연산자 이후 ‘=’ 클릭	1. 6을 입력한다. 2. ‘+’버튼을 클릭한다. 3. ‘=’버튼을 클릭한다.	FR 1 FR 2 FR 3	O

7	‘⏱’버튼 기능 검증	1. 4를 입력한다. 2. ‘*’버튼을 클릭한다. 3. 3을 입력한다 4. ‘=’버튼을 클릭한다. 5. ‘⏱’버튼을 클릭한다.	FR 1 FR 2 FR 6	O
8	‘⌫’버튼 기능 검증	1. 3을 입력한다. 2. 7을 입력한다. 3. ‘*’버튼을 클릭한다. 4. ‘⌫’버튼을 클릭한다.	FR 1 FR 2 FR 5	O
9	연산자 연속 입력 방지	1. 2를 입력한다. 2. ‘*’버튼을 클릭한다. 3. ‘/’버튼을 클릭한다.	FR 1 FR 2	O

집중 피드백 받고 싶은 사항

번호	내용
1	eval 함수를 사용하지 않고서는 후위표기법을 이용한 방법밖에 없는 지, 괄호나 더 복잡한 식이 등장하게 되면 어떻게 계산해야 할 지 궁금합니다.
2	
3	