# 기능 명세

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 세부 기능 설명 | 추가기능여부 |
| FR1 | 화면에는 숫자 키패드(0-9)와 사칙연산 기호(+,-,\*,/) 버튼이 있습니다. | Y |
| FR2 | 입력된 숫자와 연산 기호를 화면의 상단에 표시하여 사용자가 현재 입력 상황을 확인 가능합니다. | Y |
| FR3 | ‘=’ 버튼을 누르면 현재까지의 연산 결과가 화면에 표시됩니다. | Y |
| FR4 | ‘C’ 또는 ‘CE’ 버튼을 통해 입력된 내용을 초기화 할 수 있습니다. | Y |
| FR5 | **본인만의 기능을 1개 추가**합니다. (계산기 창 닫기 기능) | Y |

# 구현

## 구현 사항

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 완료여부  (O,X) | 세부사항 |
| FR1 | O | <진행사항>  화면에는 숫자 키패드(0-9)와 사칙연산 기호(+,-,\*,/) 버튼이 있습니다.  <안 된 부분>  <상세 내용> |
| FR2 | O | <진행사항>  입력된 숫자와 연산 기호를 화면의 상단에 표시하여 사용자가 현재 입력 상황을 확인 가능합니다.  <안 된 부분>  <상세 내용> |
| FR3 | O | <진행사항>  ‘=’ 버튼을 누르면 현재까지의 연산 결과가 화면에 표시됩니다.  <안 된 부분>  <상세 내용> |
| FR4 | O | <진행사항>  ‘C’ 또는 ‘CE’ 버튼을 통해 입력된 내용을 초기화 할 수 있습니다.  <안 된 부분>  <상세 내용> |
| FR5 | O | <진행사항>  **본인만의 기능을 1개 추가**합니다. (계산기 창 닫기 기능)  <안 된 부분>  <상세 내용> |

## 웹 화면 구성

제목 : 계산기 메인

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | ① 전체적으로 계산기 모양과 유사하게 디자인하였습니다.  ② 결과 화면과 키패드 부분을 직관적으로 표현하였습니다. |

## 구현 내용

### HTML

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| index.html | calculate-body라는 가장 상위 div 태그 안에는 계산 결과를 표시하는 요소가 있으며, 기본값은 0입니다. 또한 키패드 섹션이 있어, 각 div 태그를 이용해 숫자와 연산자 버튼들이 나열되어 있습니다. |

### CSS

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| style.css | NeoDunggeunmo라는 웹 폰트를 페이지 전체에 적용하고, Flexbox를 사용하여 요소들을 정렬합니다. ‘.calculate-body, .result-view, .row, .key’ 클래스에 display: flex와 정렬 속성을 적용해 중앙 정렬을 구현했습니다. ‘.key’ 클래스는 버튼의 기본 스타일을 설정하고, ‘operator, power, none’ 클래스를 통해 각 버튼의 배경색을 다르게 하였습니다. hover 시 확대되는 효과를 주어 사용자가 어떤 키를 선택하고 있는지 쉽게 인식할 수 있도록 하였습니다. |

### JavaScript

파일 명 : main.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| appendNumber(num) | 함수 인자 : num (추가할 숫자)  리턴값 : X  동작 설명 : currentInput에 숫자를 추가하고 updateResultView를 호출하여 결과를 업데이트합니다. |
| appendOperator(operator) | 함수 인자 : operator (추가할 연산자)  리턴값 : X  동작 설명 : currentInput 에 연산자를 추가하고 updateResultView 를 호출하여 결과를 업데이트합니다. |
| clearResult() | 함수 인자 : X  리턴값 : X  동작 설명 : currentInput 을 초기화하고 updateResultView 를 호출하여 결과를 0 으로 설정합니다. |
| calculateResult() | 함수 인자 : X  리턴값 : X  동작 설명 : currentInput 이 비어있지 않은 경우 checkAgain 을 호출하여 계산을 시작하고, 비어 있으면 경고 메시지를 표시합니다. |
| updateResultView() | 함수 인자 : X  리턴값 : X  동작 설명 : currentInput의 값을 결과 표시 요소에 업데이트합니다. currentInput 이 비어있으면 0으로 표시합니다. |
| exit() | 함수 인자 : X  리턴값 : X  동작 설명 : 사용자가 계산기 종료를 확인하는 대화상자를 표시하고, 확인 시 현재 창을 닫습니다. |
| checkAgain(inputText) | 함수 인자 : inputText(계산할 수식)  리턴값 : X  동작 설명 : getArr 함수를 호출하여 입력된 수식을 배열로 변환합니다. |
| getArr(string) | 함수 인자 : string(수식 문자열)  리턴값 : X  동작 설명 : 수식 문자열을 특정 기호로 나누고 필터링하여 배열로 변환한 후 startOperation() 함수를 호출합니다. |
| startOperation0(equation) | 함수 인자 : equation(수식 배열)  리턴값 : X  동작 설명 : 곱셈이나 나눗셈이 포함되어 있으면 operation1 을 호출하고, 덧셈이나 뺄셈만 포함되어 있으면 operation2 를 호출합니다. 그렇지 않으면 errorProcess를 호출합니다. |
| getResult(v, num1, num2) | 함수 인자 : v(연산자), num1(첫 번째 숫자), num2(두 번째 숫자)  리턴값 : X  동작 설명 : 주어진 연산자에 따라 두 숫자를 계산하고 결과를 반환합니다. |
| operation1(equation) | 함수 인자 : equation(수식 배열)  리턴값 : X  동작 설명 : 곱셈과 나눗셈을 우선적으로 처리하여 결과를 계산하고, 그 후 operation2 를 호출합니다. |
| operation2(equation) | 함수 인자 : equation(수식 배열)  리턴값 : X  동작 설명 : 덧셈과 뺄셈을 처리하여 결과를 계산하고, 최종적으로 errorProcess를 호출합니다. |
| errorProcess(v) | 함수 인자 : v(최종 계산 결과 또는 오류)  리턴값 : X  동작 설명 : 수식이 유효하지 않으면 오류 메시지를 표시하고, 유효하다면 showResult 함수를 호출하여 결과를 표시합니다. |
| showResult(value) | 함수 인자 : value(결과값)  리턴값 : X  동작 설명 : 결과 표시 요소에 최종 결과를 업데이트합니다. |

# 테스트 결과

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 테스트 제목 | 테스트 절차 | 관련 FR | 테스트 결과(O,X) |
| 1 | 키패드 입력 | 키패드를 누르면 입력된 내용이 결과 화면에 표시되는지 확인한다. | FR 2 | O |
| 2 | 연산 결과 표시 | ‘=’ 버튼을 누르면 현재까지의 연산 결과가 화면에 표시되는지 확인한다. | FR 3 | O |
| 3 | ‘CE’ 버튼을 통해 입력된 내용 초기화 | ‘CE’ 버튼을 누르면 입력된 연산 내용이 초기화되는지 확인한다. | FR 4 | O |
| 4 | 계산기 창 닫기 | ‘👋’ 버튼을 누르면 계산기 창이 닫히는지 확인한다. | FR 5 | O |

# 집중 피드백 받고 싶은 사항

|  |  |
| --- | --- |
| 번호 | 내용 |
| 1 | 계산기 스크립트 파일은 제가 1년 전에 작성한 내용이라 변수 명 짓기나 메서드 사용에 미흡한 부분이 있습니다. 양해 부탁드립니다. |