

Web Page 에디터




Web Page 에디터 프로그램

담당 Skill & Stack	C/C++ HTML5 MFC Windows Form
요약	MFC와 HTML를 사용하여 HTML에 대해 모르더라도 간단한 웹페이지를 만들 수 있는 프로그램을 기획 및 개발
진행 기간	@2021년 9월 6일 → 2021년 9월 10일
팀 구성	개인


Source

GitHub - yeonseoksong/WebPage_Editor

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

 https://github.com/yeonseoksong/WebPage_Editor.git

yeonseoksong/
WebPage_Editor



1 Contributor 0 Issues 0 Stars 0 Forks

프로그램 버전 관리

버전	태그	프로젝트 진행 기간	비고
Ver 1.0	C++ HTML5 MFC	@2021년 9월 6일 → 2021년 9월 10일	초기 기획 및 개발
Ver 1.1	C++ HTML5 MFC	@2021년 10월 24일 → 2021년 10월 25일	노트북 고장으로 인하여 다시 프로그램 만듦

Aa 버전	☰ 태그	📅 프로젝트 진행 기간	☰ 비고
Ver 1.2	C++ HTML5 MFC	@2021년 12월 12일	클린 코드를 위해 기능 별로 함수 구현하려고 노력함
Ver 2.0	C++ CSS HTML5 Windows Form	@2021년 12월 12일	- MFC 라이브러리는 지원이 끊겼다는 점과 C++로 .NET Windows Form를 개발할 수 있다는 소식을 알게 되어 ver 2.0부터 Windows Form 라이브러리를 사용하기로 함 - modern C++, CSS 언어 기능 도입, wysiwyg 구현 등 다양한 기능을 도입할 예정 - 기획 및 개발 예상 기간 : 2주 - 자세한 계획은 페이지로 구성하여 업로드할 예정

개발 배경

- HTML 언어를 알지 못한다면 웹 페이지를 만들 수 없다는 점을 알게 되었다.
- 파워포인트 프로그램과 같이 HTML 언어 없이도 간단하게 웹 페이지를 만들 수 있는 프로그램을 만들었으면 좋겠다는 생각을 하게 되었다.

본인의 역할

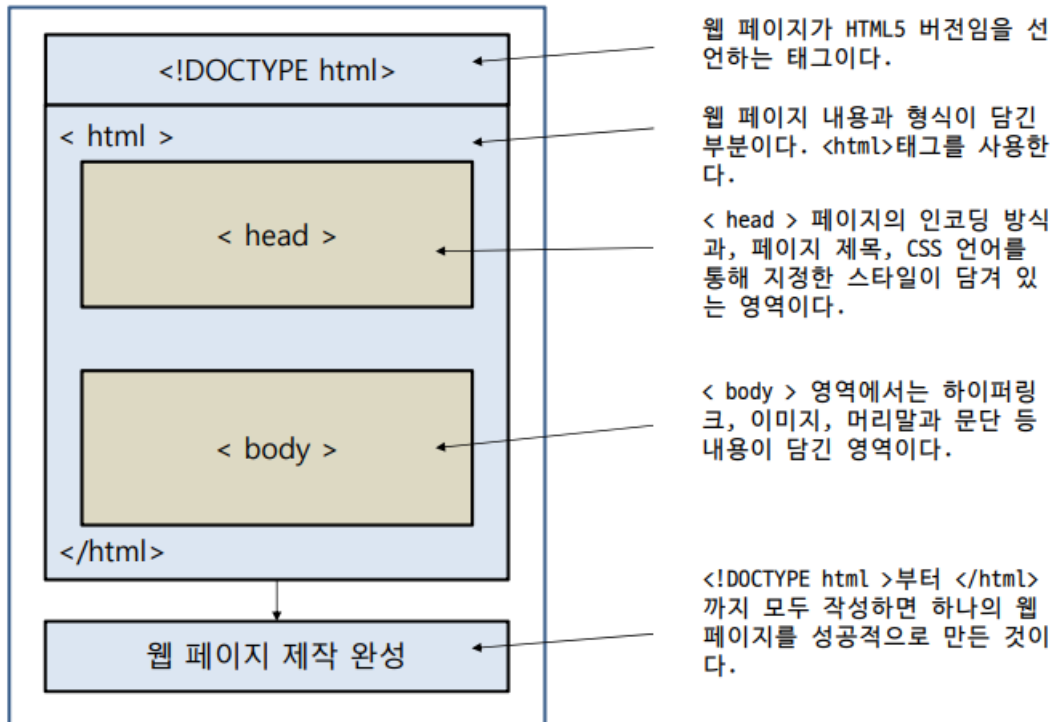
- 프로젝트 기획
- 프로젝트 기획에 따른 서비스 구현
- 프로젝트 기획에 따른 서비스 검토

기획 일정

- 서비스 기획 : 2021년 9월 6일
- 서비스 구현 : 2021년 9월 7일 ~ 2021년 9월 9일
- 서비스 검토 : 2021년 9월 10일

상세 내용

🌐 웹 페이지의 구성요소



메뉴 구성

- File : Open / Save / Exit
- Edit : 머리말 / 내용 / 하이퍼링크 / 이미지
- Information : Web Maker 정보 / 도움말

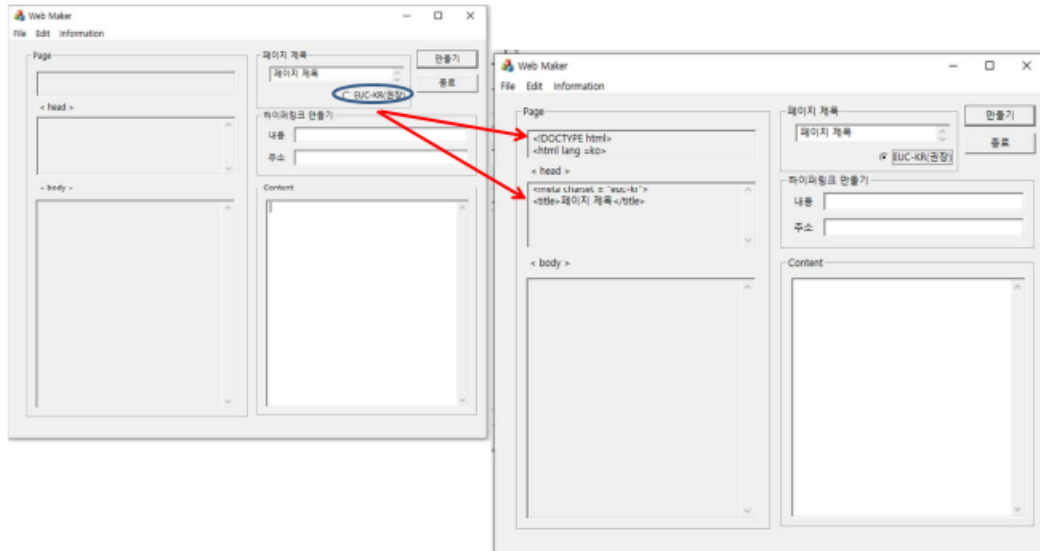


• 기능 요약

- Page 구역에 있는 3개의 Edit Dialog을 합친 후, [만들기] 버튼을 클릭하여 HTML 파일을 저장한다.
- 페이지 제목을 입력한 후, EUC-KR 버튼을 클릭하면 PRimary Edit Dialog에 <!DOCTYPE html>, <html lang = "ko">가 입력된다.
- 그리고 2번 Edit Dialog에 인코딩 정보(EUC-KR)와 페이지 제목이 입력된다.
- Content Edit Dialog에 값을 입력한 후, Edit 메뉴에 있는 머리말, 내용, 이미지 기능을 사용해 <body> Edit Dialog에 값을 입력할 수 있다.
- 하이퍼링크 내용과 주소를 각각 입력하여 Edit 메뉴에 있는 하이퍼링크 기능을 이용해 <body> Edit Dialog에 값을 입력할 수 있다.

번호	설명
1	<!DOCTYPE html>과 <html>이 담긴 Primary Edit Dialog. 수정이 불가능하다.
2	웹 페이지의 인코딩 방식과 페이지 제목을 입력 받는 Edit Dialog. 수정이 불가능하다.
3	웹 페이지의 내용을 입력 받는 Edit Dialog. 수정이 불가능하다.
4	웹 페이지의 내용을 입력할 수 있는 Edit Dialog
5	하이퍼링크의 내용과 주소를 입력 받는 Edit Dialog
6	페이지 제목을 입력 받는 Edit Dialog
7	페이지 제목과 EUC-KR 인코딩 방식을 '2'번 Edit Dialog에 입력하는 Radio Button
8	1과 2, 3의 Edit Dialog을 합쳐서 html 문서로 저장하는 버튼
9	프로그램의 종료 기능이 담긴 버튼

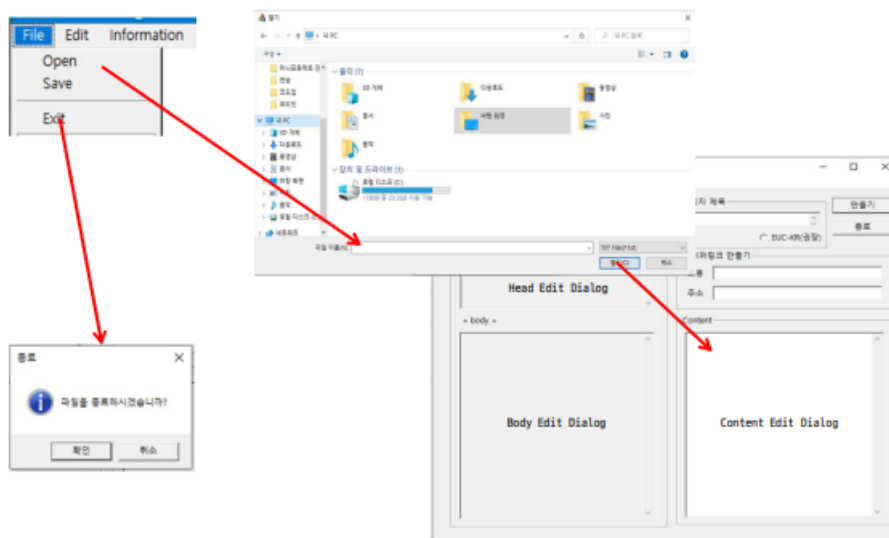
페이지 제목, 인코딩 유형 입력



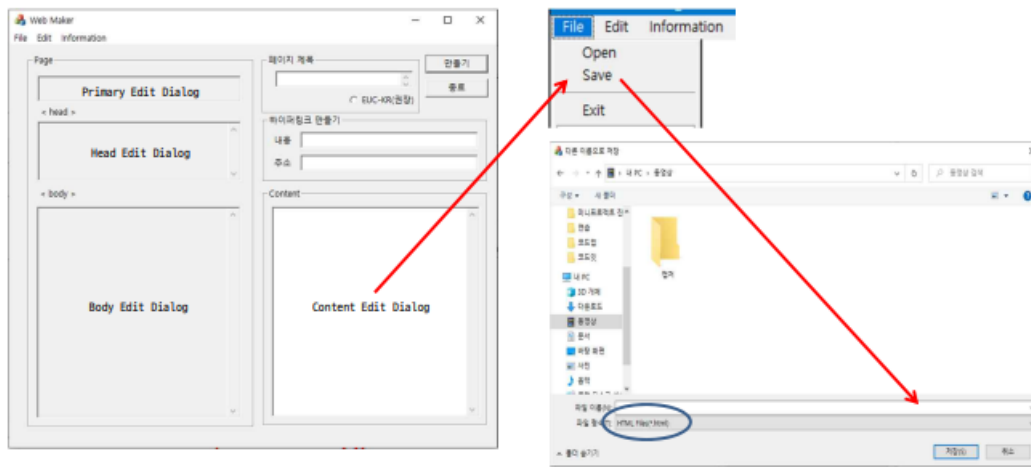
Main Menu - File

- Open : txt File open 및 Migration
- Save : Content Edit Dialog의 내용을 HTML 문서로 저장
- Exit : 프로그램 종료

Menu – File/New process



Menu – File/New process

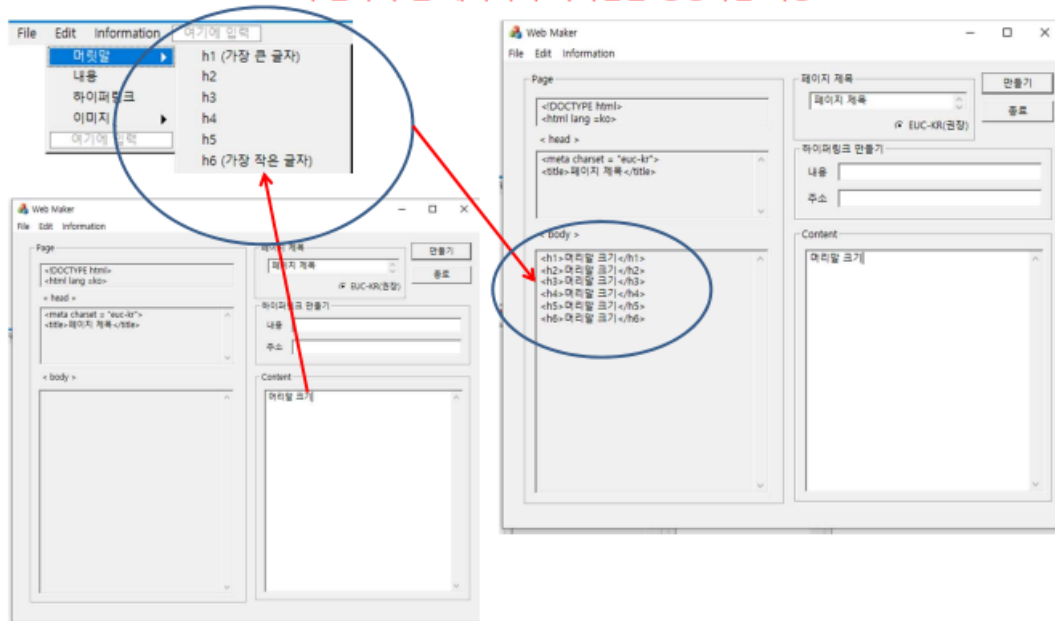


Main Menu - Edit

- 머리말 : h1(가장 큰 글씨) ~ h6(가장 작은 글씨)
 - <h1> ~ <h6> 태그
- 내용 : 내용 입력 기능 (<p> 태그)
- 하이퍼링크 : 하이퍼링크 생성 및 입력 기능
 - <a href "주소">내용
- 이미지 : 웹 이미지 첨부하기 / 이미지 (비활성화)
 -

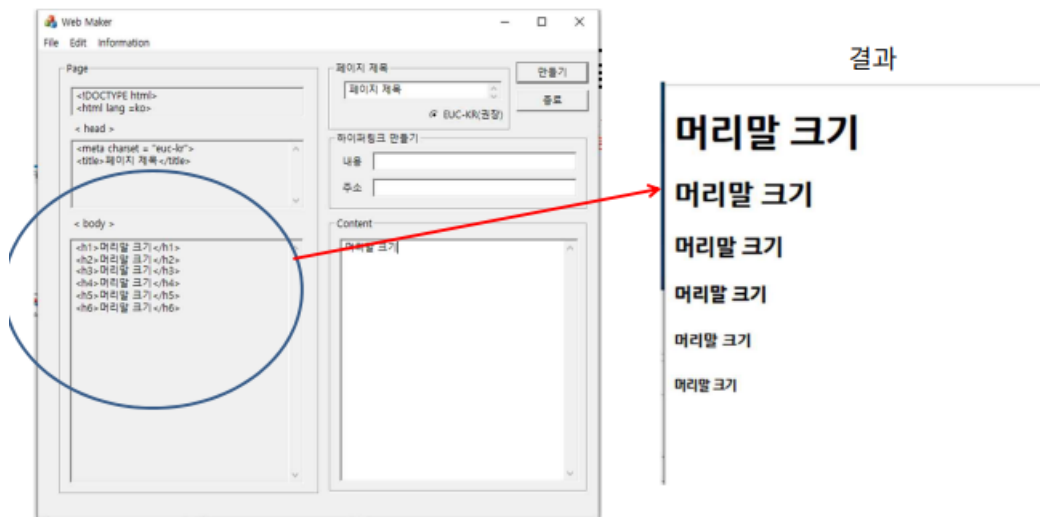
Menu – Edit / 머리말

- HTML의 머리말에 해당하는 <h1 ~ h6> 태그를 구현하여 웹 페이지의 머리말을 생성하는 기능

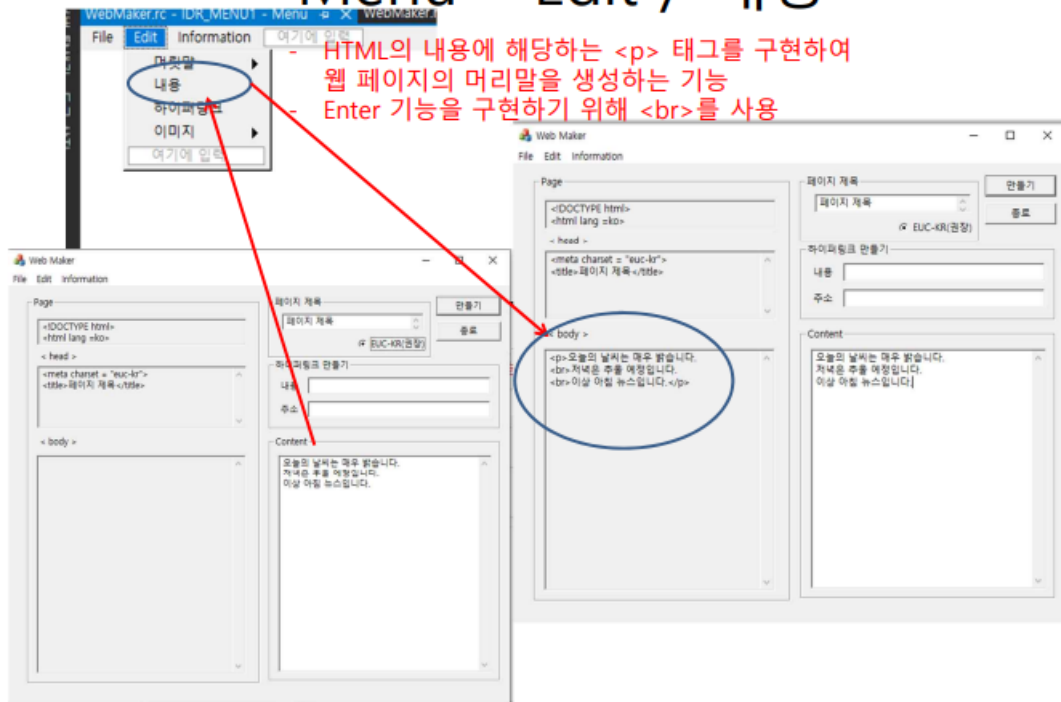


Menu – Edit / 머리말

- HTML의 머리말에 해당하는 <h1 ~ h6> 태그를 구현하여 웹 페이지의 머리말을 생성하는 기능

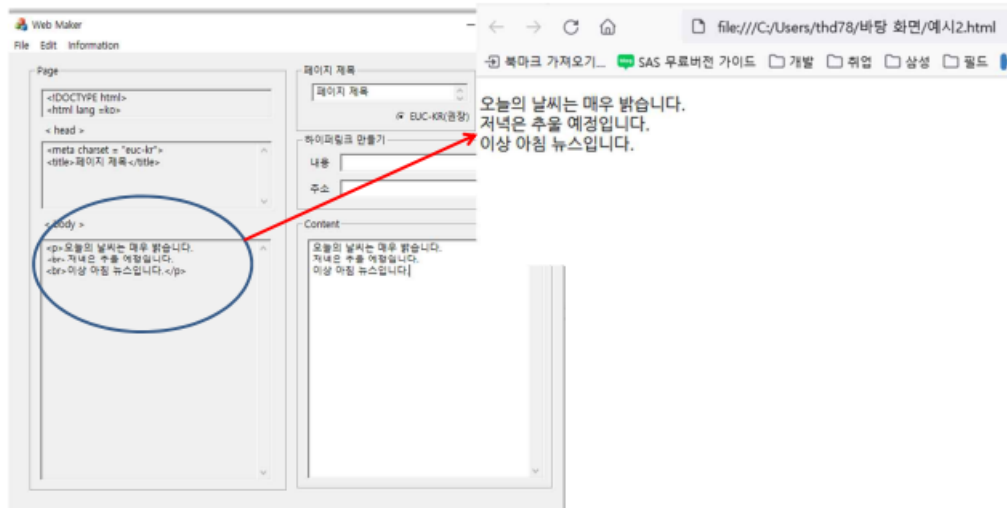


Menu – Edit / 내용

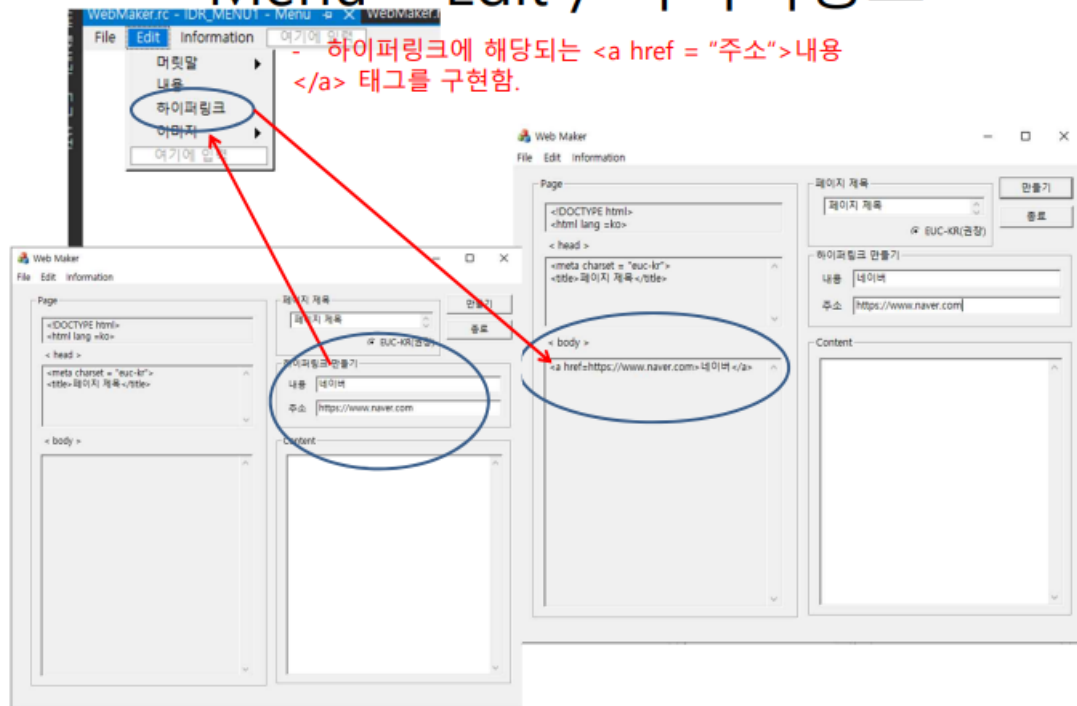


Menu – Edit / 내용

- HTML의 내용에 해당하는 <p> 태그를 구현하여 웹 페이지의 머리말을 생성하는 기능
- Enter 기능을 구현하기 위해
태그를 사용

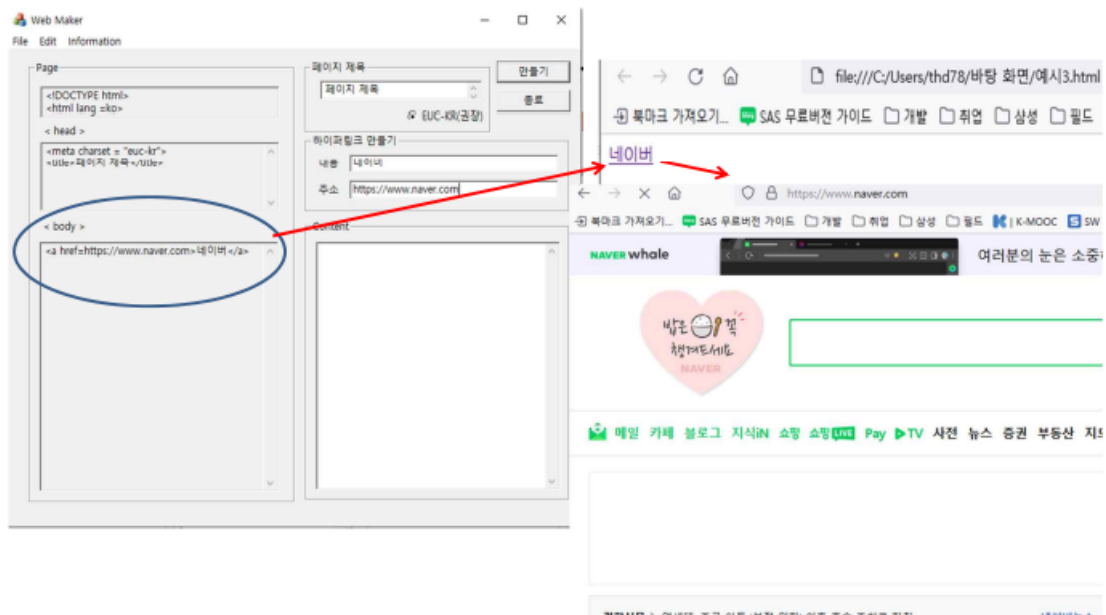


Menu – Edit / 하이퍼링크



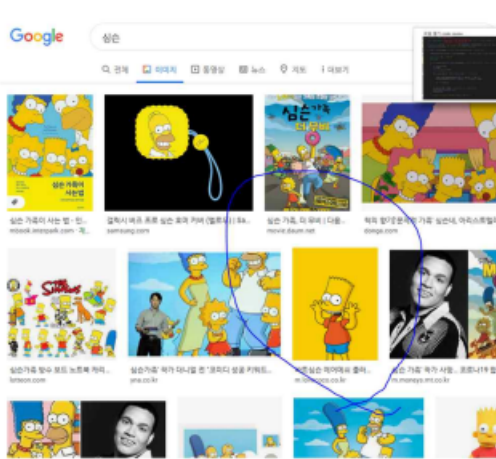
Menu – Edit / 하이퍼링크

- 하이퍼링크에 해당되는 `내용` 태그를 구현함.

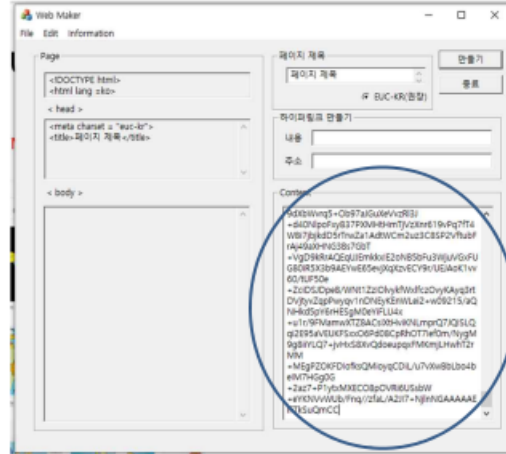


Menu – Edit / 이미지 – 웹 이미지 첨부하기

- HTML의 이미지 첨부 태그인 를 구현



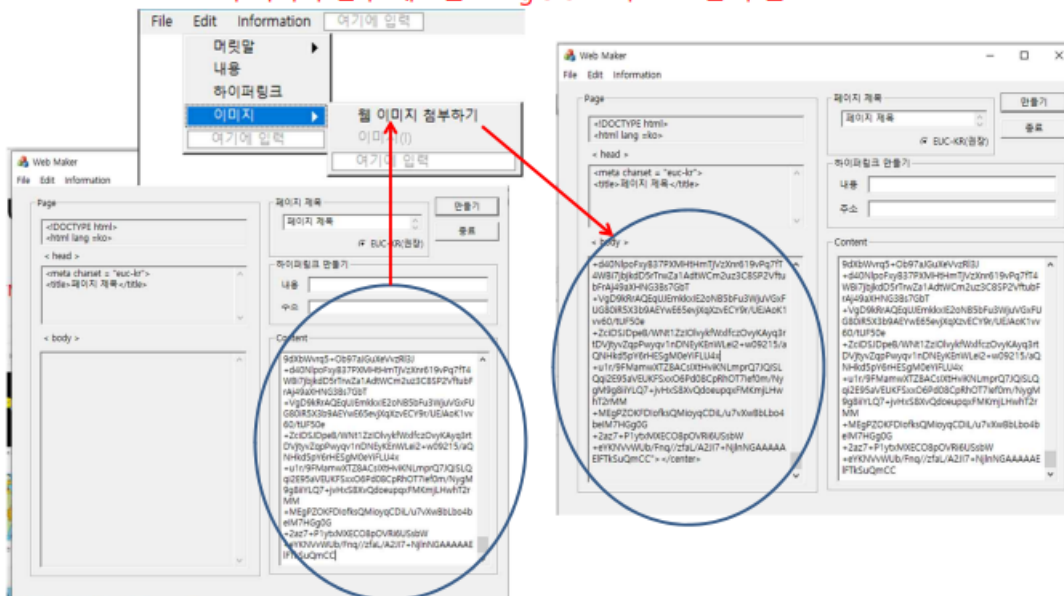
- 인터넷에서 원하는 그림 링크 주소를 가져옴



- Content Edit Control에 주소를 복사함

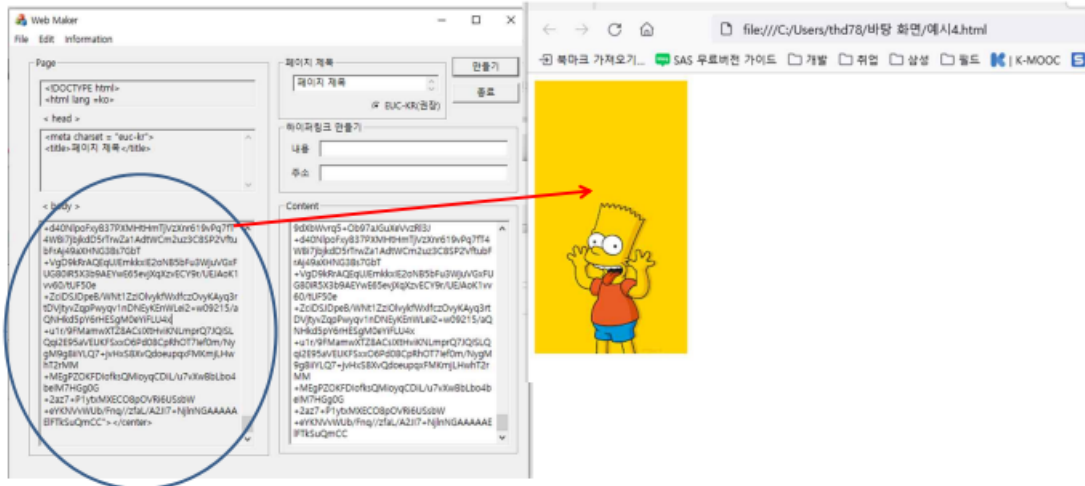
Menu – Edit / 이미지 – 웹 이미지 첨부하기

- HTML의 이미지 첨부 태그인 를 구현



Menu – Edit / 이미지 – 웹 이미지 첨부하기

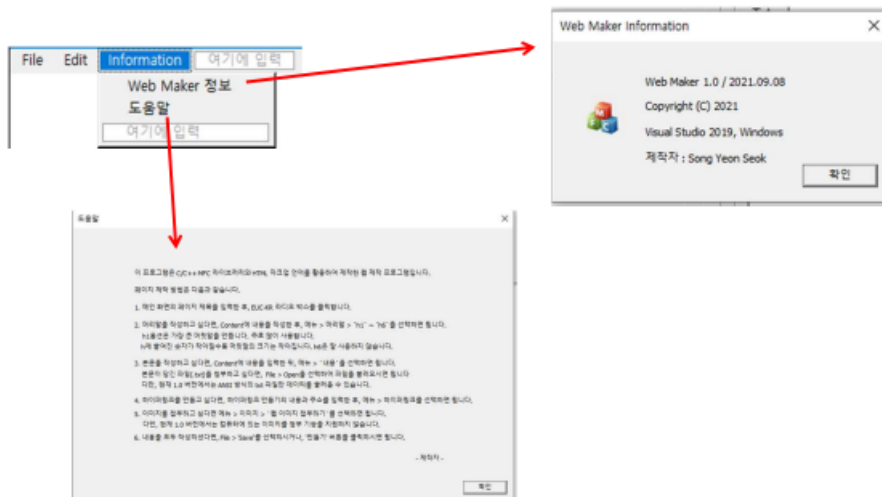
- HTML의 이미지 첨부 태그인 를 구현



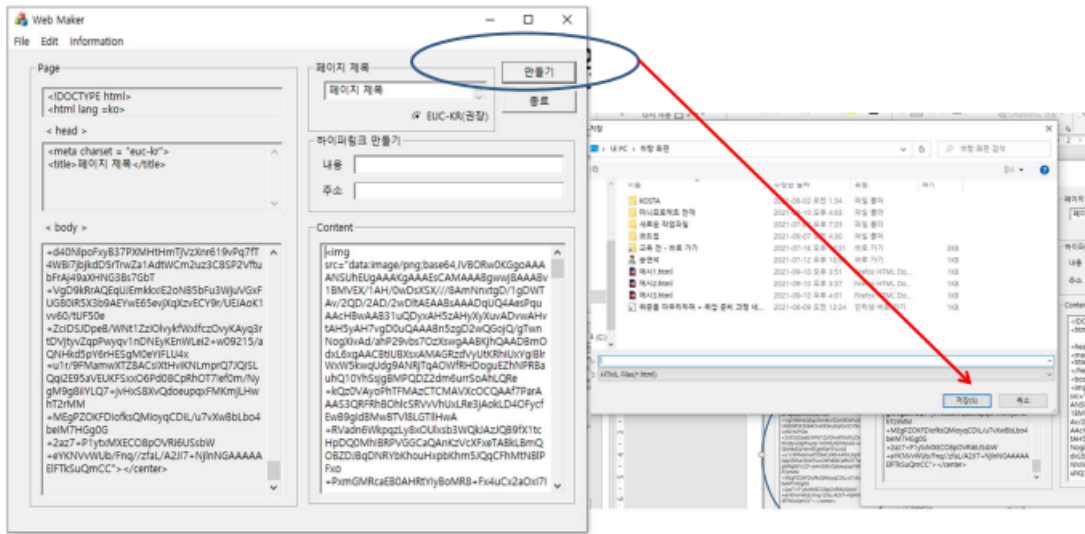
Main Menu - Information

- Web Maker 정보 : 프로그램 제작 버전, 제작자 등의 프로그램 정보가 담긴 Dialog 실행
- 도움말 : 프로그램 사용 방법이 작성된 Dialog 실행

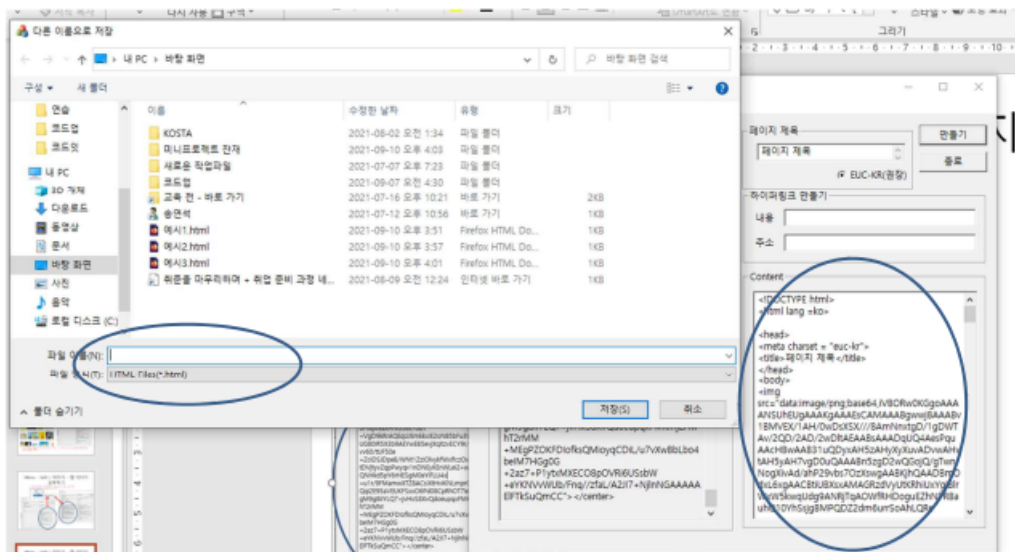
Menu – Information



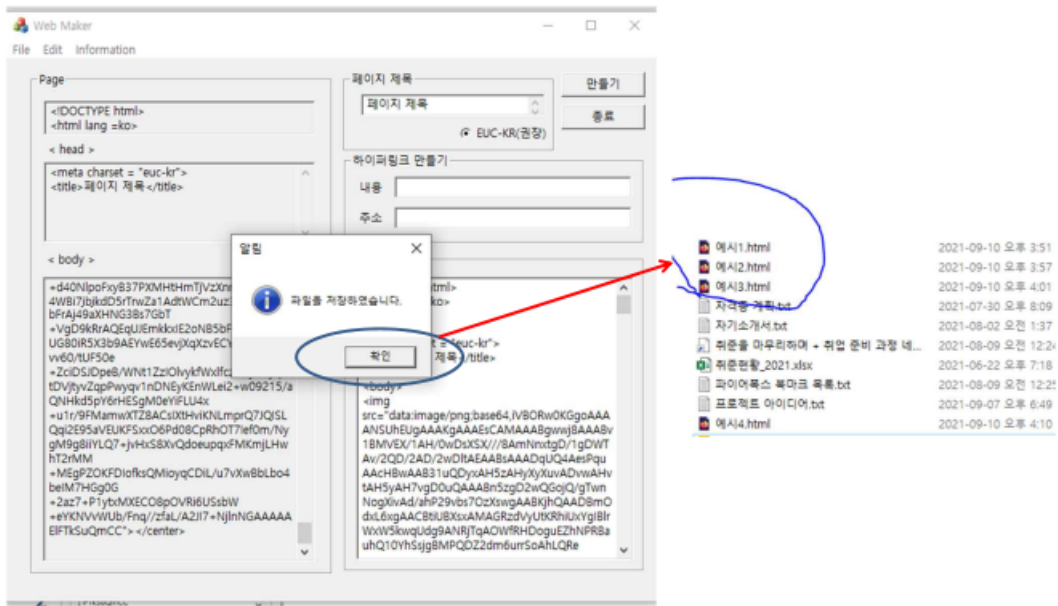
「만들기」 버튼 구현 원리



「만들기」 버튼 구현 원리



「만들기」 버튼 구현 원리



결론 및 느낀 점

- 초기 계획으로는 HTML과 CSS 기능을 구현하는 것이었으나, 5일이라는 짧은 개발 기간에서는 HTML 기능을 구현할 수 있었지만, CSS 기능을 구현할 수는 없었다.
- 첫 프로그램 기획 및 개발 경험으로, 처음에 기획할 때 가능한지 아닌지 의문이 들었는데, 5일이라는 개발 기간 동안에 열심히 노력하다 보니 원하는 프로그램을 개발할 수 있게 되었다.
- 이 프로그램을 기획하게 된 배경으로, 대학교 때 웹 페이지를 개발했던 적이 있다. 이때 HTML과 CSS에 대해 전혀 모르는 상태였기 때문에 개발하는 데 힘들었던 적이 있었다. 그때 파워포인트 프로그램 처럼, HTML과 CSS 언어 없이도 웹페이지를 만들 수 있는 프로그램이 있으면 좋겠다는 생각을 하게 되었고, 우연히 기회가 생겨서 Web Page Editor 프로그램을 개발하게 되었다.
- 이 경험을 통해서 개발자와 나의 성향이 일치한다는 사실을 알게 되었고, 불편한 점을 프로그램을 개발함으로써 해결하는 프로그램 개발자가 되겠다는 다짐하게 되었다.

향후 계획

- 현재 MFC 프레임워크는 마이크로소프트에서 지원되지 않는 프레임워크이다. MFC를 사용하는 업체들은 있지만, 새로 개발하는 프로그램에 적용하는 것보다 유지보수하는 데 사용한다는 사실을 알게 되었다.
- MFC와 달리 WinForm과 WPF, QT, 플러터 등 윈도우 애플리케이션을 개발할 수 있는 라이브러리는 매우 많았다. 이 프로젝트는 계속 진행하고 싶어했기 때문에 MFC 대신에 WinForm, WPF, QT 중에 고르는 것으로 결정하였다.

프레임워크 종류	내가 생각한 특징
.NET Windows Form	- C# 언어로 윈도우 프로그램을 개발할 수 있는 프레임워크 - 윈도우 프로그램을 개발한다고 하면 대부분 생각하는 프레임워크 - 윈도우에서만 작동함(윈도우를 벗어나면 호환성 및 정상 동작을 보장 못함) - MFC보다 생산성이 훨씬 높다.
WPF	- Windows Form과 마찬가지로 C# 언어로 윈도우 프로그램을 개발할 수 있다. - Windows Form로 만든 프로그램보다 더 자유로운 디자인을 구성할 수 있다. - Windows Form에 비해 생산성이 떨어진다.
QT	- C++, Python 등 여러 언어를 사용할 수 있다. - 여러 운영 체제에 적용할 수 있는 프레임워크 - 상업적으로 사용하기 위해서는 돈을 지급해야 함. - WinForm, WPF, MFC에 대한 구직자를 모집하는 공고는 본 적은 있지만, QT는 본 적이 없다.
MFC	- 지원 끊김 - MFC만의 개발 방식으로 인해 스파게티 코드가 만들어질 가능성이 높다. - 생산성이 낮지만, 제조업 및 고성능 처리 분야에서 사용하고 있다. - 현재로는, MFC 개발자가 되는 것은 매우 비효율적
Win32 API	- Low Level의 윈도우 개발 프레임워크 - MFC 이전에 사용했던 프레임워크로, MFC보다 더 고성능을 요구해야 하는 경우에 사용된다.(그렇게 알고 있음)

- 이를 보았을 때, MFC, Win32 API를 통해 프로그램을 개발하는 것 보다, WinForm 프레임워크를 사용하여 프로그램을 개발하는 것이 효율적이라고 판단된다.
- 추가로, C++언어로 Windows Form 프레임워크를 사용할 수 있다. 이 점이 맘에 들어 최종적으로 C++ 언어로 Windows Form 프레임워크를 사용하는 것으로 결정하였다.