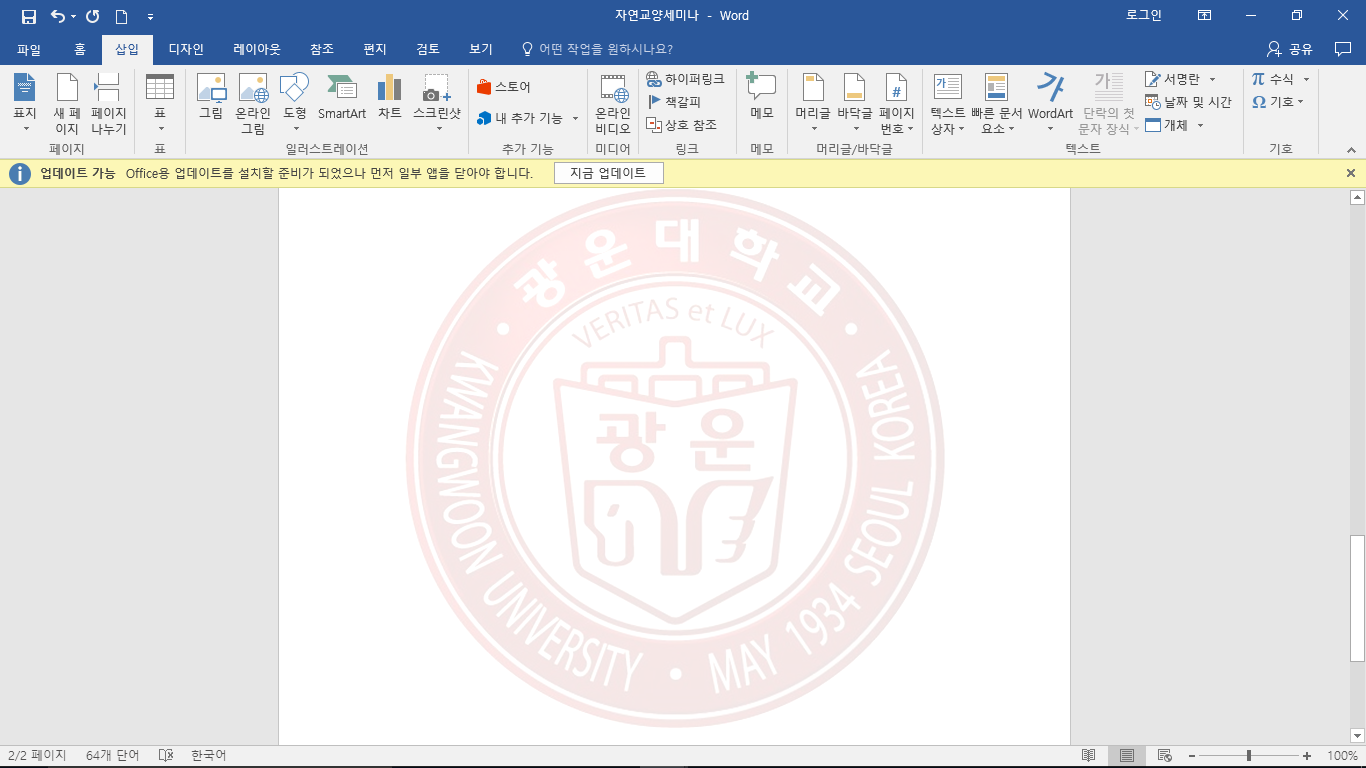
|  |  |
| --- | --- |
| 강의:웹프로그래밍 | |
| 교수님: 서두옥교수님 | |
| 2 0 1 8 2 0 4 0 9 4 | 박형빈 |

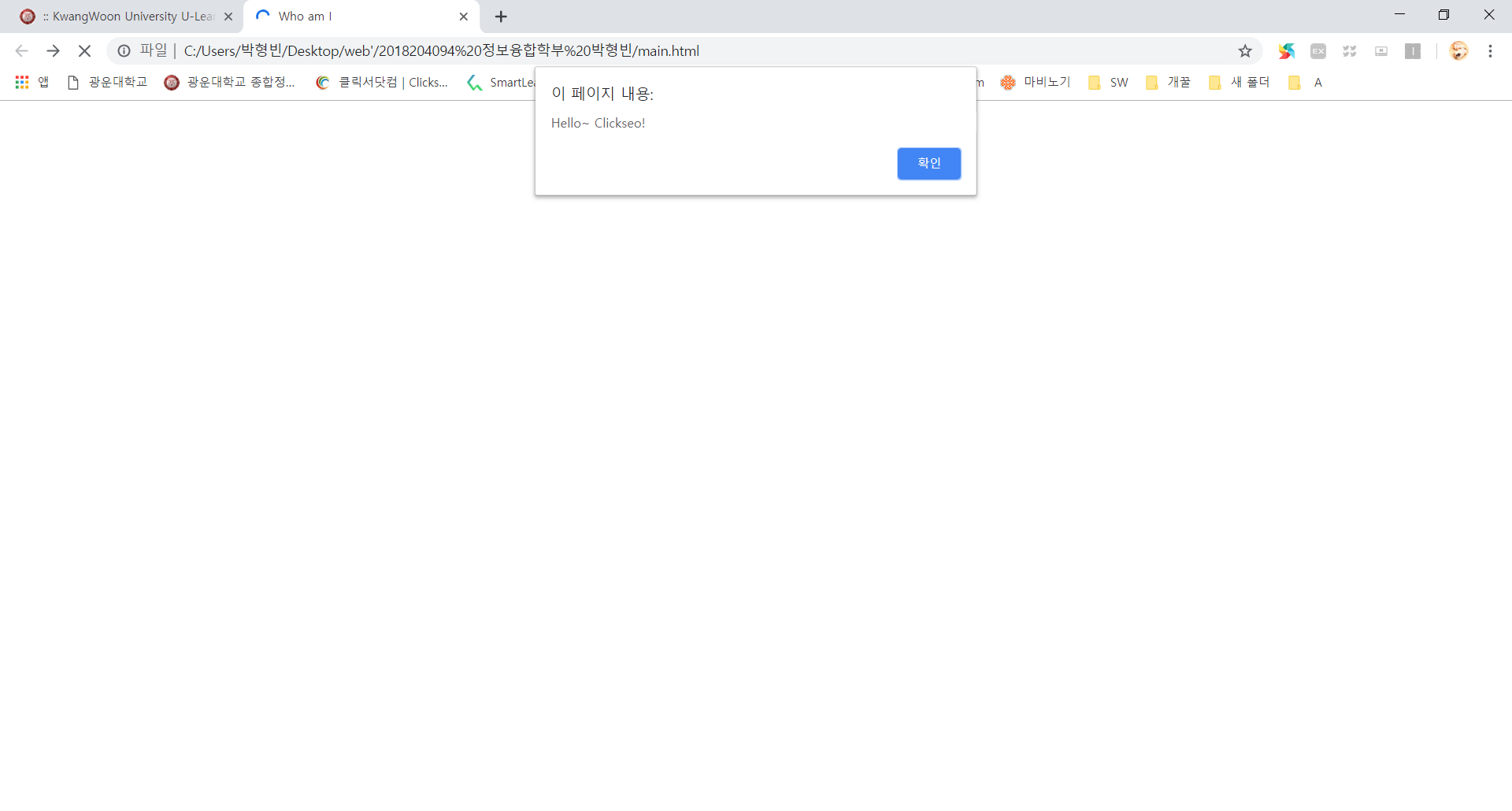
**REPORT**



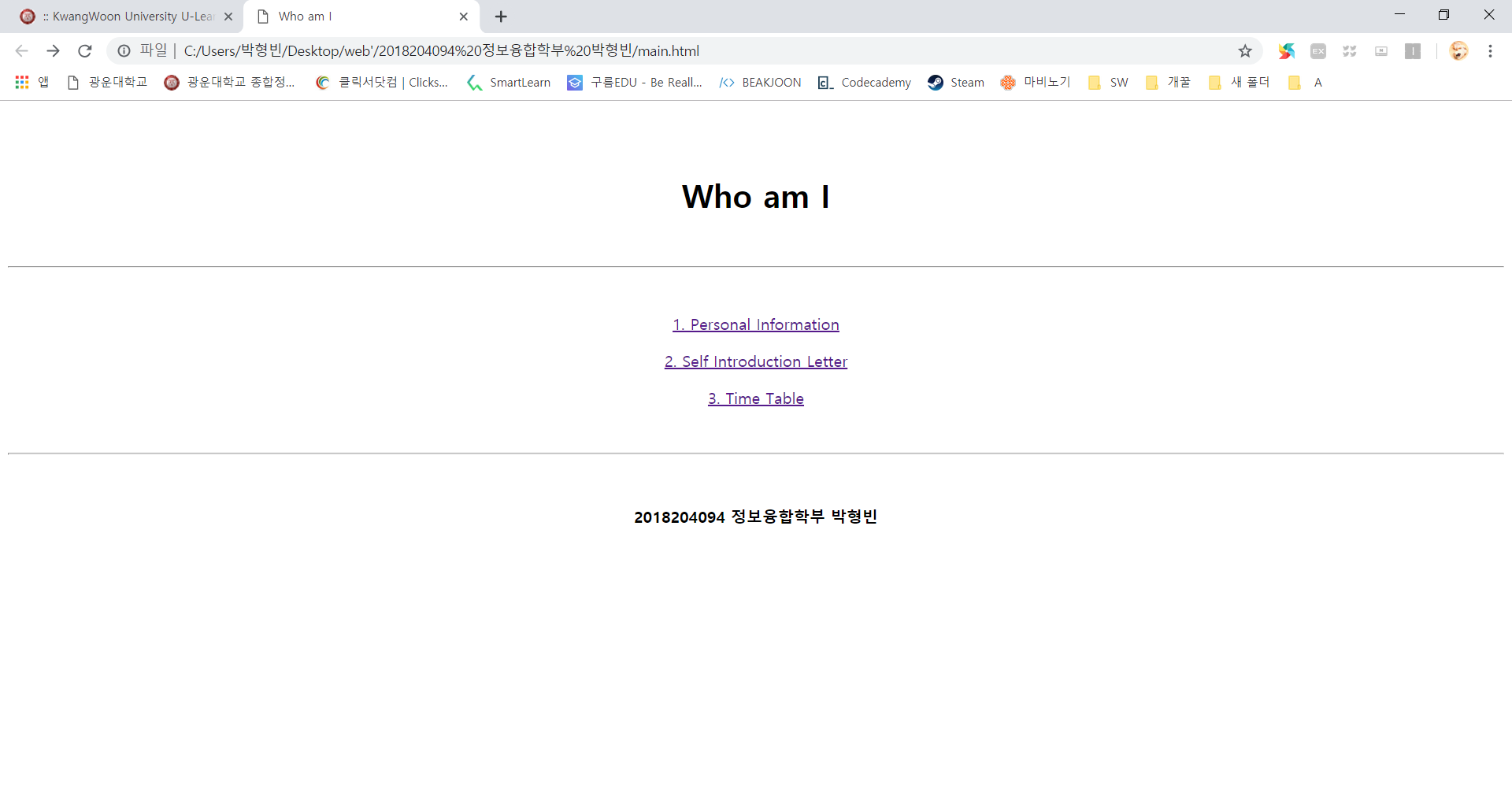
**웹페이지 제작과**

**구글 크롬 개발자 도구**

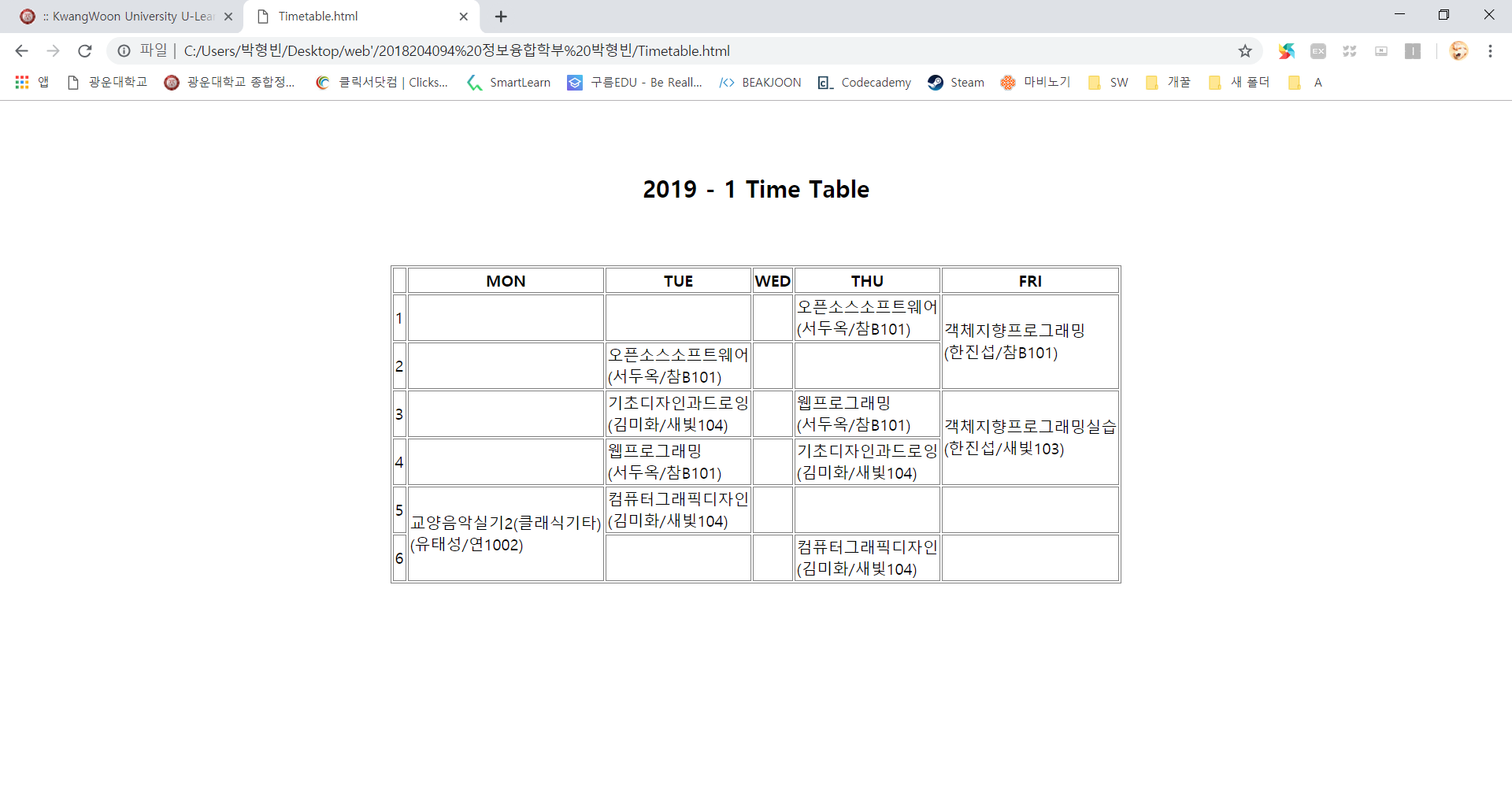
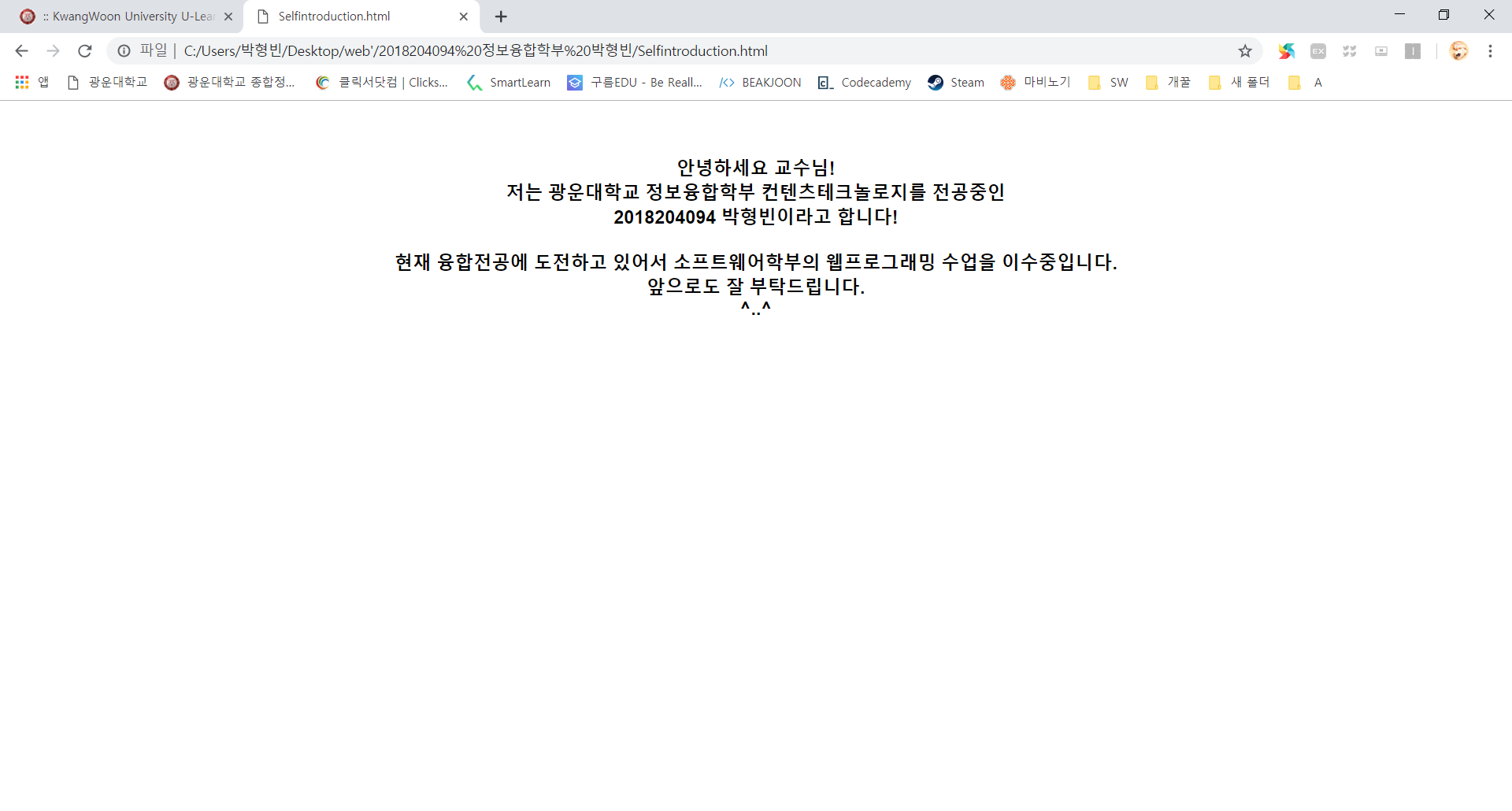
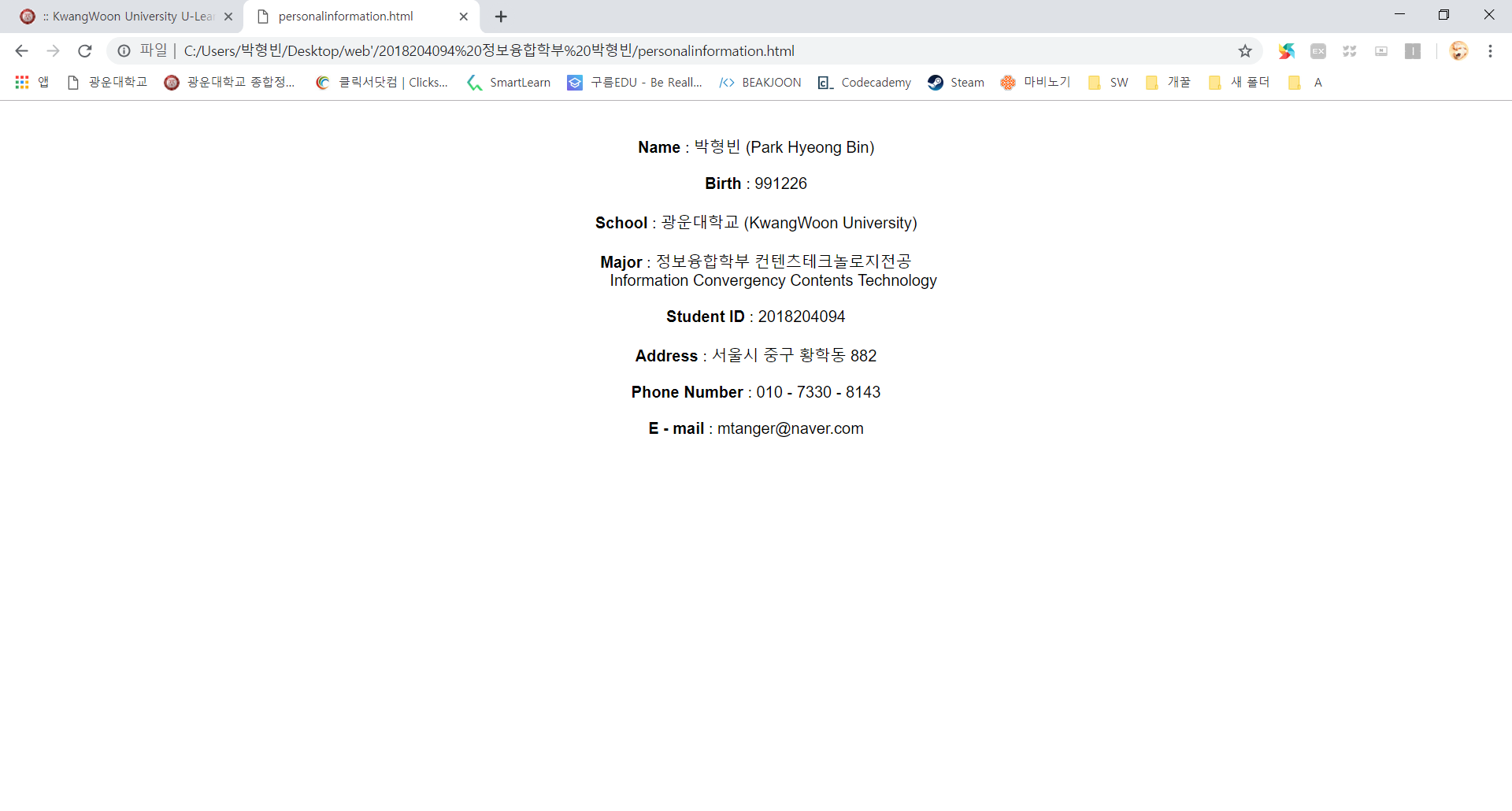
**#01. 웹페이지 제작 : 자기소개**



완성된 자기소개 페이지의 모습이다. 사이트에 들어가면 Hello~ Clickseo!의 알림창과 함께 접속되어진다.

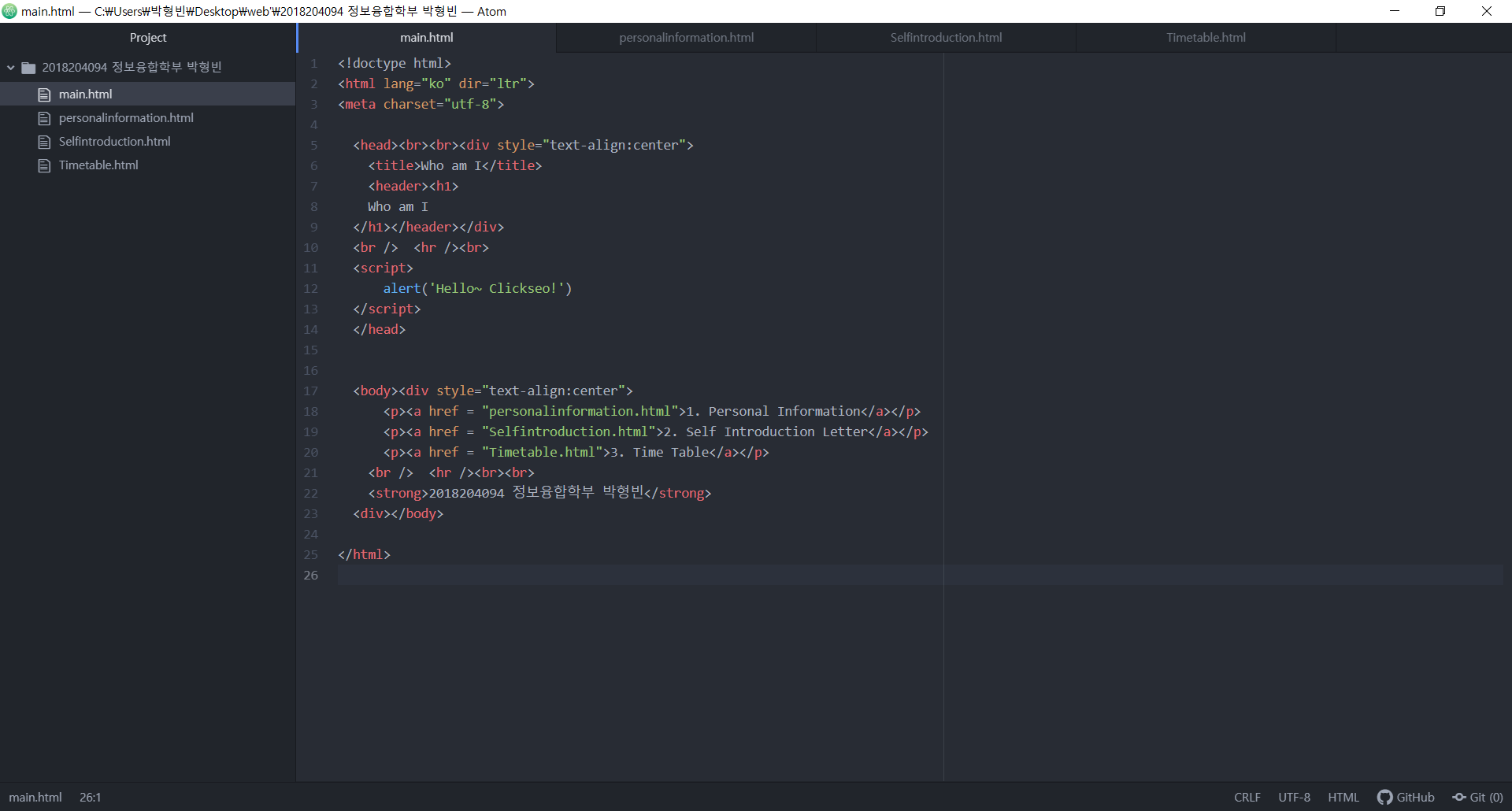


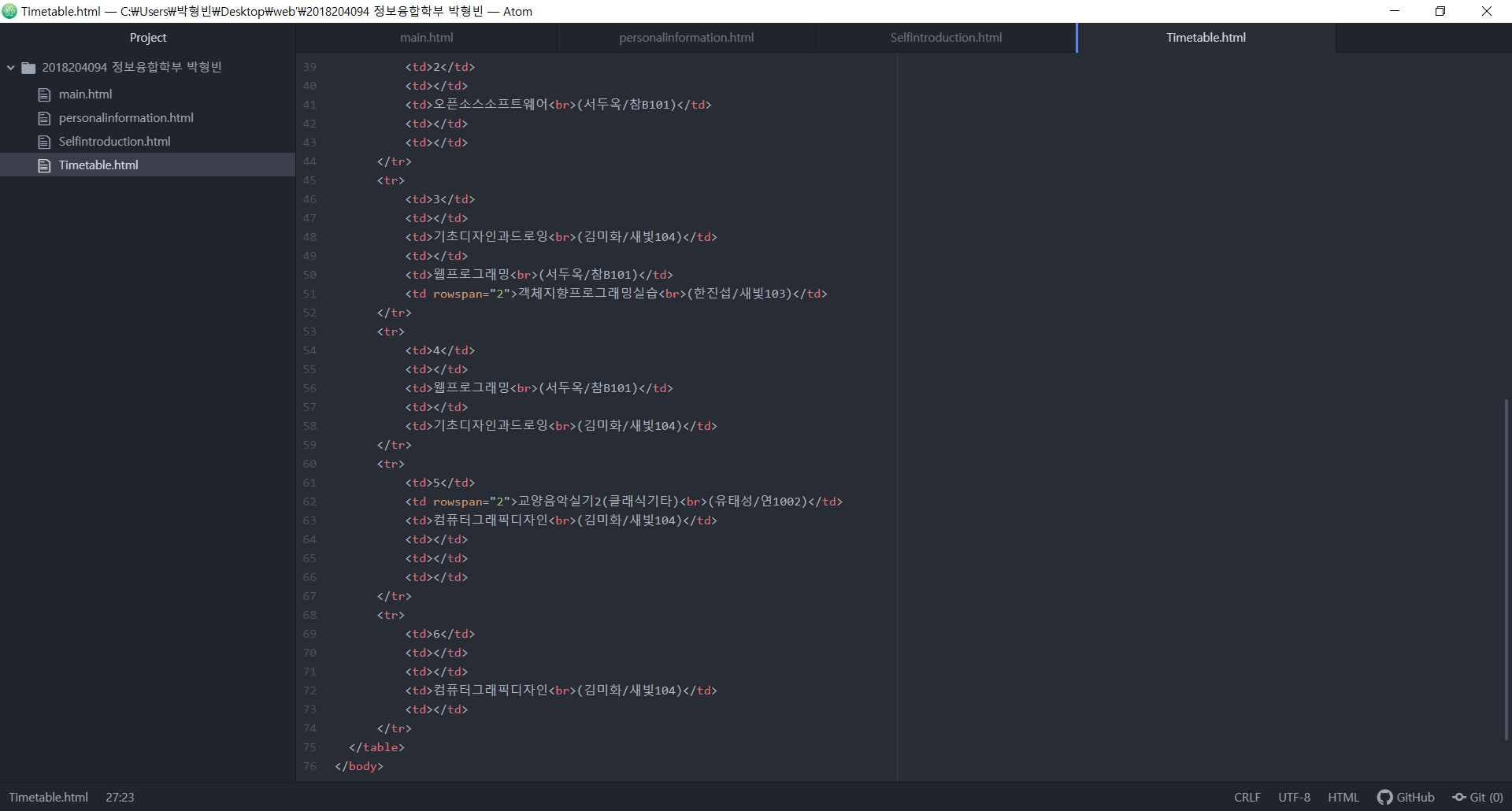
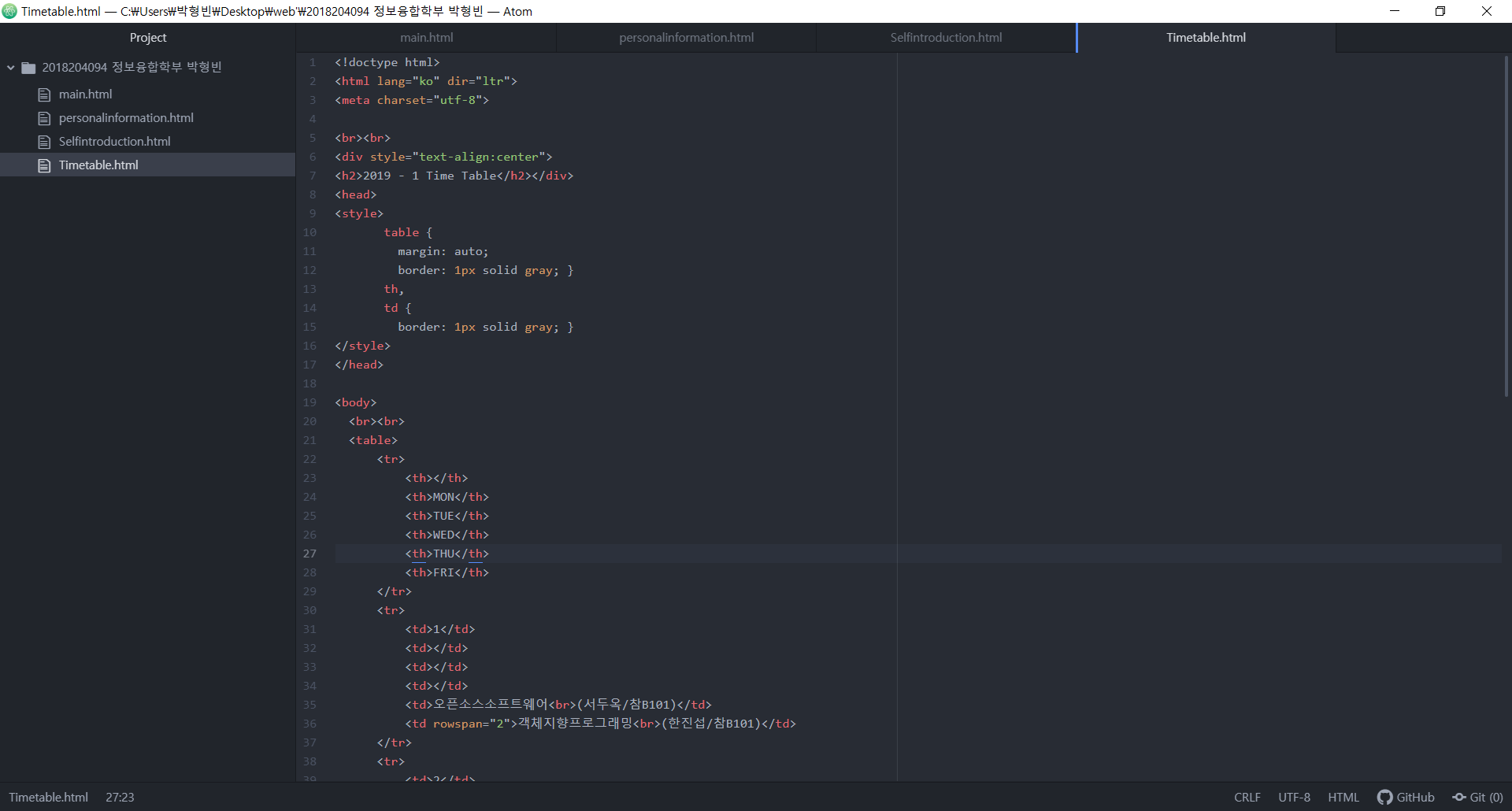
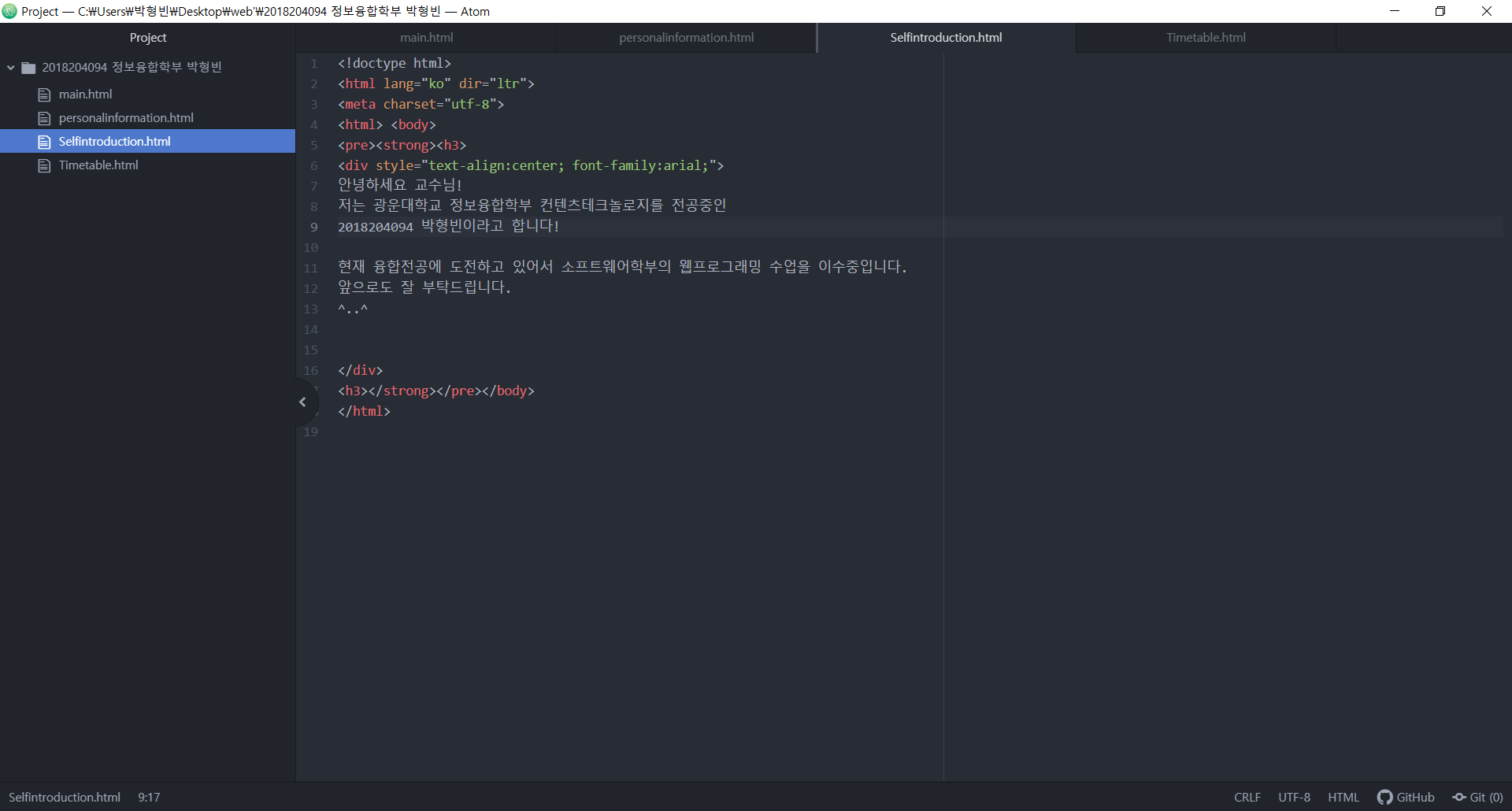
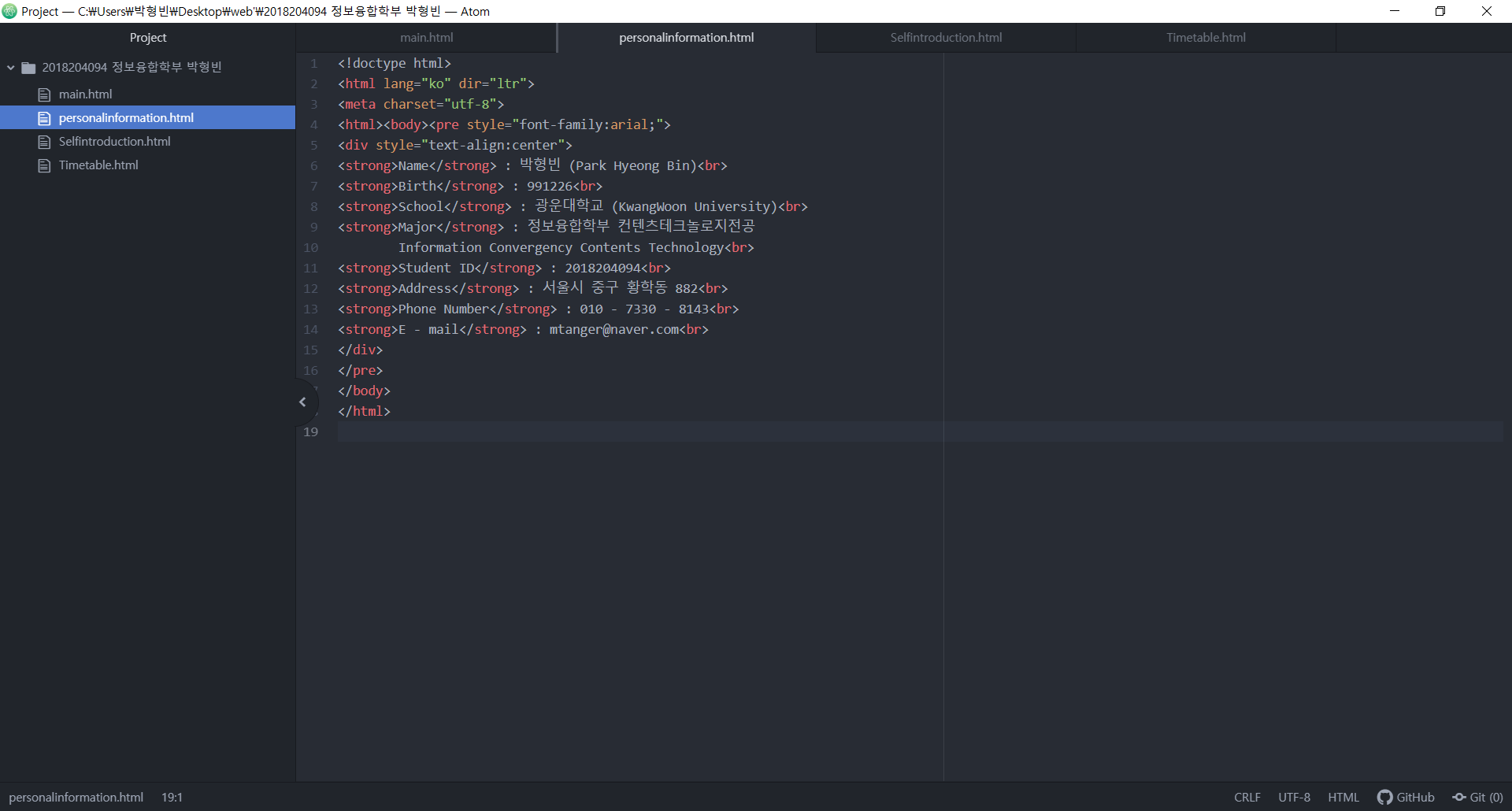
Who am I 라는 글씨와 본인의 학번, 학과, 이름과 함께 인적 사항(Personal Information), 자기소개서(Self Introduction Letter), 1학기 시간표(Time Table)를 링크로 달아놓았다.



각 사진은 세 가지 링크에 걸려있는 페이지이다. 첫 번째 링크에는 간단한 인적사항을, 두 번째 링크에는 간단한 자기소개를, 세 번째 링크에는 이번 학기 시간표를 넣었다.

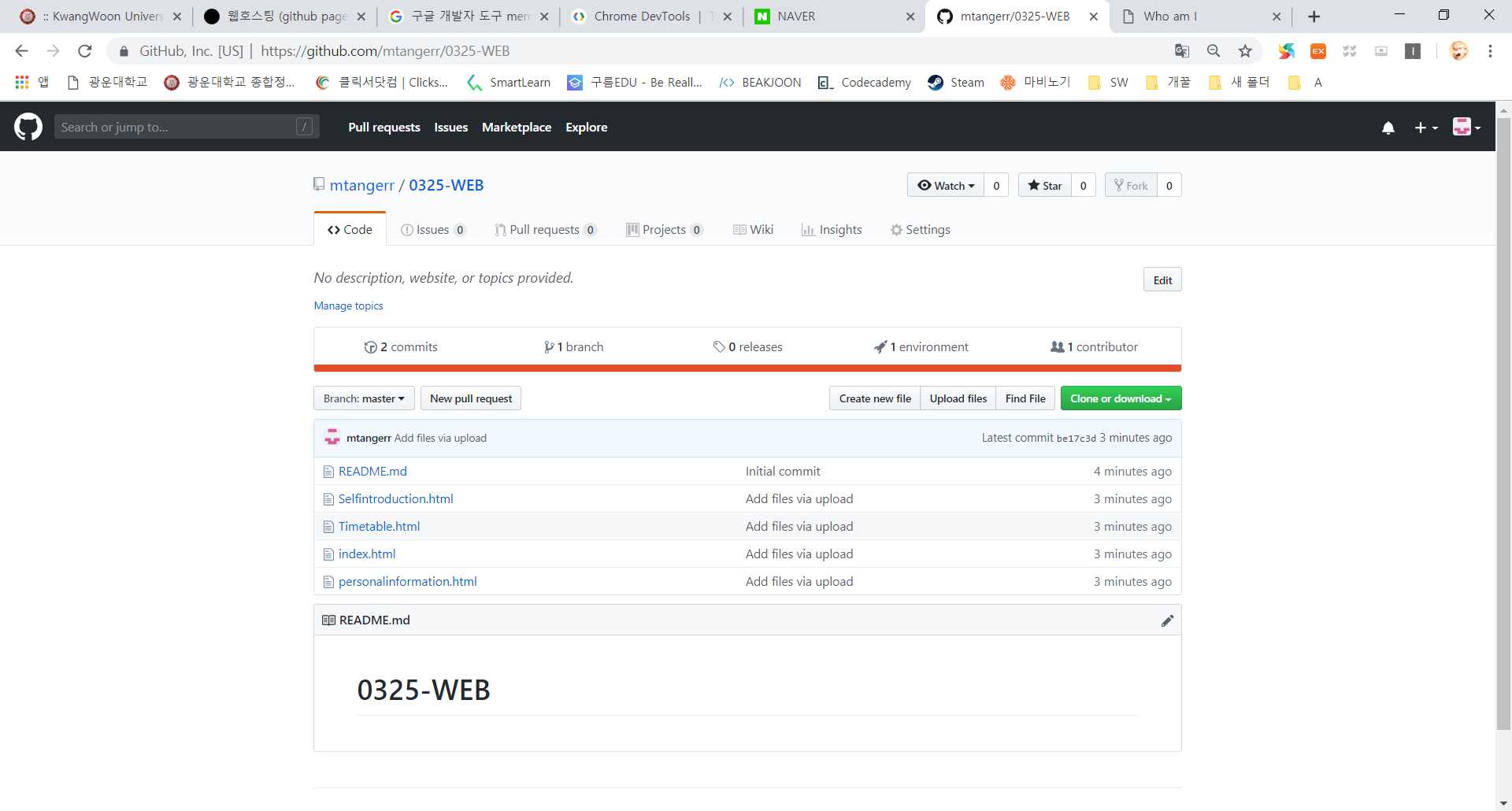
밑에는 github.com에서 만든 Atom(아톰)에디터를 통해 작성한 소스코드이다.

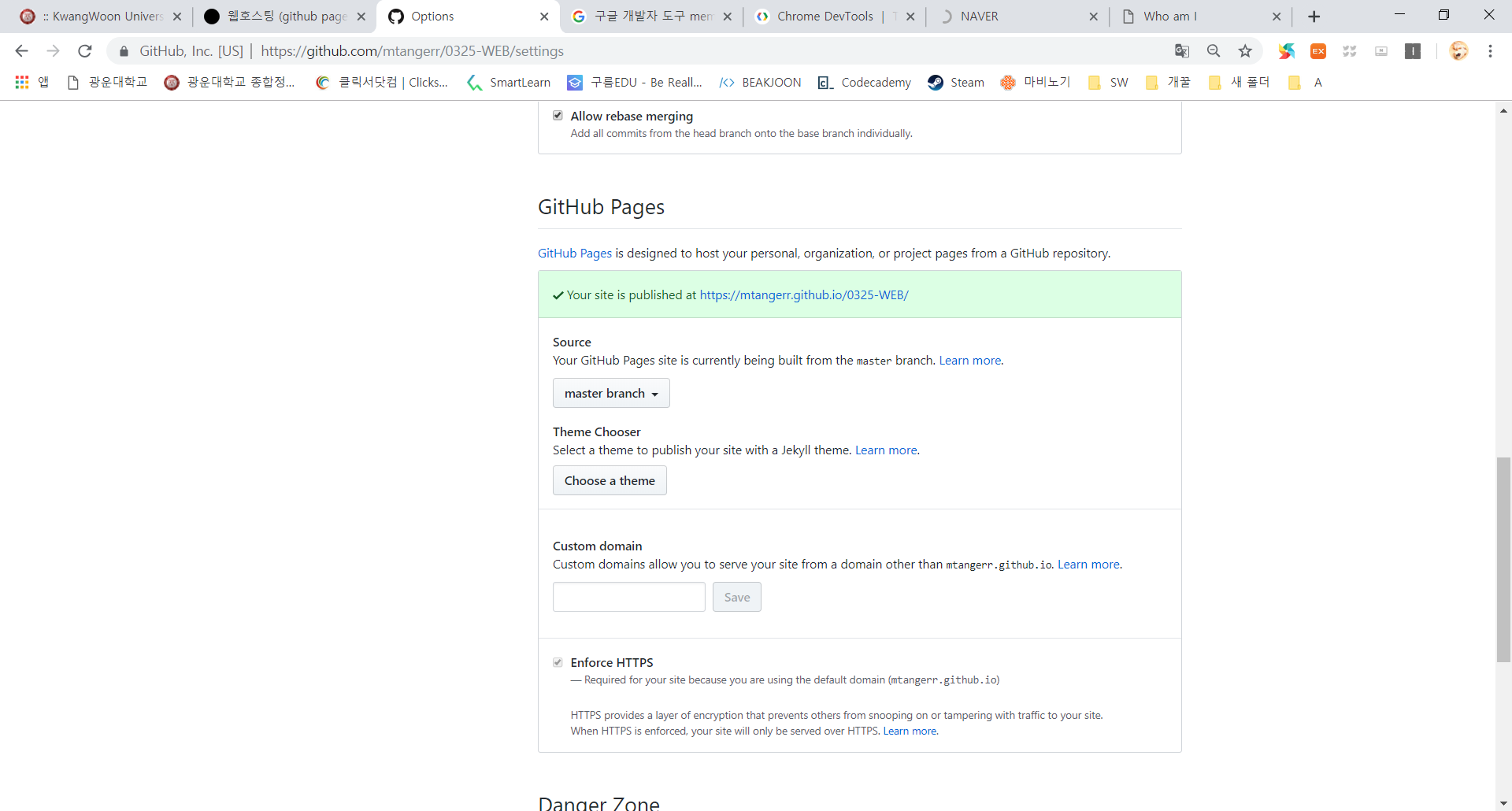




위의 소스코드에는 HTML, CSS, JavaScript로 구성되어있다. 사이트 접속시 띄워지는 알림창을 JavaScript로, <style>태그를 통해 시간표를 구성하는데에 CSS, 전반적인 작성은 HTML로 이루어졌다.

**#02. 웹 호스팅**

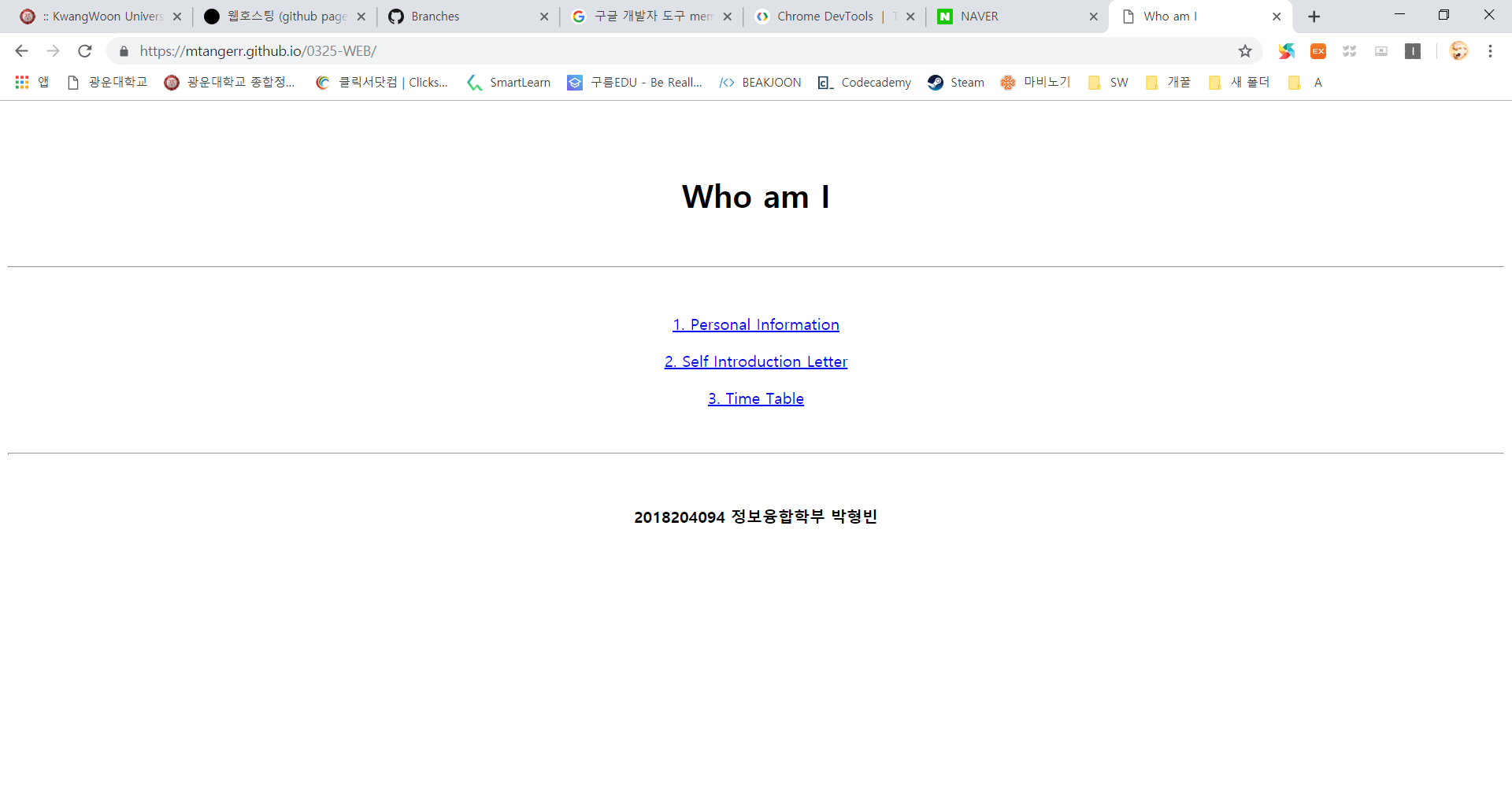
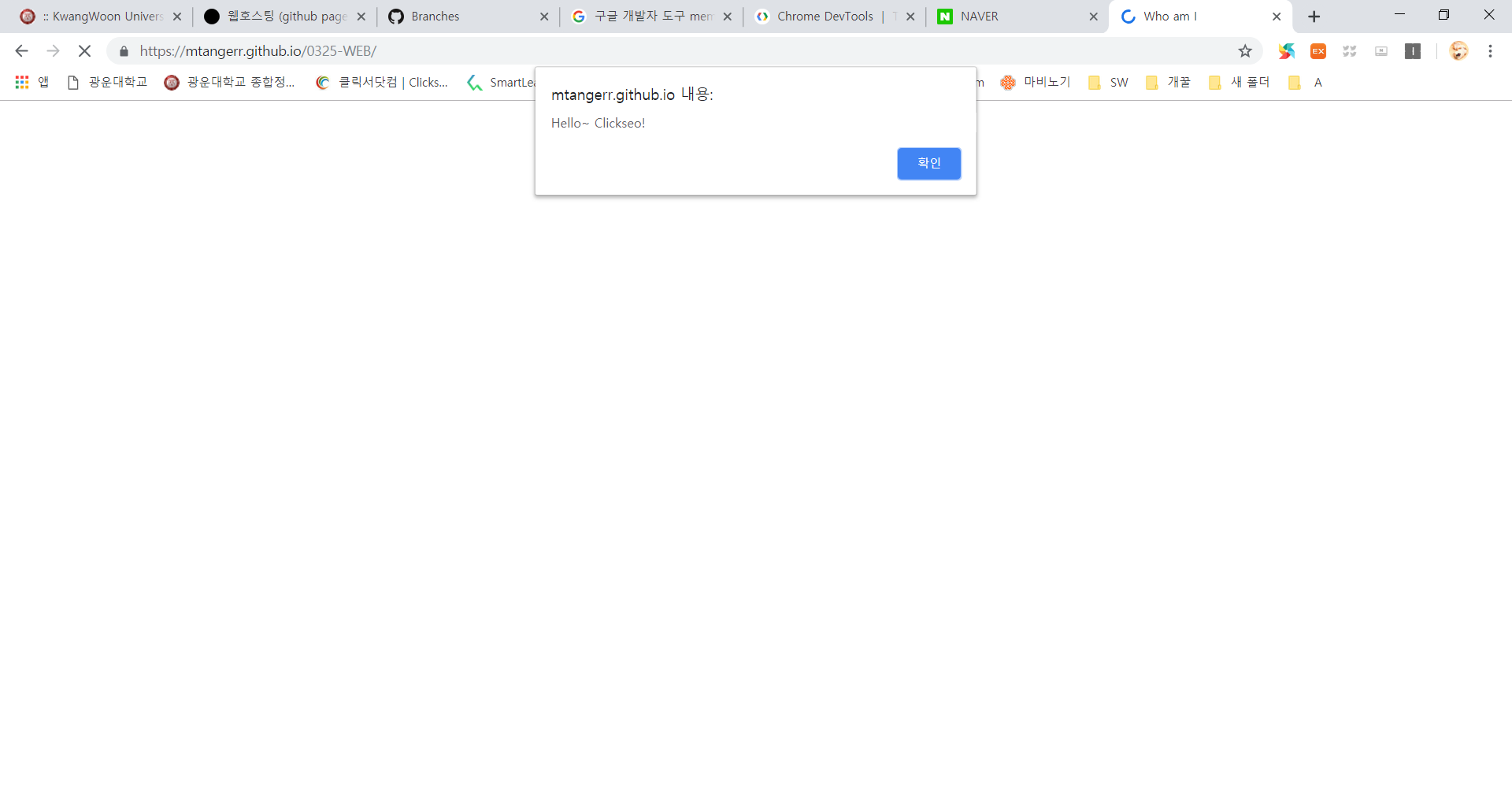
Atom 에디터를 통해 만든 자기소개 사이트를 이제는 웹을 통해서 호스팅하였다. 웹호스팅 업체로는 github.com을 이용하였고 이 사이트를 이용하기 위해 간단한 가입 절차를 진행하였다. 

New reposity 버튼을 통해서 소스코드를 저장하는 보관소를 생성하였고 이름은 0325-WEB이라 지정하였다. 이후 index.html과 링크가 걸린 페이지를 작성한 3개의 HTML파일들을 업로드하였다. 

저장소에 저장된 소스코드, 웹페이지를 인터넷을 통해서 서비스하기 위하여 Settings 탭을 들어가였다. Gibhub Pages 항목에서 master branch를 선택하고 Select branch 중 master branch를 선택 후 저장하였다. 이후 만들어진 주소는 다음과 같다.

[**https://mtangerr.github.io/0325-WEB/**](https://mtangerr.github.io/0325-WEB/)

실제 인터넷 브라우저에서 위 주소를 입력하면



성공적으로 접속되는 것을 확인하였다. 웹호스팅을 하는 일련의 과정은 생활코딩 사이트의 WEBn- 웹호스팅(github pages)을 참고하였다.

**#03. 구글 크롬 개발자 도구**

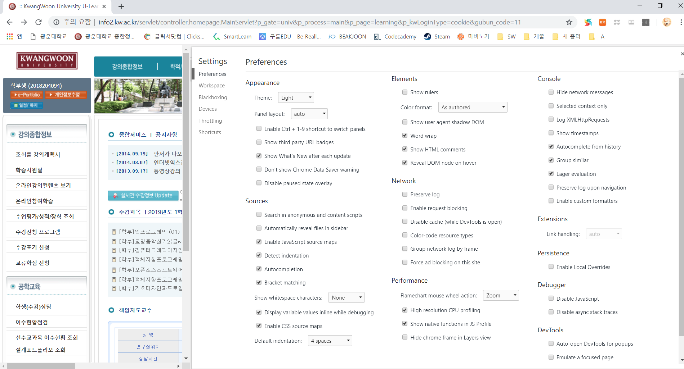
크롬 개발자 도구란 구글에서 만든 웹브라우저인 크롬에 있는 개발을 도와주는 다양한 도구가 기본적으로 제공되며 이를 개발자 도구라고 한다. 개발자 도구를 이용하면 HTML,CSS, JavaScript의 생산성을 극대화 할 수 있다. 크롬 뿐만 아니라 인터넷 익스플로러, 파이어 폭스 등 대부분의 웹브라우저에서 지원하고 있는 기능이지만 그 중에서도 구글 크롬의 개발자도구가 가장 속도가 빠르고 기능도 뛰어나다. 이러한 이유 때문에 많은 웹 개발자가 웹브라우저에서 실시간으로 웹 사이트를 분석해야 하는 필요가 있는 경우에 주로 크롬 개발자도구를 이용한다.

크롬 개발자도구는 1) [Chrome맞춤 설정 및 제어] – [도구 더보기] – 개발자 도구, 2) Ctrl + Shift + I, 3) F12단축키를 통해 시작할 수 있다.

**1) Setting**

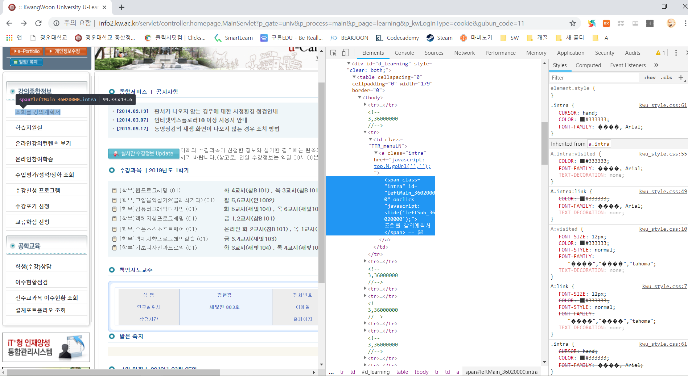
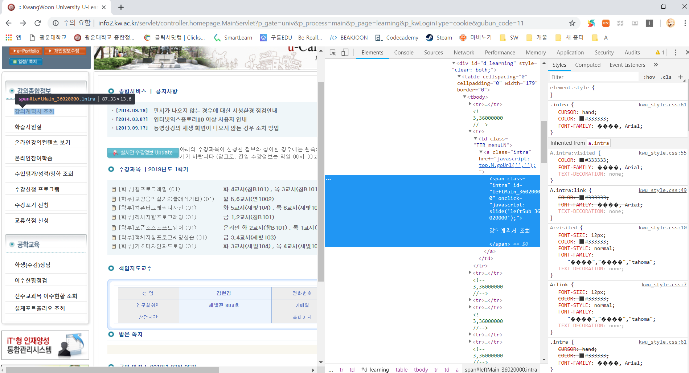
구글 개발자 도구의 Setting 설정 창에서 유용한 설정들을 한 번에 다룰 수 있다. 다양한 설정들을 실습해본 결과 몇 가지 기능들을 추려보았다.

개발자 도구 창의 테마 변경(Theme Light/Dark), 새로고침과 같은 캐시를 참조하는 일을 방지하는 캐싱 사용하지 않기(Disable Cashe), 콘솔패널 타임스태프 활성화를 통해 console log 표기(Show timestamps), 자바스크립트 디버깅에 유용하게 쓰이는 기능으로 디버거로 에러 스택을 추적하는 과정에서 불필요한 라이브러리 코드를 건너뛰는 기능(Blackboxing), 크롬 개발자 도구의 단축키 정보(Shortcuts) 등 개발자의 기호에 따라 설정을 조절할 수 있다.



**2) Element 패널**

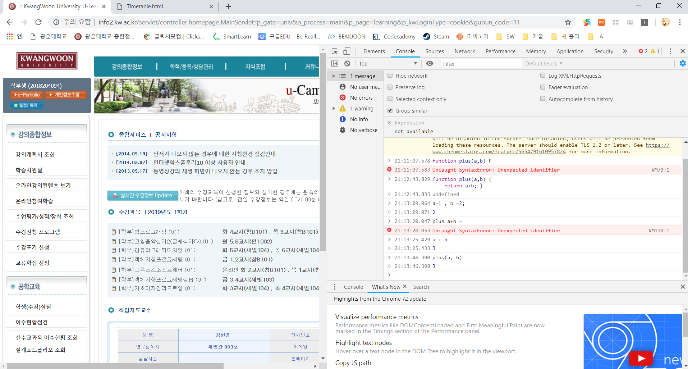
구글 개발자 도구 중에서 엘레멘트 편집기는 현재 자신이 보고 있는 웹페이지의 HTML 과 CSS 속성들을 마음대로 바꿀 수 있다. 자신이 바꾸고자 하는 속성을 FTP 업로드 없이 웹사이트에 실제 영향을 주지 않으면서 확인할 수 있다.



<수정하고자 하는 HTML요소를 더블클릭하여 수정해본 화면>

**3) Console 패널**

현재 편집중인 페이지의 로그 메시지를 확인할 수 있는 패널이다. JavaScript를 실시간으로 편집, 실행, 확인할 수 있다. 콘솔 기록에 작업에 관해서 콘솔창에 clear(), console.clear()의 호출을 통해 콘솔 기록을 지울 수 있고, 콘솔 맨 위에 있는 Preserve log 확인란을 활성화하여 페이지 새로고침 또는 변경 사이에 콘솔 기록을 유지할 수 있으며, Save as 선택 시 콘솔의 출력을 로그 파일에 저장할 수 있다. 이외에도 메시지 스태킹이라 하는 기술이 있다. 이는 메시지가 연속으로 반복되는 경우 메시지의 각 인스턴스를 새로운 줄에 출력하지 않고 콘솔이 메시지를 ‘스택’하고 왼쪽 여백에 숫자를 대신 표시한다. 해당 숫자가 메시지의 반복 횟수를 나타내는 것이다.



<간단한 함수를 만들어서 적용해 본 화면>

**4) Source 패널**

소스를 편집하거나 Break Point를 걸어 디버깅할 수 있다. 원하는 줄에 break point를 걸고 디버그 도구를 이용하여 디버깅하면 해당 시점의 모든 변수 값을 시간 내에 검사할 수 있다. 지속적으로 관찰하고 싶은 변수는 Watch탭에 두고 값의 변화를 관찰할 수 있으며 디버깅 시점의 Scope내의 모든 값을 찾아볼 수 있다. 이는 곧 console.log() 메서드보다 더 빠르게 버그를 찾고 수정하는데에 도움이 된다.

**5) Network 패널**

애플리케이션에서 발생하는 각각의 네트워크 동작에 대한 정보를 기록한다. 정보는 상세한 시간 정보와 HTTP 요청, 응답 헤더, 쿠피, 웹소켓 데이터 등이다. 네트워크 패널은 별 다른 설정을 하지 않아도 개발자 도구가 오픈되어 있는 동안에는 자동으로 모든 네트워크 관련 동작들을 기록한다. 보고 싶은 정보만 볼 수 있으며 정보들을 정렬시킬 수도 있다. 더 상세한 필터링을 통해서(filter queries) 원하는 정보들을 선택적으로 관찰할 수 있다.

**6) Perfomance 패널**

사이트의 라이프사이클 동안 발생한 다양한 이벤트를 기록하고 탐색하여 페이지의 런타임 성능을 향상시킬 수 있다. 이는 사이트나 어플리케이션의 인지된 성능 문제를 조사하는데에 도움이 된다. 간단하게 Overview 창을 통해 초당 프레임 수 FPS, CPU 리소스, 색상의 막대를 통한 리소스를 나타내는NET을 확인할 수 있다. 기록을 만들고, 영역대를 선택해 선택 검색, 분석이 가능하다.

**7) Memory 패널**

메모리 사용 행태를 작성하고 누수를 탐색하는 탭이다. 페이지 성능에 영향을 미치는 메모리 문제를 찾아내는 방법을 알 수 있다. 메모리 누수, 팽창 및 잦은 가비지 수집 등이 대표적이다.

**8) Application 패널**

IndexedDB와 웹 SQL 데이터베이스, 로컬 및 세션 스토리지, 쿠키, 어플리케이션 캐시, 이미지, 폰트, 스타일시트를 포함한 로딩된 모든 리소스를 검사한다.

**9) Security 패널**

HTTPS는 웹사이트와 해당 웹사이트를 믿고 개인 정보를 제공하는 사람들 모두를 위해 [중요한 보안 및 데이터 무결성](https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https?hl=ko)을 제공한다. 구글 개발자도구의 Security 패널은 보안 문제를 디버그하고 웹사이트에 HTTPS를 적절히 구현했는지 확인할 수 있다.

위 내용들은 구글에서 제공하는 Tools for Web Developers 에서 Chrome DevTools 사용법을 참고하여 작성하였다.