

휴먼 컴퓨터 인터페이스


## 과제 #2. WebUI 계산기 구현

이 강 훈

광운대학교 소프트웨어학부

# 요구조건: 0단계

- WebUI 위젯 기능 구현
  - 템플릿 프로젝트에 기초
    - 코드(\): web\_ui\_v00.html, web\_ui\_v0.js
    - 리소스(\resources): HTML5.png, CSS3.png, JS.png
  - 위젯 관련 코드 입력
    - 슬라이드에서 제시한 대로 채워넣기
      - Widget, Text, Image, PushButton, TextField
    - 슬라이드에서 설명한 대로 구현하기
      - Switch
  - 로그인 페이지 구성
    - initWidgets()
    - layoutWidgets()



Introduction to HCI

HTML CSS JS

ID

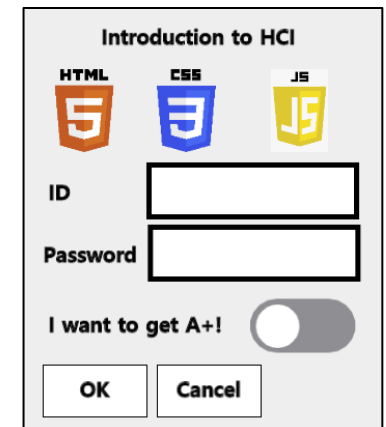
Password

I want to get A+! ☐

OK Cancel

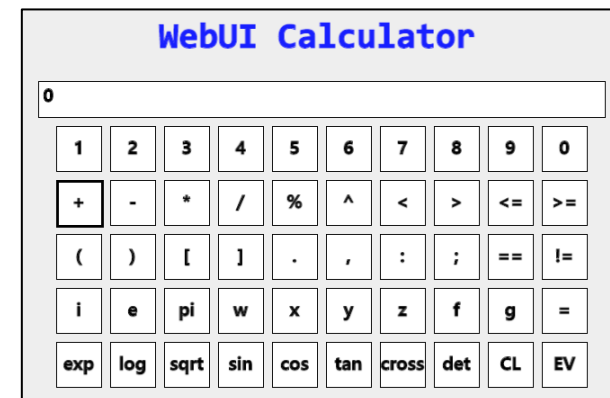
# 요구조건: 1단계

- WebUI 레이아웃 기능 구현
  - 템플릿 프로젝트에 기초
    - 코드(\): web\_ui\_v1.html, web\_ui\_v1.js
    - 리소스(\resources): HTML5.png, CSS3.png, JS.png
  - 위젯 관련 코드 복사 (0단계와 동일한 부분 빠짐)
    - Text, Image, PushButton, TextField, Switch
  - 레이아웃 관련 코드 입력 (남은 빠진 부분 채워 넣기)
    - WidgetTypes, Alignment, Widget, maxSize, minSize, ...
  - 로그인 페이지 재구성
    - initWidgets()
    - Container, Row, Column 이용
    - ❖ layoutWidgets() 함수는 불필요함



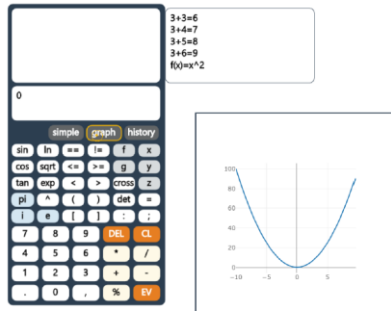
# 요구조건: 2단계

- WebUI를 이용하여 기본 계산기 구현
  - 1단계 결과를 새로운 이름으로 복사해서 작업
    - `web_ui_v2.html`, `web_ui_v2.js`
  - 수식 계산을 위한 객체 정의
    - `WebUI.parser = math.parser();`
  - `CalcButton` 클래스 정의
    - `PushButton`으로부터 상속
    - `handleButtonPushed` 멤버 함수 추가 (수식, CL, EV 입력 처리)
    - 생성자에 다음 코드 추가
      - `this.onPushed = WebUI.CalcButton.handleButtonPushed;`
  - `initWidgets()` 함수 재정의
    - 제목: `Container > Text`
    - 화면: `Container > Text`
    - 버튼: `Row > Column > CalcButton`



# 요구조건: 3단계

- WebUI를 이용하여 확장 계산기 구현
  - 2단계 결과를 새로운 이름으로 복사해서 작업
    - web\_ui\_v3.html, web\_ui\_v3.js
  - 보다 편리한 기능/인터페이스 제공
    - 다음 중 하나를 포함한 새로운 위젯 2개 이상 정의 및 사용
      - GraphPlot: 사용자 정의 함수에 대한 그래프 시각화
      - GridView: 2차원 격자 형태의 레이아웃 관리
- ❖ 참고: <https://flutter.dev/docs/development/ui/widgets>

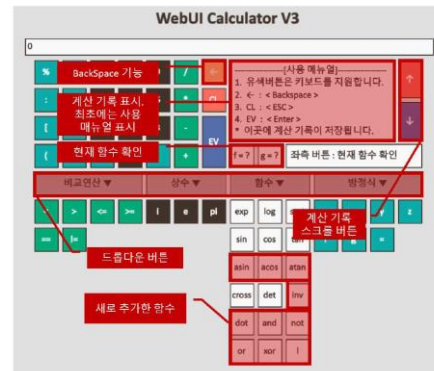


아래와 같은 입력의 결과이다.

$f(x)=x^2 \rightarrow EV$ 클릭  $\rightarrow$  "function"출력됨(정의완료)  $\rightarrow f(x) \rightarrow Graph$ 클릭  $\rightarrow$  그래프 출력

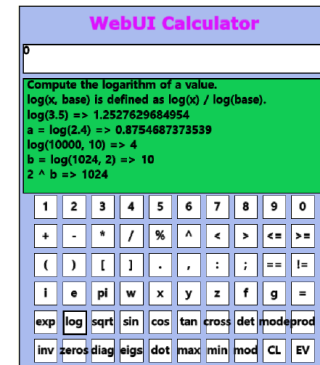
그래프 창 내에 마우스를 올리면 plotly.js에서 제공하는 각종 버튼들이 보인다. 이를 클릭하여 그래프를 확대, 축소, 이미지로 저장할 수 있으며 그래프 영역 이외의 곳을 클릭하면 그래프는 사라진다.

그래프 창은 div의 border속성을 이용해서 테두리를 표현해주었다.



추가된 새로운 기능에 대한 개요는 위와 같습니다. 앞서 설명한 BackSpace를 통해 한 문자만 지우는 것이 가능하고, 계산 기록에 대한 모든 사항이 저장됩니다. 저장된 계산 기록은 한 번에 6개만 표시되며, 스크롤 버튼이나 키보드 위아래 방향키로 이전 기록을 보거나, 양의 기록을 조회할 수 있습니다.  $f = ?$ ,  $g = ?$  버튼으로는 현재 함수가 어떤 형태인지 확인할 수 있습니다.

선승연산을 제외한 나머지 연산은 드롭다운 버튼을 통해 분류하였습니다. 드롭다운 버튼 클릭을 통해 하위 버튼들을 접었다 펼 수 있습니다. 또한, 함수에  $\sin$ ,  $\cos$ ,  $\tan$ 과 역행렬 함수( $\sin^{-1}$ ),  $\cos^{-1}$ ,  $\tan^{-1}$ )를 추가하였습니다.



마우스 커서를 특정 버튼 위에 놓기만 해도 관련 도움말이 도움말 창에 뜨는 것을 확인 할 수 있으며, 추가된 계산기 함수가 정상적으로 작동함을 확인하였다. 계산기 사용 중 오류가 발생한 경우에는 결과창에 오류 메시지가 출력하는 것을 알 수 있다. 또한 모든 과정에서 정상적으로 버튼클릭음, 초기화음, 에러음이 출력되었다.

# 제약조건 완화

- 클라이언트 측 스크립트만 사용
  - 서버 측 스크립트 X, 데이터베이스 X
- 오픈소스 라이브러리 허용
  - jQuery, fabric.js, math.js 외에도 자유롭게 사용
  - 단, 추가적으로 사용된 라이브러리 내역은 반드시 명시
- 추가 리소스 사용 권장
  - resources/ 폴더에 자유롭게 다양한 리소스 추가
- 구글 크롬 웹 브라우저 호환 필수
  - 타 웹 브라우저 호환 권장하나 필수 X

# 보고서

- 표지
  - 과목명, 과제명, 제출일, 소속, 학번, 이름 기재
- 본문
  - 요구조건과 제약조건 만족도 요약 (표)
    - 단계별 만족도 (0~3단계)
    - 추가적인 오픈소스 라이브러리 사용할 경우 명시
  - 0단계: 위젯 구현
    - 위젯 기능 구현을 위한 코드 제시 및 의미 설명
    - 로그인 페이지 구성을 위한 코드 제시 및 결과 분석
  - 1단계: 레이아웃 구현
    - 레이아웃 기능 구현을 위한 코드 제시 및 의미 설명
    - 로그인 페이지 재구성을 위한 코드 제시 및 결과 분석
  - 2단계: 기본 계산기
    - 구체적인 구현 방법 (CalcButton 클래스, initWidgets() 함수 등)
    - 실행 과정 제시 및 결과 분석
  - 3단계: 확장 계산기
    - 새로운 인터페이스/기능 개요
    - 새롭게 추가된 위젯들의 구현 방법
    - 실행 과정 제시 및 결과 분석 (기본 계산기로부터 향상된 측면 강조)
- 논의
  - 구현 측면에서 성공적인 부분과 실패한 부분
  - 전반적인 자체 평가 및 향후 개선점

# 제출

- “보고서+소스코드+리소스”를 \*.zip 파일로 압축해서 제출
  - 압축 파일의 이름은 학번 (예. 1234567890.zip)
  - 파일 용량 30MB 초과시 e-mail로 제출 (kang@kw.ac.kr)
  - \*.zip 파일 내부 폴더 구조
    - | - report.pdf
    - | - web\_ui0.html, web\_ui1.html, web\_ui2.html, web\_ui3.html
    - | - web\_ui0.js, web\_ui1.js, web\_ui2.js, web\_ui3.js
    - | - fabric.min.js, math.js
    - | + resources/
      - | - HTML5.png, CSS3.png, JS.png
      - | - 추가로 사용된 각종 리소스
      - | - 추가로 사용된 오픈소스 라이브러리
- 마감
  - 5월 8일 (토) 오후 11:59