



전반 3시간

## ✓ 체크인

### [0.5시간] 자기소개 & 출석체크



튜터님은 체크인과 함께 [출석 체크\(링크\)](#)를 진행해주세요.

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

#### ▼ "30초 체크인"

- 튜터님은 타이머를 띄워주세요! ([링크](#))
- 매주 한 주간 어떤 일이 일어났는지, 숙제와 복습은 어땠는지 이야기합니다. 같이 공부하는 우리는 동료입니다. 동료의 이야기에 귀기울여들어주세요.
- 오늘은 첫날이니 이름, 스파르타코딩클럽 오게 된 계기, 배워가고 싶은 것을 이야기해주세요. (30초가 넘어도 괜찮습니다 😊)  
예) 안녕하세요, 김로탄이에요. 왕초보예요. 코딩하는 사람으로 살고 싶어 찾아왔어요. 웹 크롤링을 배워서 내가 좋아하는 영화 배우 소식을 보여주는 프로젝트를 만들고 싶어요.

#### ▼ 8주간 수업 소개

##### ▼ 수업 과정

1) 1~5주차에는 3가지 프로젝트([나홀로메모장](#), [나홀로북리뷰](#), [マイムビースタ](#))을 만들며 학습하고, 숙제로 원페이지쇼핑몰을 만들며 복습합니다. 즉, 네 번을 반복학습합니다.

- 1~2주차 [프론트엔드]: 웹 기초 동작 원리를 이해하고, 유저가 보는 부분인 웹 화면을 만듭니다. 새롭게 배우는 기술은 [HTML](#), [CSS](#), [Bootstrap](#), [Javascript](#), [jQuery](#), [Ajax](#), [API](#) 기초입니다.
- 3주차 [파이썬 응용] : 파이썬을 가지고 놀아보며 데이터베이스 사용을 배웁니다. 파이썬으로 할 수 있는 웹 스크래핑(크롤링), API로 정보 가져오기, 정보를 저장하는 데이터베이스를 배웁니다. 새롭게 배우는 기술은 [웹 스크래핑\(크롤링\)](#), [OpenAPI 활용](#), [MongoDB](#), [Git](#)입니다.
- 4~5주차 [백엔드/프론트엔드]: API와 서버(백엔드)를 만들면서, 화면(프론트엔드)과 서버(백엔드)를 연결해봅니다. 새롭게 배우는 기술은 [Flask Framework](#)입니다.

##### 2) 6~8주차 [[나만의 웹서비스 만들기](#)]

- 목표를 가지고 그동안 배운 것을 내 프로젝트에 적용해봅니다. 누구나, 언제나 내가 만든 서비스에 접근할 수 있도록 하는 '배포'를 배웁니다. 새롭게 배우는 기술은 [배포 과정](#), [AWS\(클라우드\)](#), [리눅스](#), [도메인 등록](#)입니다.

##### ▼ 수업 구성

1) 🧑💻 수업목표 : 이번주 집중해서 배울 것은 무엇일까요? 목표의식을 가지고 수업을 듣습니다.

2) ✓ 체크인, 체크아웃 : 우리 모두 스파르타코딩클럽! 수업 시작 전에는 한 주간 균황을 이야기하는 체크인, 수업 후에는 수업을 되돌아보는 체크아웃을 진행합니다. 8주동안 서로 도울 동료가 되어 서로에게서 찾은 좋은 점은 같이 시도해보고, 부족한 점은 함께 채워봅니다.

3) 🧑💻 배우고 적용하기 : 인생은 실전! 튜터에게 먼저 개념을 배운 후, 같이/스스로 실습합니다. 전반부와 후반부로 구성되어 있습니다.

- 👩‍💻튜터와 같이 : 튜터님과 튜티들이 같이 코드를 따라치면서 차근차근 실습을 해볼 거에요.
- 💡실습하기 : 스스로 먼저 해 본 다음에 튜터님과 같이 문제를 풀어봅니다.
- 😎문제뱅크 : 조금 더 실습하고 싶은 분들을 위해 수준별로 실습문제를 풀 수 있도록 '😎문제뱅크'를 만들어두었습니다.

4) 🎂 5분 꿀팁 : 나도 개발자 마인드 탑재! 수업 진도 전, 개발자답게 사고하는 습관을 기를 '5분 꿀팁 개발자편'을 배웁니다.

5) 🤩소화타임 : 바로바로 복습합시다! 튜터는 숙제 설명을 하고, 필요에 따라 그날 배운 내용을 주요 키워드 중심으로 복습 설명합니다. 숙제를 시작하고, 스스로 복습하며 튜터에게 질문합니다.

### ▼ 스파르타 코딩클럽 활용하기

- 1) Slack에서 소식 나누기 : 혼자 공부하면 외롭죠? 개발 질문, 나만의 꿀팁 모두 환영합니다.
- 2) 금요일 오피스아워 / 리뷰세션에 참여 : 매주 금요일 열리는 오피스아워와 리뷰세션을 신청해서 동료 튜티들과 같이 코딩하고, 오피스지기 튜터에게 질문해보세요.
- 3) 영상강의로 복습하기 : 개념 복습할 땐 수업자료 상단에 있는 영상강의를 참고해보세요.

## 💡💻 배우고 적용하기 - 전반부

### 💡 5분 꿀팁 - 파일 관리



앞으로 수업 진도 전, 개발자답게 사고하는 습관을 기를 '5분 꿀팁 개발자편' 을 배웁니다.



앞으로 [💡튜터와 같이] 표시가 나오면,  
한 줄, 한 줄씩, 튜터와 튜티 모두 코드를 따라치면서 실습을 해볼 거에요.

### ▼ 개발자처럼 파일 관리하는 방법

- ❗ 개발자의 보물은 바로 코드! 내 보물이 어디있는지 모르면 난감하겠죠?

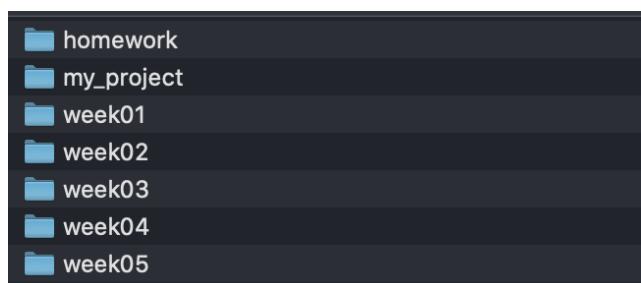
  - 1) 어떤 역할을 하는 폴더와 파일인지 한눈에 파악할 수 있게
  - 2) 다른 사람들과 협업할 때도 함께 정한 규칙대로 관리하면 유용하답니다.

#### ▼ 1) 파일/폴더 이름 짓기(naming) 기본 규칙

- 폴더/파일이 어떤 내용인지 파악할 수 있게 적기
  - 개발자는 이름짓기(naming)을 정말 중요하게 생각한답니다. 특히 웹 프로그래밍은 데이터를 주고받는 과정입니다. 어떤 데이터(내용)을 담고 있는지 한 눈에 알 수 있는게 좋아요. 앞으로 배울 내용에도 꾸준히 이름짓기(naming)에 대한 내용이 나올 거에요.
  - 예) a → 무슨 폴더지? / homework → 숙제 폴더구나
- 파일과 폴더 이름은 영어로 : 가끔 컴퓨터가 한글을 인식하지 못하는 경우가 있어요.
- 특수문자는 ❌만 사용하기 : 다른 특수문자(띄어쓰기 포함)을 컴퓨터가 알아듣게 하려면 조금 수고로워요. 우리는 단어를 연결할 때, ❌를 사용하겠습니다.

#### ▼ 2) [💡튜터와 같이] 수업 폴더 만들기

- 8주동안 사용할 수업 폴더를 만들어 보겠습니다.
- 바탕화면에 sparta 폴더를 만들어주세요. 폴더 안에는 아래와 만들어주세요.



- week01, week02, week03, week04, week05 : 매 주차 수업에서 배운 파일을 저장할 폴더
- homework: 숙제 폴더
- my\_project: 6~8주차 진행할 내 프로젝트

## [0.5시간] 개발 도구 사용 & 웹 기초 동작원리

### ▼ 1) [💡튜터와 같이] 개발자 도구 사용하기 - Pycharm Pro 등록 & 사용방법

 Mac & Window 모두 동일. 반 별 프로모션 코드는 투터님께 미리 드렸답니다!

1) <https://account.jetbrains.com/licenses> 에 접속해서, 로그인을 완료합니다.



## Welcome to JetBrains Account

-  **Access your purchases**  
and view your order history
-  **Identify expired and outdated licenses,**  
order new licenses and upgrades
-  **Manage your company licenses**  
and distribute them to end users

2) Purchase Product license(s)를 클릭!

▼ "저는 아래와 같은 화면이 안 나와요! 예전에 한번 받은 적이 있어서 그런가.."

→ 앗. 이미 개발 천재...! 아이디를 다시 만들어 진행해주세요!



3) Buynew license를 클릭!

Product	Price yearly	Price monthly	Action
All Products Pack	\$249.00 yearly	\$24.90 monthly	<a href="#">Buy new license</a>
AppCode	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
CLion	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
DataGrip	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
GoLand	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
IntelliJ IDEA Ultimate	\$149.00 yearly	\$14.90 monthly	<a href="#">Buy new license</a>
PhpStorm	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
PyCharm	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
ReSharper Ultimate	\$149.00 yearly	\$14.90 monthly	<a href="#">Buy new license</a>
ReSharper	\$129.00 yearly	\$12.90 monthly	<a href="#">Buy new license</a>
ReSharper C++	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
ReSharper Ultimate + Rider Pack	\$179.00 yearly	\$17.90 monthly	<a href="#">Buy new license</a>
Rider	\$139.00 yearly	\$13.90 monthly	<a href="#">Buy new license</a>
RubyMine	\$89.00 yearly	\$8.90 monthly	<a href="#">Buy new license</a>
WebStorm	\$59.00 yearly	\$5.90 monthly	<a href="#">Buy new license</a>

4) 체크박스를 눌러서 PROCEED AS NEW CUSTOMER를 클릭해주세요

## eStore Identification

Let's get started!

Please provide an email address where we should send your licenses or an existing Subscription Pack reference:

younghwang@kakao.com

We cannot seem to find any previous purchases associated with this address.

Please confirm your country/region:

Korea

I am eligible to use the email address provided above.

I have read and I accept the [JetBrains Account Agreement](#)

**PROCEED AS NEW CUSTOMER**

If you're an existing customer maybe try [providing a license ID or license key](#) to help us find you?

Copyright © 2000–2020 JetBrains s.r.o. | [Help](#) | [Support](#) | [JetBrains Privacy Policy](#) | [JetBrains Account Agreement](#) | Build #2020.06-2104

5) Have a discount code?를 클릭! (거의 다왔어요 두구두구..)



## eStore Order Checkout or [request quote](#)

Subscription:

Enable automatic renewal (Credit Card or PayPal only)  
You can disable automatic renewals any time in your [JetBrains Account](#).

Product	Order / PO Number	Item Price	Quantity	Extended Price
SKU: P.S.PC.Y <b>PyCharm</b> – Personal annual subscription Valid for 1 year since order delivery		US \$89.00 /year	1	US \$89.00
<input type="button" value="Have a discount code?"/>		Total		US \$89.00
		TAX (10%)		US \$8.90
		Grand Total		US \$97.90

Licensee information

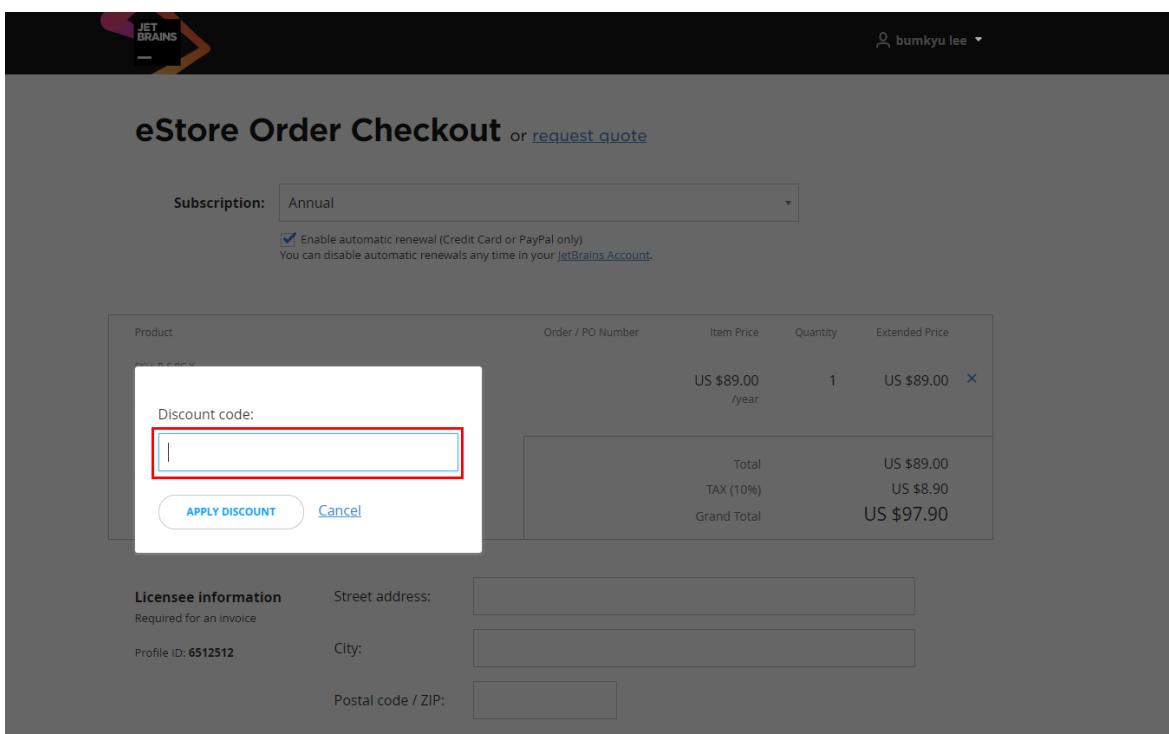
Required for an invoice

Profile ID: **6512512**

Street address:

City:

5) 튜터님께 각자 부여 받은 promotion code를 입력!



6) \$0로 되었으면 완료! Place Order를 클릭해주세요!

## eStore Free Checkout

Subscription: Annual

Product	Order / PO Number	Quantity	Extended Price			
<b>PyCharm</b> – Personal annual subscription		1	US \$0.00			
4 months training Discount code BA92Z-U7BYS-Q7V4U-E7MNR-PNWXG						
Subscription Term	Order / PO Number	Regular Price (per 1 year)	Subscription Period	Adjusted Price	Quantity	Extended price
SKU: P-S-PC-V Personal annual subscription Jun 27, 2020 — Oct 26, 2020		US \$0.00	122 days	US \$0.00	1	US \$0.00
			Total	US \$0.00		

**Licensee contact**  
Will receive license information  
Profile ID: 6512512

Email address: bumkyu.lee@gmail.com

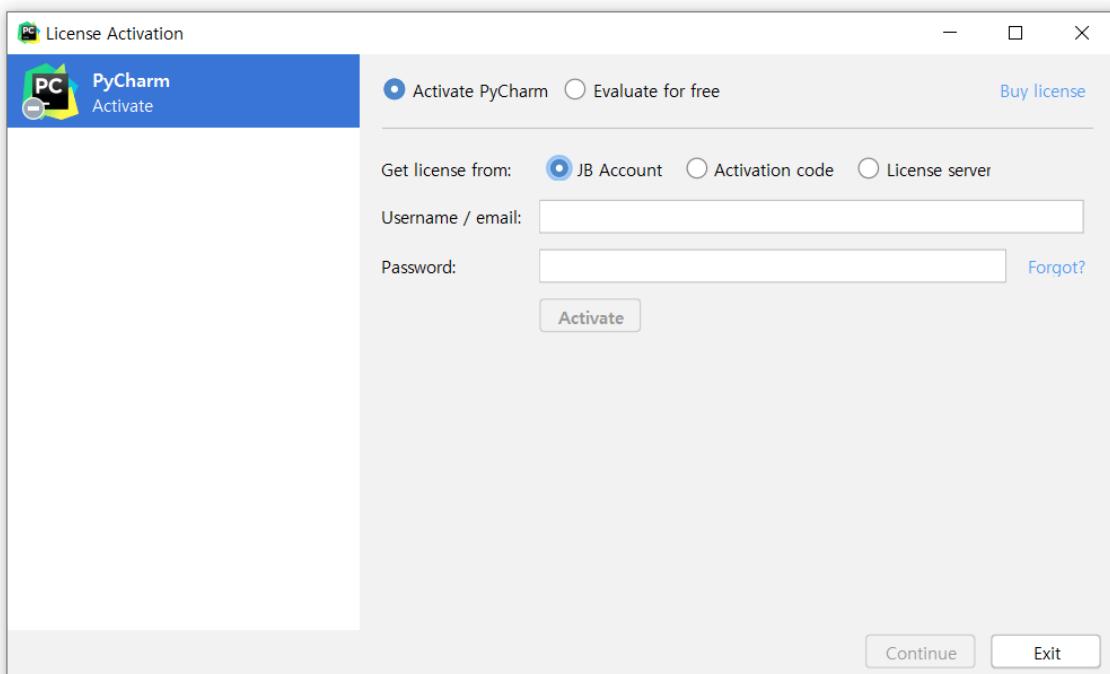
Phone (Optional):

By proceeding to purchase, you are contracting with JetBrains s.r.o.  
We will process your personal data to perform our contractual obligations, to improve our products, \* for promotional purposes, \* for our protection, \* and to fulfill our legal duties. For more details, please refer to [JetBrains Purchase Terms](#) and [JetBrains Privacy Policy](#).

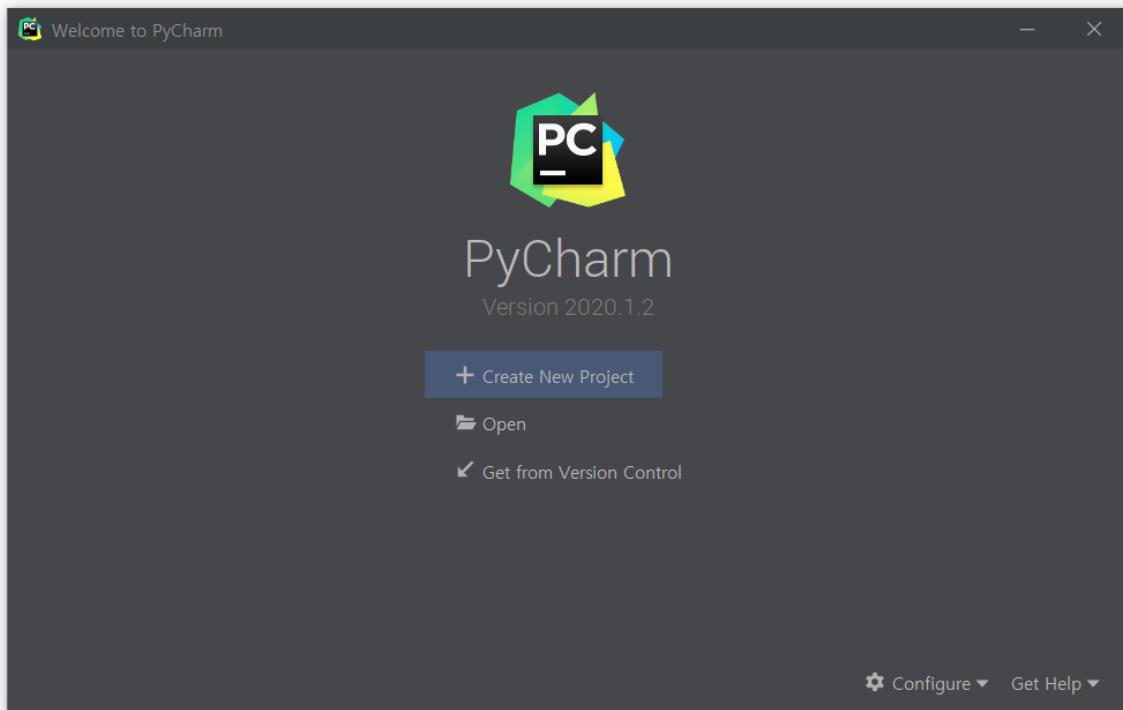
I have read and I agree to the [JetBrains Purchase Terms](#) and [Toolbox Subscription License Agreement](#)

**PLACE ORDER**

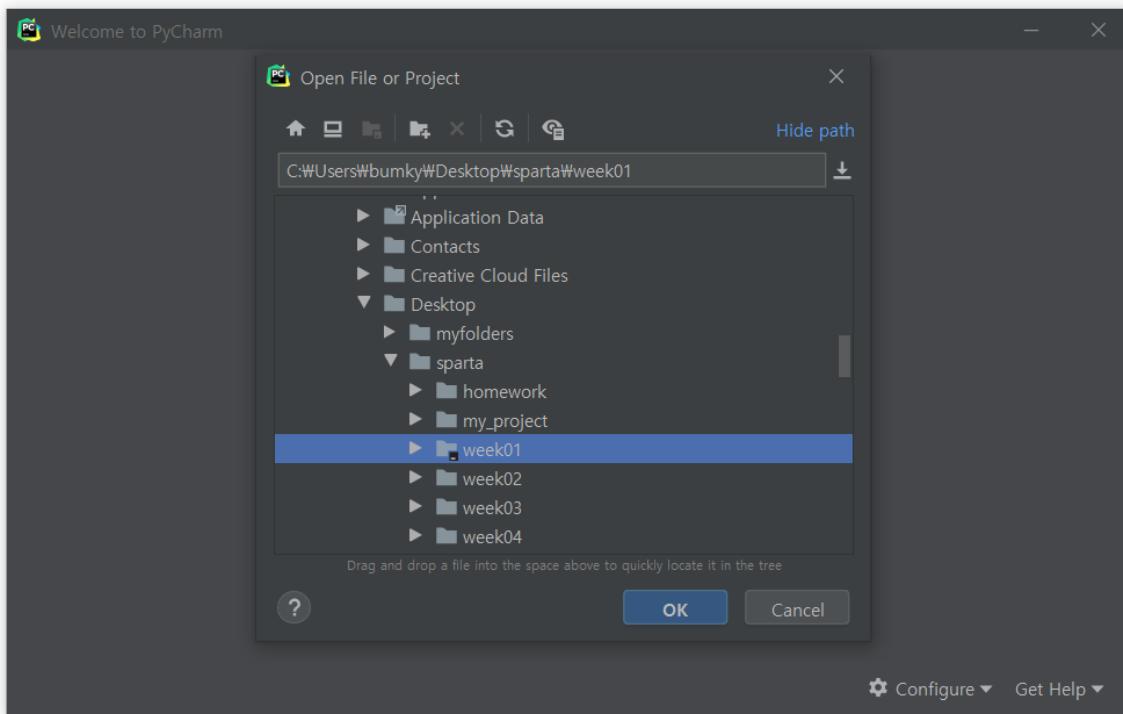
7) 자, 이제 Pycharm Pro를 실행하고, email / password를 입력하고 → Activate를 클릭!



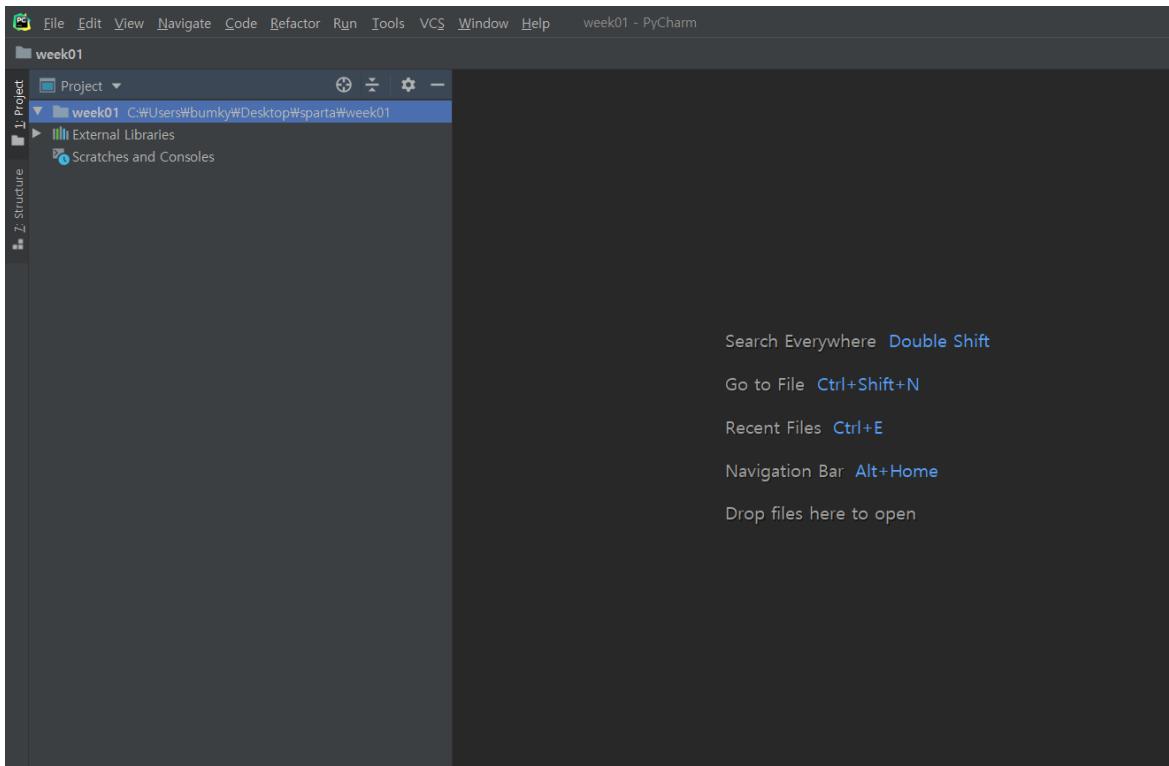
8) 10초만 기다리면 → 짠! 설치가 모두 완료되었습니다!



9) Open을 클릭해서, 바탕화면→sparta폴더→week01을 눌러주세요!



10) 야호! 1주차 준비 완료!



## ▼ 2) 웹 기초 동작원리 - 서버/클라이언트/웹의 동작 개념



앞으로 8주 동안 '웹 기초 동작원리'가 여러분들의 **여정 지도**가 될 거에요! 내가 왜 이걸 하고 있지? 헷갈릴 땐 우리가 어디 부분을 배우고 있는지 떠올려봅시다 🧠

### ▼ 1. 웹을 데이터 관점으로 바라보기

- IT라는 단어가 익숙하신가요? 바로 Information Technology 의 약자. 즉, 정보 기술입니다.
- 웹은 정보를 빠르게 교환하기 위해 태어났어요. 각 웹 페이지 하나 하나는 정보를 담은 문서(Document)입니다. 결국, 화면은 정보(데이터)를 표현한 것이지요.
- 데이터가 우리 눈에 보이도록 화면을 만드는 것이 프론트엔드(Front-end)의 역할이고,
- 데이터를 관리(전송, 저장, 수집, 가공)하는 것이 백엔드(Back-end)의 역할입니다.
- 이제 구체적으로 웹의 기초 동작 원리를 살펴볼까요?

### ▼ 2. [👉튜터와 같이] 네이버 웹페이지를 해킹해보자!



크롬 창에서 보이는 웹페이지는 어떤 원리로 보여지는 것일까요?

원리를 알기 위해서, 일단 해킹부터 해보죠! (응?)

'네이버뉴스'라고 쓰인 곳을 원하는 문구로 바꾸어보죠!

**NAVER**

메일 카페 블로그 지식IN 쇼핑 Pay ▶TV 사전 뉴스 증권 부동산 지도 영화 뮤직 책 웹툰 더보기 1 2020 3월 모의고사

대한체육회 100년, 국민과 함께 할 100년  
100년 후대에게 전달할 꿈과 희망의 손편지를 보내주세요  
공모 접수 : 2020. 4. 6(월) ~ 5. 8(금)  
결과 발표 : 20. 6. 1(월) \*홈페이지 공지 및 개별통보

대한체육회 100

연합뉴스 > 270만 저소득세대 현금→1천900만 세대 상품권·카드... 네이버뉴스 · 연예 스포츠 경제

뉴스스탠드 > 구독한 언론사 전체언론사

YTN	스포츠조선	The Korea Herald	Net Korea	중앙일보	OSEN
한국일보	전자신문	MBC	MBN	머니투데이	이데일리
석간 문화일보	한국경제TV	ChosunBiz	UPI뉴스	EBS	산

앗! 바뀌었다!

**NAVER**

메일 카페 블로그 지식IN 쇼핑 Pay ▶TV 사전 뉴스 증권 부동산 지도 영화 뮤직 책 웹툰 더보기 4 악쿠르트

대한체육회 100년, 국민과 함께 할 100년  
100년 후대에게 전달할 꿈과 희망의 손편지를 보내주세요  
공모 접수 : 2020. 4. 6(월) ~ 5. 8(금)  
결과 발표 : 20. 6. 1(월) \*홈페이지 공지 및 개별통보

대한체육회 100

연합뉴스 > 김종인 "통합당 비대위, 할 일 다했다고 생각하면 그..." 스파르타 뉴스! 연예 스포츠 경제

뉴스스탠드 > 구독한 언론사 전체언론사

프레시안	KBS WORLD	sportalkorea	NewDaily	머니투데이	KBS
이데일리	국민일보	SBS	ChosunBiz	JTBC	NEWSIS
조선일보	헤럴드경제	한국경제TV	arirang	Daily	SBS 연예스포츠

크롬을 새로고침하면 어? 바뀐 부분이 사라졌네요. 왜 그럴까요?

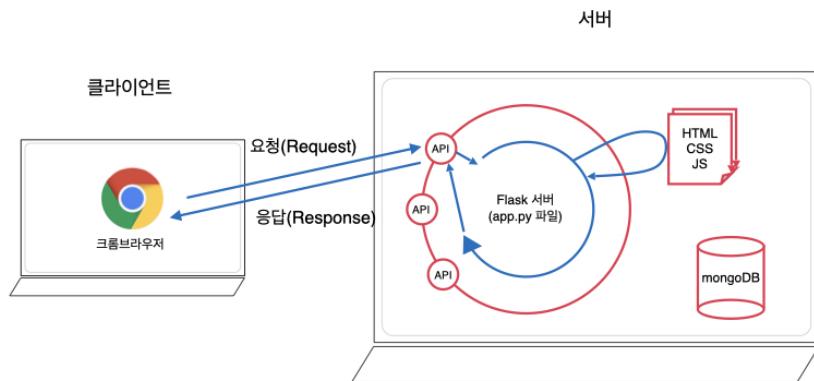
### ▼ 3. 웹 기초 동작원리 (HTML을 받는 경우)

네! 우리가 보는 웹페이지는 모두 서버에서 미리 준비해두었던 것을 "받아서", "그려주는" 것입니다. 즉, 브라우저가 하는 일은 1) 요청을 보내고, 2) 받은 화면 파일(HTML 파일)을 그려주는 일 뿐이죠.

 근데, 1)은 어디에 요청을 보내나요? 좋은 질문입니다. 서버가 만들어 놓은 "API"에 미리 정해진 약속대로 요청을 보내는 것이랍니다.

예) <https://naver.com/>

→ 이것은 "naver.com"이라는 이름의 서버에 있는, "/" 창구에 요청을 보낸 것!



- 웹은 **HTTP(HyperText Transfer Protocol)**라는 통신 규약(**Protocol**, 서로 지키도록 협의한 **규칙**)을 따른답니다. url에서 `http://` 가 바로 HTTP라는 통신 규약을 따른다는 표시에요.
- HTTP에서 **요청(Request)**하는 쪽을 **클라이언트**, 요청을 처리해 그에 맞게 **응답(Response)**하는 쪽을 **서버**라고 합니다.

▼ 4. 웹 기초 동작원리 (데이터만 받는 경우)

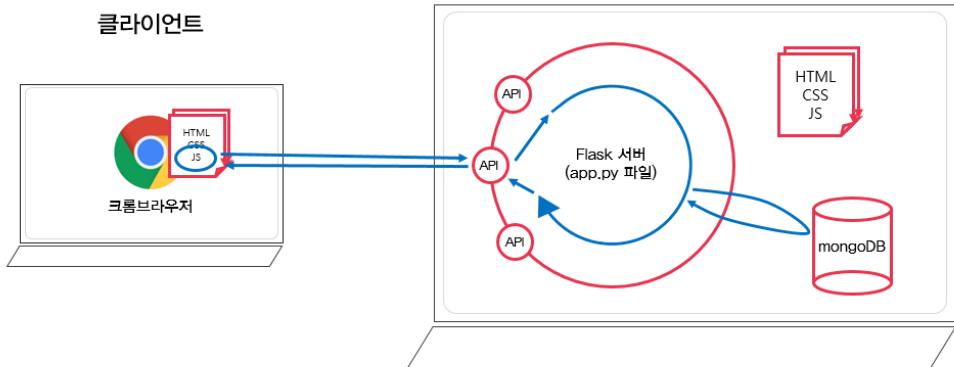
 앗, 그럼 항상 서버는 요청에 응답해서 이렇게 화면 파일(HTML 파일)만 보내주나요?  
아뇨! 데이터만 보내줄 때가 더~ 많아요.

사실 화면 파일(HTML 파일)도 줄글로 쓰면 이게 다 '데이터' 아닌가요?

 자, 콘서트 티켓을 예매하고 있는 상황을 상상해봅시다!  
좌석이 찰 때마다 예매 페이지가 새로고침 되면 난감하겠죠ㅠ?

이럴 때! 서버에서 데이터만 받아서 데이터 내용만 바꿔주게 된답니다.

## 서버



👉 데이터만 보내줄 경우는, 이렇게 생겼어요!  
(소곤소곤) 이렇게 데이터를 표현하는 것을 **JSON** 형식이라고 한답니다.

A screenshot of a web browser window. The address bar shows 'openapi.seoul.go.kr:8088/6d4d7...'. The page content displays a JSON object:

```
{  
  - RealtimeCityAir: {  
      list_total_count: 25,  
      - RESULT: {  
          CODE: "INFO-000",  
          MESSAGE: "정상 처리되었습니다"  
      },  
      - row: [  
          - {  
              MSRDT: "202004241900",  
              MSRRGN_NM: "도심권",  
              MSRSTE_NM: "종구",  
              PM10: 44,  
              PM25: 20,  
              O3: 0.039,  
              NO2: 0.02,  
              CO: 0.4,  
              SO2: 0.003,  
              INDEX_NM: "보통",  
              INDEX_MVL: 59,  
              ARPLT_MAIN: "PM10"  
          },  
          - {  
              MSRDT: "202004241900",  
              MSRRGN_NM: "도심권",  
              MSRSTE_NM: "종구",  
              PM10: 44,  
              PM25: 20,  
              O3: 0.039,  
              NO2: 0.02,  
              CO: 0.4,  
              SO2: 0.003,  
              INDEX_NM: "보통",  
              INDEX_MVL: 59,  
              ARPLT_MAIN: "PM10"  
          }  
      ]  
  }  
}
```

### ▼ 5. 웹 기초 동작원리로 보는 1~5주차 학습 순서!

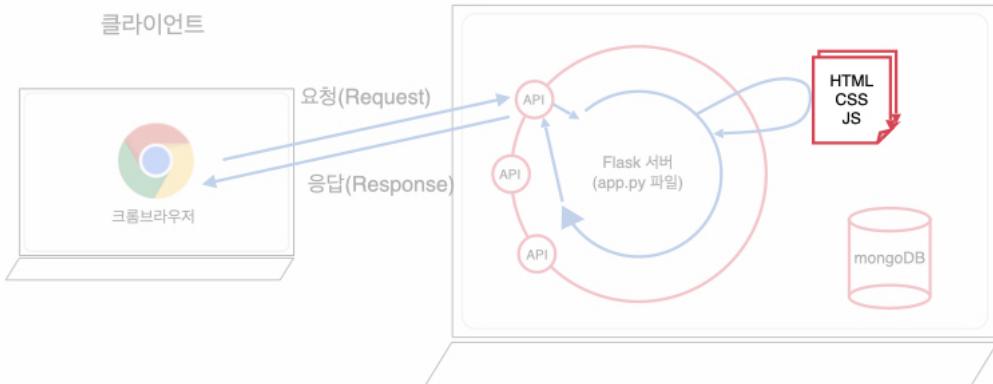
#### ▼ 1주차: HTML, CSS, JavaScript(기초)



HTML과 CSS를 배우는 날! 즉, 4주차에 클라이언트에 보내줄 HTML파일을 미리 만들어 두는 과정입니다. +  
또, 2주차에 JavaScript(줄여서 JS)를 능숙하게 다루기 위해서, 문법을 먼저 조금 배워둘게요!

서버

클라이언트



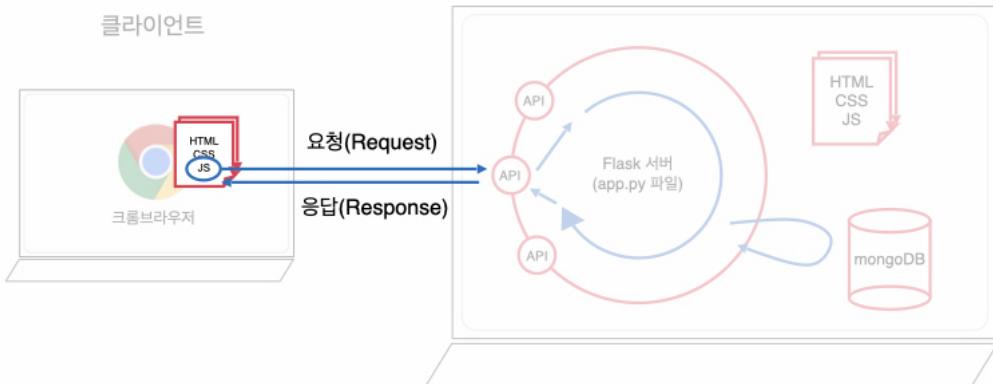
▼ 2주차: JS(응용), jQuery, Ajax



서버가 이미 HTML파일을 보내주었다고 가정하고, JS로 서버에 데이터를 요청하고, 응답받은 데이터를 처리하는 방법을 배워볼거예요

서버

클라이언트



▼ 3주차: Python, 웹 스크래핑(크롤링), mongoDB



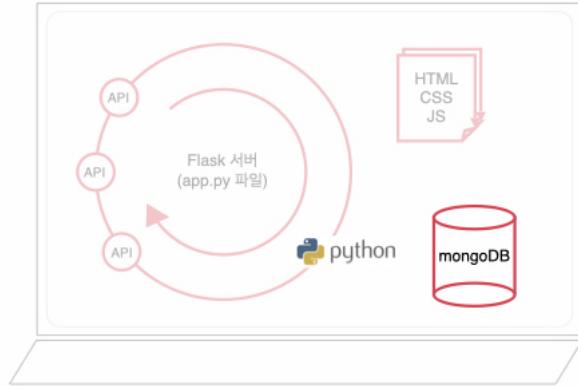
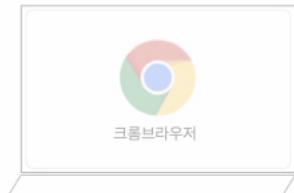
드디어 'Python'을 배울거예요. 먼저 문법을 연습하고, 라이브러리를 활용하여 네이버 영화목록을 짹 가져와보겠습니다. (기대되죠!)

+

그리고, 우리의 인생 첫 데이터베이스. mongoDB를 다뤄볼게요!

서버

클라이언트



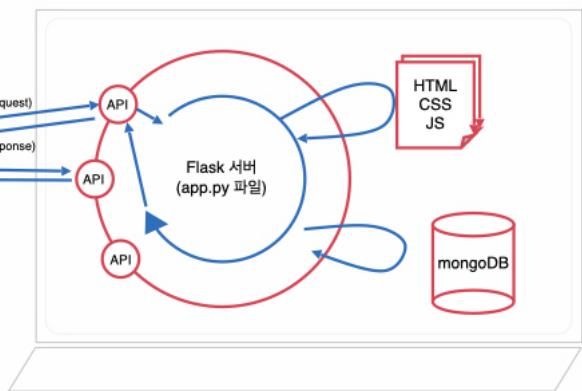
▼ 4주차: Flask, 미니프로젝트1 - 모두의책리뷰



서버(백엔드)를 만들어봅니다! HTML과 mongoDB까지 연동하여,  
미니프로젝트1을 완성해보죠! 굉장히 재미있을 거예요!

서버

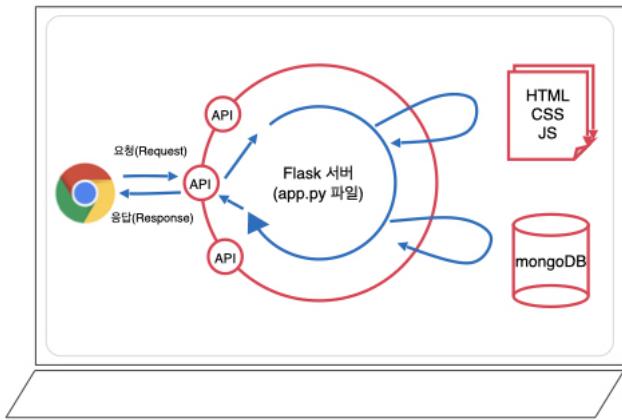
클라이언트



나중에 또 이야기하겠지만 헷갈리면 안되는 것!

우리는 컴퓨터가 한 대잖아요... 그래서 같은 컴퓨터에다 서버도 만들고, 요청도 할 거예요. 즉, 클라이언트 = 서버가 되는 것이죠.

이것을 바로 "로컬 개발환경"이라고 한답니다! 그림으로 보면, 대략 이렇습니다.

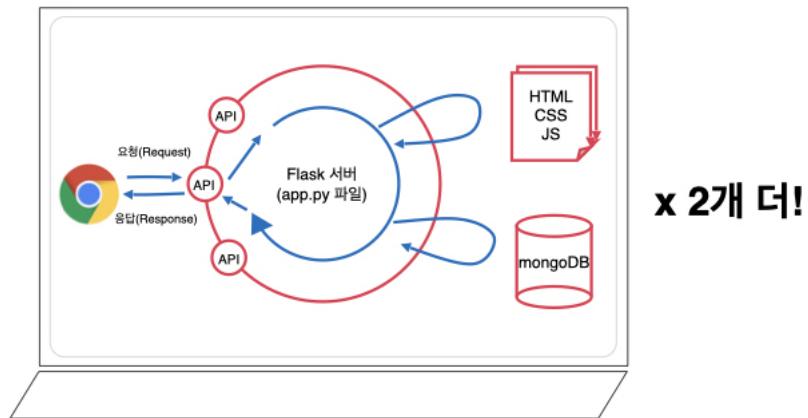


▼ 5주차: 미니프로젝트2-나홀로 메모장, 미니프로젝트3 - 페이보릿무비스타



아직 익숙해지지 않았을 여러분을 위해! 같은 난이도의 유사한 두 개의 프로젝트를 더 진행하여 머릿속의 퍼즐을 맞출 예정입니다.

여기까지 배웠다면, 이제 6~8주차 프로젝트 준비 완료!!



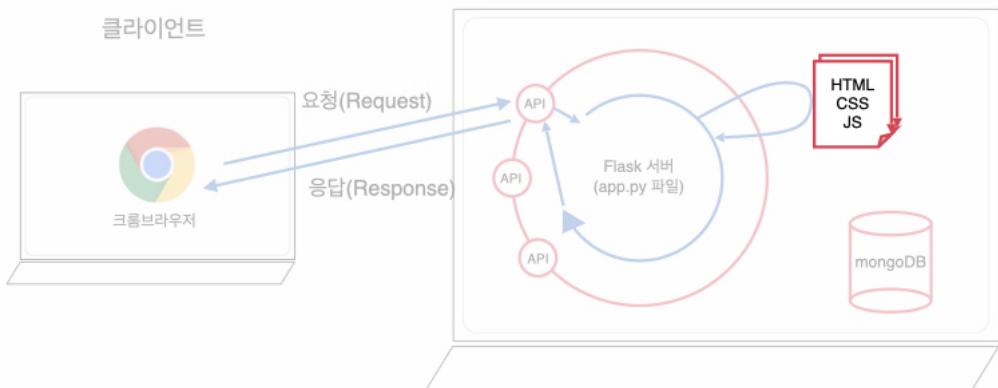
웹 기초 동작 원리 - 1주차 : HTML, CSS, Javascript



HTML과 CSS를 배우는 날!

즉, 4주차에 클라이언트에 응답해서 보내줄 HTML파일을 미리 만들어 두는 과정입니다.

또, 2주차에 JavaScript(줄여서 JS)를 능숙하게 다루기 위해서, 문법을 조금 배워둘게요!



오늘 최종적으로 완성해볼 것! 나홀로 메모장([링크](#))  
튜터님과 어떤 화면을 만들어볼지 먼저 보고 시작하죠!

## [2시간] 화면 만들기 - HTML, CSS 기초

### ▼ 4) HTML과 CSS의 개념

### ▼ 5) HTML 기초

- HTML 문서는 크게 head와 body로 구성되어 있습니다. head 안에는 웹페이지의 속성 정보를, body안에는 웹페이지의 내용을 담습니다.
- HTML 요소(element) 생김새
  - 웹을 데이터 관점으로 바라보겠다는 말 기억하시나요? 화면에 보여지는 각 요소(element)들은 모두 각각의 정보입니다! 요소는 태그(tag)로 표현합니다. 대부분의 요소들은 여는 태그(`<태그이름>`)와 닫는 태그(`</태그이름>`)는 쌍으로 이루어져 있습니다.

```
<태그이름 세부속성="속성값"> 내용 </태그이름>
```

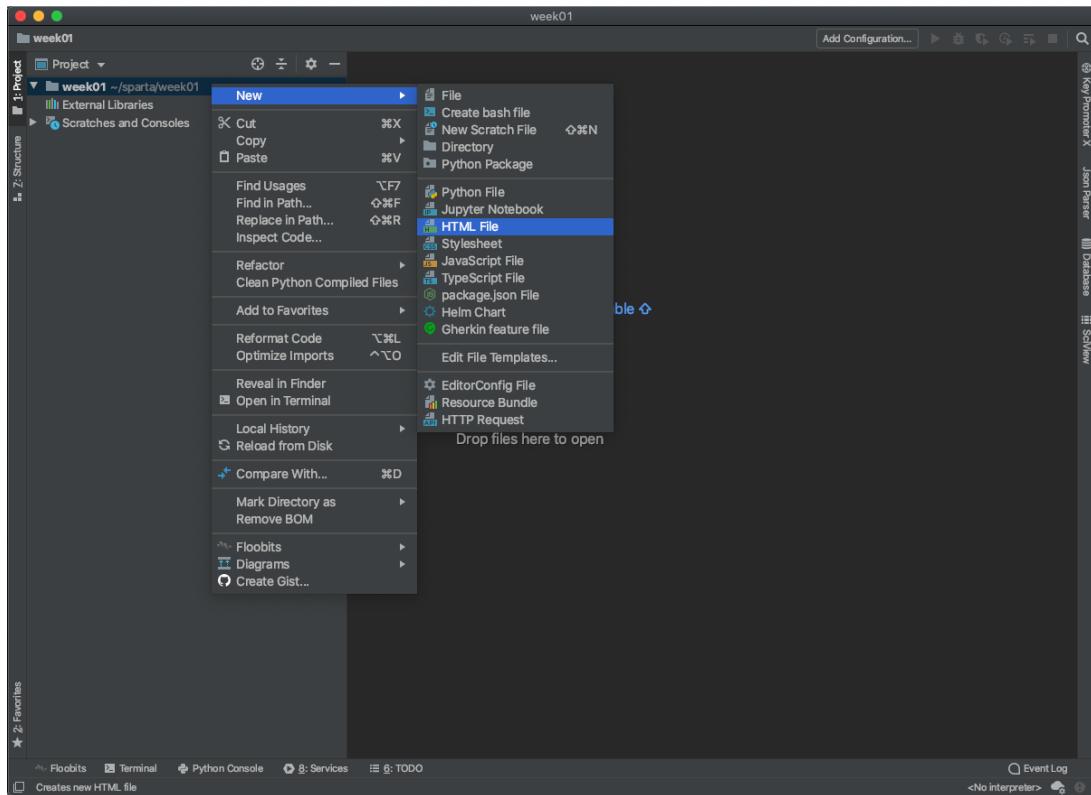
- 【💡튜터와 같이】 이제 우리 첫 코드를 같이 만들어볼게요!



### [Pycharm에서 HTML 파일 만들기]

- 파일을 생성할 폴더(week01)를 마우스 클릭하고 -
- 우클릭 - New - HTML File 클릭하고 이름을 입력하면 아래처럼 파일이 자동 생성됩니다.
- Pycharm은 파일이 자동 저장이 된답니다. 저장버튼을 누르지 않아도 됩니다.

### ▼ Pycharm 화면



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>

</body>
</html>
```

- head 안에 들어가는 대표적인 요소들: meta, script, title 등



페이지를 설명하는 정보(meta)와, CSS 링크(script) 등을 넣습니다. 즉, 화면에 보이지 않지만 웹 페이지에 필요한 것들을 담습니다. 아래 실습을 하면서 필요한 정보들을 넣어보겠습니다.

- body 안에 들어가는 대표적인 요소들

- 아래 코드를 작성한 후, 크롬 브라우저에 화면을 띄우겠습니다. Pycharm에서 코드 화면에 커서를 두고 shift를 누른채 브라우저 아이콘을 클릭하면 됩니다.

#### ▼ Pycharm 화면

```

<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | HTML 기초</title>
  </head>
  <body>
    <!-- 구역을 나누는 태그들 -->
    <div>나는 구역을 나누죠.</div>
    <p>나는 문단이에요</p>
    <ul>
      <li>bullet point!1</li>
      <li>bullet point!2</li>
    </ul>
    <!-- 구역 내 콘텐츠 태그들 -->
    <h1>h1은 제목을 나타내는 태그입니다. 페이지마다 하나씩 꼭 써주는 게 좋아요. 그래야 구글 검색이 잘 되거든요.</h1>
    <h2>h2는 소제목입니다.</h2>
    <h3>h3~h6도 각자의 역할이 있죠. 비중은 작지만..</h3>
    <hr>
    <span>태그입니다: 특정 <span style="color:red">글자</span>를 꾸밀 때 써요
    <hr>
    <a>태그입니다: <a href="http://naver.com/"> 하이퍼링크 </a>
    <hr>
    <img>태그입니다: 
    <hr>
    <input>태그입니다: <input type="text"/>
    <hr>
    <button>버튼입니다</button>
    <hr>
    <textarea>태그입니다: <textarea>나는 무엇일까요?</textarea>
  </body>
</html>

```

- html을 chrome으로 확인하기 단축키 : alt(또는 option)+f2 → shift 누르고 크롬 클릭

## ▼ 코드

```

<!DOCTYPE html>
<html lang="ko">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | HTML 기초</title>
  </head>

  <body>
    <!-- 구역을 나누는 태그들 -->
    <div>나는 영역을 나누죠. 속성을 주어야 눈에 보이는 모양이 달라져요.</div>
    <p>나는 문단이에요</p>
    <ul>

      <li>bullet point!1</li>
      <li>bullet point!2</li>
    </ul>

    <!-- 구역 내 콘텐츠 태그들 -->
    <h1>h1은 제목을 나타내는 태그입니다. 페이지마다 하나씩 꼭 써주는 게 좋아요. 그래야 구글 검색이 잘 되거든요.</h1>
    <h2>h2는 소제목입니다. 숫자가 커지면 제목 크기가 작아집니다. </h2>
    <h3>h3~h6도 각자의 역할이 있죠. 비중은 작지만..</h3>
    <hr>
    <span>태그입니다: 특정 <span style="color:red">글자</span>를 꾸밀 때 써요
    <hr>
    <a>태그입니다: <a href="http://naver.com/"> 하이퍼링크 </a>
    <hr>
    <img(이미지)>태그입니다: 
    <hr>
    <input>태그입니다: <input type="text"/>
    <hr>
    <button>버튼입니다</button>
    <hr>
    <textarea>태그입니다: <textarea>나는 무엇일까요?</textarea>
  </body>

</html>

```



이 외에도 아주 많으며, 개발자들도 모두 외우고 있지 않습니다. 필요할 때마다 구글에서 검색해서 넣어보세요!

▼ 6) 📝 간단한 로그인 페이지 만들기



앞으로 실습이 필요한 부분이 나오면 "📝" 모양으로 알려줄게요!

퀴즈설명을 먼저 보고 → 정해진 시간동안 혼자 한 다음 → 함께하기를 보세요!

▼ Q. 퀴즈 설명

▼ 모습 보기

## 로그인 페이지

ID:

PW:

[로그인하기](#)



힌트: HTML기초에서 봤던 코드들을 적절히 조합해보세요!  
ctrl+c,v 신공도 괜찮아요!

▼ A. 함께 하기

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 로그인페이지</title>
</head>

<body>
    <h1>로그인 페이지</h1>
    <div>
        <p>
            ID: <input type="text" />
        </p>
        <p>
            PW: <input type="password" />
        </p>
    </div>
    <button>로그인하기</button>
</body>
```

```
</html>
```

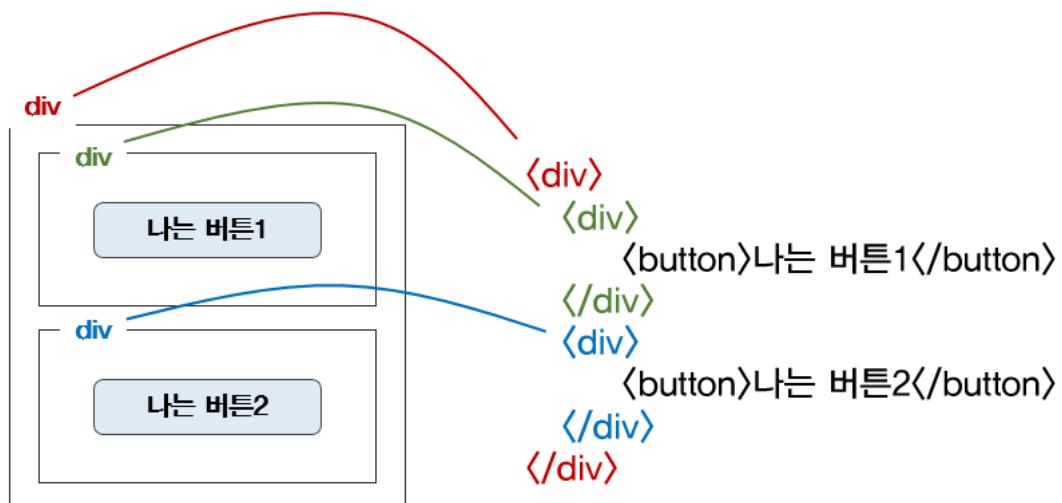
#### ▼ 7) HTML 부모-자식 구조 살펴보기



HTML 요소는, "어떤 요소 안에 어떤 요소가 있느냐"를 이해하는 것이 가장 중요합니다(이런 구조를 가계도 같다고 해서 부모-자식 구조, 뿌리-줄기- 잎 같다고 해서 tree 구조라고 합니다).

나를 감싸고 있는 태그가 바뀌면, 그 안의 내용물도 모두 영향을 받지요! 앞으로 배울 CSS 선택자에서

- 상위에 하위 요소를 감싸고 있는 요소를 부모(parent), 하위에 있는 요소를 자식(children) 이라고 합니다. 여기서는 빨간색 div가 부모, 초록색 div, 파란색 div 가 자식입니다.
- 빨간색 div 안에, 초록색/파란색 div가 들어있습니다. 아래와 같은 상황에서 빨간색 div를 가운데로 옮기면, 내용물인 초록/파란 div도 모두 함께 이동하겠죠! 박스를 옮기면 안의 내용물도 함께 옮겨지는 것과 같은 원리입니다.
- 같은 원리로, 빨간색 div의 글씨색을 바꾸면, 초록/파란 div안의 글씨색도 바뀐답니다.



#### ▼ 8) CSS 기초

- CSS는 Cascading Style Sheet 의 약자입니다. HTML 부모-자식 구조를 배웠었죠? CSS는 폭포(cascading)가 위에서 아래로 물이 흐르는 것처럼, 위에 있는 부모요소에 디자인을 적용하면 자동으로 안쪽 자식 요소까지 디자인 적용이 됩니다.
- 선택자(selector)의 개념: class 값으로 태그를 선택(select)해서 꾸미기



무엇을 꾸민다는 것은, 일단 '선택(select)해야' 가능한 일입니다. 그래서 CSS에선 class값으로 '이름표'를 달아줍니다. 예를 들어 mytitle이라고 이름표를 달아준 뒤, "mytitle 이름표가 붙은 부분들의 글씨를 빨간색으로 바꿔라"라고 하는 것이죠.



##### 잠깐! 선택자(selector) 이름 잘 짓기(naming)

선택자 이름은 자유롭게 지을 수 있습니다. 앞서 배운 파일명 이름짓기처럼 어떤 역할을 하기 위한 선택자인지 잘 알 수 있게 이름을 정하는 것이 중요합니다. 예를 들어, 화면에서 기사 제목을 꾸미기 위한 class를 만들었다면, 이름을 'news-title'로 하면 좋겠죠?



또는, 이름표를 달지 않고, 아래 코드의 body 처럼 태그명을 지칭해서 쓸 수도 있습니다. **body**는 태그명을, **red-font**은 class 명을 만들어 가리켰네요.

- CSS는 어떻게 사용하나요?



<head> ~ </head> 안에 <style> ~ </style>로 공간을 만들어 작성합니다.  
아래 코드를 통해 간단한 사용 방법을 알아봅니다.

red-font라는 클래스를 가리킬 때, .red-font { ... } 라고 써줘야 하는 것을 꼭! 기억하세요! 이름 앞에 . 을 빼먹지 마세요!

- [💡튜터와 같이] 코드를 쓰면서 이해해볼까요? 일단 따라 써 봅니다!

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 로그인페이지</title>
    <style>
        /* class를 만들어 가리킬 때 */
        .red-font {
            color: red;
        }
    </style>
</head>

<body>
    <h1 class="red-font">로그인 페이지</h1>
    <div class="red-font">
        <p>
            ID: <input type="text" />
        </p>
        <p>
            PW: <input type="password" />
        </p>
    </div>
    <button>로그인하기</button>
</body>
</html>
```

- 더 다양한 선택자들을 살펴보기



선택자는 CSS의 꽃!

가독성 있는 코드를 위해 협업에선 규칙/상황에 따라 아주 다양한 선택자를 사용하지만, 여기서는 꼭 알고 있어야 할 몇 개만 배워봅니다!

#### ▼ 다양한 선택자들을 살펴볼까요?

1. 전체 선택자 (HTML 페이지 내부의 모든 태그를 선택합니다)

```
/* CSS */
* {
    margin: 0;
    text-decoration: none;
}
```

#### 2. 태그 선택자

```
/* CSS */
/* 모든 P 태그를 지칭 */
p {
    background: yellowgreen;
    color: darkgreen;
}

<!-- HTML -->
<p>태그 선택자(Type Selector)</p>
```

### 3. 클래스 선택자 & 자식선택자

```
/* CSS */
/* class1 클래스를 가진 모든 태그를 지칭 */
/* 케이스1에 적용됩니다. */
.class1 {
    background: yellowgreen;
    color: darkgreen;
}

/* div 태그 중 class2 클래스를 가진 태그만 지칭 */
/* 케이스3에만 적용됩니다. 케이스2는 p 태그이므로, 적용되지 않습니다.*/
div.class2 {
    background: darkgreen;
    color: yellowgreen;
}

.class3 > h1 {
    color: red;
}

<!-- HTML -->
<p class="class1">케이스1</p>
<p class="class2">케이스2</p>
<div class="class2">케이스3</div>
<div class="class3"><h1>케이스4</h1></div>
```

- class는 중첩이 가능합니다.



"1학년" 이름표와 "개발자" 이름표를 동시에 가질 수 있는 것처럼요.

- [👉튜터와 같이] 아래 코드를 따라해 볼까요? 여기선 yellow-font과 small-font가 중첩됐네요! 노란색이면서, 작은 글씨가 나오게 되는 것이죠.

```
<!DOCTYPE html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 로그인페이지</title>
    <style>
        .red-font {
            color: yellow;
        }

        .small-font {
            font-size: 10px;
        }
    </style>
</head>

<body>
    <h1 class="red-font">로그인 페이지</h1>
    <div class="red-font small-font">
        <p>
            ID: <input type="text" />
        </p>
        <p>
            PW: <input type="password" />
        </p>
    </div>
    <button>로그인하기</button>
</body>

</html>
```

- 자주 쓰이는 css들은 다음과 같은 것들이 있어요!

참고: <http://www.walterz.net/2017/08/18/자주-사용되는-css-속성/>

 배경관련  
background-color  
background-image  
background-size

사이즈  
width  
height

폰트  
font-size  
font-weight  
font-family  
color

간격  
margin  
padding

#### ▼ 9) 자주 쓰이는 CSS 연습하기

 [연습할 것들]  
h1, h5, background-image, background-size, background-position  
color, width, height, border-radius, margin, padding

#### ▼ (1) 튜터와 같이] 아래와 같은 화면을 만들어볼까요? → 튜터님이 하나씩 적용하며 설명해주실거에요!



margin과 padding (→ 헷갈리지 말기!)

- margin은 요소 외부 여백(요소와 화면과의 여백)을, padding은 요소 내부의 여백을 의미합니다.
- 헷갈릴땐, '내가 패딩을 입었으니까 요소 내부 여백이구나'를 떠올려보세요.
- div에 색깔을 넣고, 직접 사용해서 차이를 비교해보세요!

- 참고: 이미지 주소: <https://www.ancient-origins.net/sites/default/files/field/image/Agesilaus-II-cover.jpg>

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 로그인페이지</title>
    <style>
        .login-title {
            color: white;
            width: 300px;
            height: 200px;
```

```

background-image: url('https://www.ancient-origins.net/sites/default/files/field/image/Agesilaus-II-cover.j
background-position: center;
background-size: cover;

border-radius: 10px;
text-align: center;
padding-top: 40px;
}

```

</style>

</head>

<body>

<div class="login-title">

<h1>로그인 페이지</h1>

<h5>아이디, 비밀번호를 입력해주세요</h5>

</div>

<div>

<p>

ID: <input type="text" />

</p>

<p>

PW: <input type="password" />

</p>

</div>

<button>로그인하기</button>

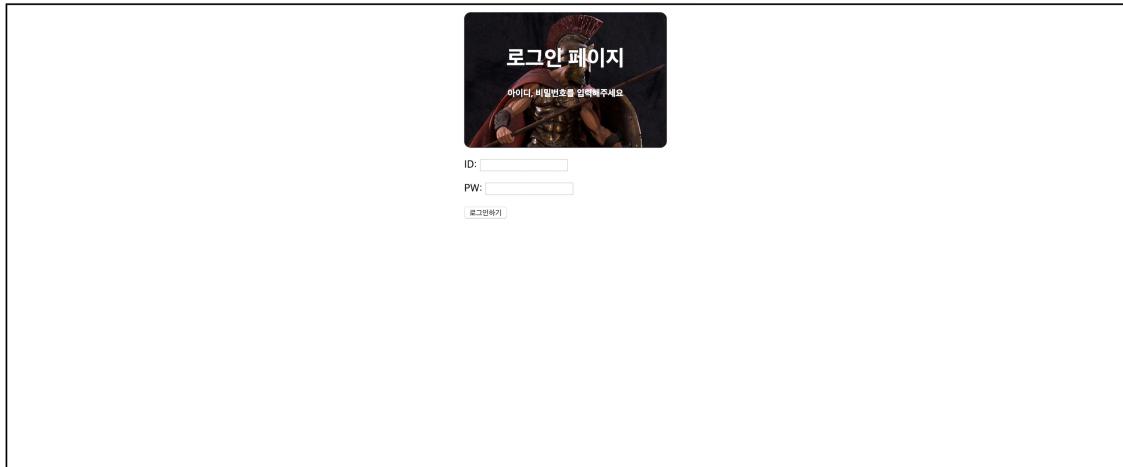
</body>

</html>

▼ (2) [💡튜터와 같이] 만들어둔 로그인 화면을 가운데로 가져와볼까요? → 튜터님이 하나씩 적용하며 설명해주실거에요!



width를 주고, margin: auto를 사용하자!



```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 로그인페이지</title>
    <style>
        .login-title {
            color: white;
            width: 300px;
            height: 200px;
            background-image: url('https://www.ancient-origins.net/sites/default/files/field/image/Agesilaus-II-cover.j
            background-position: center;
            background-size: cover;

            border-radius: 10px;
            text-align: center;
            padding-top: 40px;
        }

        .wrap {
            margin: 10px auto;
            width: 300px;
        }
    
```

```

        }
    </style>
</head>

<body>
    <div class="wrap">
        <div class="login-title">
            <h1>로그인 페이지</h1>
            <h5>아이디, 비밀번호를 입력해주세요</h5>
        </div>
        <div>
            <p>
                ID: <input type="text" />
            </p>
            <p>
                PW: <input type="password" />
            </p>
        </div>
        <button>로그인하기</button>
    </div>
</body>
</body>

</html>

```

#### ▼ 10) [扈튜터와 같이] 구글 웹폰트 입하기

- 구글 웹폰트: <https://fonts.googleapis.com/?subset=korean>

- (1) 마음에 드는 폰트를 클릭합니다.
- (2) [+ Select this style] 을 클릭합니다.

Google Fonts

Return to classic site   Browse fonts   Featured   Articles   About

**Nanum Gothic**

Designed by Sandoll

Select styles   Glyphs   About   License   **Pairings**

**Styles**

Type here to preview text  
Almost before we knew it, we had left the ground.

Size 30 px

Regular 400   Almost before we knew it, we had left the ground.   + Select this style

Bold 700   Almost before we knew it, we had left the ground.   + Select this style

Extra-bold 800   Almost before we knew it, we had left the ground.   + Select this style

- (3) 우측 상단의 모음 아이콘을 클릭합니다.

Return to classic site   Browse fonts   Featured   Articles   About

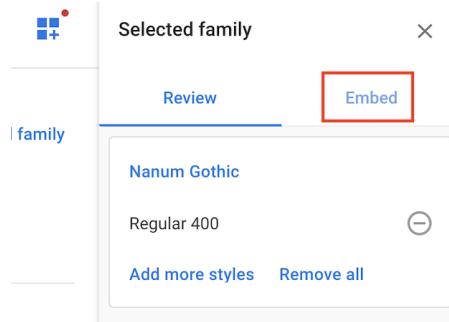
**airings**

Font styles

Size 30 px

+ Download family

- (4) Embed 탭을 클릭합니다.



(5) link 태그를 복사해서 <head> ~ </head> 사이에, CSS를 복사해서 <style> ~ </style> 사이에 넣습니다.

To embed a font, copy the code into the <head> of your html

<link> @import

```
<link href="https://fonts.googleapis.com/css2?family=Nanum+Gothic&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Nanum Gothic', sans-serif;
```

- Nanum Gothic이라는 폰트를 예제로 추가해보면, 아래와 같은 코드를 복사하면 됩니다.

```
<!-- HTML에 이 부분을 추가하고 -->
<link href="https://fonts.googleapis.com/css2?family=Nanum+Gothic&display=swap" rel="stylesheet">
```

```
/* CSS에 이 부분을 추가하면 완성! */
* {
  font-family: 'Nanum Gothic', sans-serif;
}
```

#### ▼ 11) CSS 파일 분리



<style> ~ </style> 부분이 너무 길어지면, 보기 어렵겠죠? 이럴 땐 아래와 같이 파일을 분리해둘 수 있습니다.

지금은 css가 간단하고, 한 파일에서 보는 게 편하기 때문에 적용은 하지 않을게요~

```
<!-- style.css 파일을 HTML 파일과 같은 폴더에 만들고, head 태그에서 불러오기 -->
<link rel="stylesheet" type="text/css" href = "(css파일이름).css">
```

## 😎 문제뱅크 - HTML, CSS



조금 더 실습하고 싶은 사람을 위한 '문제뱅크' 입니다. 문제 난이도가 표시되어 있어요.(★- 쉬움 ~ ★★★★★★ - 매우 어려움. 좌절하지 마세요)

▼ 게임하면서 CSS 연습하기

▼ 선택자(selector)를 연습합니다 (난이도 : ★)

### CSS Diner

You're about to learn CSS Selectors! Selectors are how you pick which element to apply styles to. Exhibit 1 - A CSS Rule p { margin-bottom: 12px; } Here, the "p" is the selector (selects all elements) and applies the margin-bottom style.

🔗 <https://flukeout.github.io/>



▼ CSS 컬러링 북 - CSS파일의 속성값을 수정해 색깔을 칠해보세요.(난이도 : ★)

### Hello Spring - CSS colouring page

🔗 <https://codepen.io/Lubna/pen/dyoqzwP>



### Spooky Graveyard - Colour Page

🔗 <https://codepen.io/Lubna/pen/zYYzOxN>



### Colouring with code

Do you like colouring books? Are you getting a little bored and running out of things to do at home? Well, I have created some digital colouring pages that you can colour with code. There is a lot going on in the world right now so I wanted to share something lighthearted and fun.

🔗 <https://lubna.dev/articles/colouring-with-code/>



▼ CSS 속성인 flexbox 를 게임으로 배워봅니다 (난이도 : ★★★)

### Flexbox Froggy

A game for learning CSS flexbox

🐸 <https://flexboxfroggy.com/#ko>



후반 3시간

[월수/화목반] : 체크인 & 출석체크



튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!

### 스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

▼ "15초 체크인"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)

- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.

## 배우고 적용하기 - 후반부

### 5분 꿀팁 - 코드 가꾸기

#### ▼ 코드에 주석 달기

- ! 주석은 언제 사용하나요?
  - 1) 필요없어진 코드를 삭제하는 대신 임시로 작동하지 못하게 하고 싶을 때
  - 2) 코드에 대한 간단한 설명을 붙여두고 싶을 때 사용합니다.
- 주석을 붙여놓으면, 브라우저/컴퓨터가 읽지 않아요. 즉, 개발자 본인 또는 동료를 위해 붙여두는 것!
- Pycharm에서 적용하기 : 주석처리하고 싶은 라인들을 선택(블록처리) → **ctrl(또는 command) + / (슬래시)**

#### ▼ 코드 정렬하기

- ! 정렬이 왜 중요한가요?
  - 코드 정렬이 제대로 되어있지 않으면, 코드의 생김새가 한 눈에 들어오지 않아요. 오류(버그)를 해결하기가 무척 어려워집니다.
  - Pycharm에서 적용하기: 정렬하고 싶은 파일에 커서를 두고 → **control(또는 command) + alt(또는 option) + L**

## [1시간] 화면 디자인 - 부트스트랩(Bootstrap)

#### ▼ 12) 부트스트랩(Bootstrap)이란?

- 참고자료: ([부트스트랩 기본 페이지](#)).
- 웹사이트 화면을 쉽게 만들 수 있도록 도와주는 일종의 디자인 키트입니다. 미리 만들어둔 예쁜 디자인을 가져다가 사용할 수 있어요.
  - 정확히는 프론트엔드 프레임워크(Front-end Framework)라고 부릅니다. (프레임워크가 무엇인지는 나중에 배울테니 넘어가셔도 좋습니다).
  - CSS를 다룰 줄 아는 것과, 미적 감각을 발휘하여 예쁘게 만드는 것은 다르다는 말 기억나시죠? 개발자들도 미리 완성된 부트스트랩을 가져다 쓰는 경우가 많습니다.

#### ▼ 13) 부트스트랩 - 시작 템플릿

- 아래 파일을 복사해서 새로운 HTML 파일을 하나 만들고, 시작해봅시다.



다른 사람이 미리 작성한 CSS를 내 HTML 파일에 적용한다는 점에서, 부트스트랩 적용은 CSS 파일 분리와 원리가 동일합니다. 다만, CSS의 파일을 인터넷 어딘가에서 다운로드받는다는 점이 다를 뿐입니다.

#### ▼ 코드

- head 부분 코드는 부트스트랩 공식 문서의 '[Getting started](#)' 를 참고했어요.

```
<!doctype html>
<html lang="ko">

  <head>
    <!-- Required meta tags -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
          integrity="sha384-Gn5384xqq1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGFAW/dAiS6JXm" crossorigin="anonymous">

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
           integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
           crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
           integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCMY1"
           crossorigin="anonymous"></script>
```

```
<title>스파르타코딩클럽 | 부트스트랩 연습하기</title>
</head>

<body>
    <h1>이걸로 시작해보죠!</h1>
</body>

</html>
```

- <h1>~</h1> 바로 아래에 button 코드를 넣어봅니다.

```
<button type="button" class="btn btn-primary">Primary</button>
```

- 코드를 크롬에 띄워보세요.



예쁜 버튼이 생겼습니다! (부트스트랩이 미리 css를 작성해뒀기 때문)  
즉, btn 과, btn-primary 라는 class의 디자인 요소를 미리 정의해둔 것이죠.

#### ▼ 14) 부트스트랩 - 함께 만들어보며 코드 분석하기



참고자료: [부트스트랩 컴퍼넌트 페이지](#)에서 마음에 드는 컴퍼넌트들을 검색해 활용해보세요.

#### ▼ 아래와 같은 모양을 함께 만들어볼까요?

- 아래 파일을 클릭하면 브라우저에서 화면이 열립니다.

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7feb37f4-5666-4595-b983-7b7146  
3baaeb/bootstrap_component.html
```

- 우리가 만들 화면을 분석해보겠습니다!

##### 1) 문제 분석하기 - [크롬 개발자도구]를 열어서, Elements 탭에 있는 검사하기(Inspector) 클릭

- 단축키! Windows: F12 또는 alt + ctrl + i / Mac : option + command + i
- 화면에서 "마우스 오른쪽 클릭 → 검사"도 가능!
- 화면 요소는 큰 단위부터 보는게 좋습니다. HTML의 부모-자식 구조 기억나시나요? 상위 태그를 먼저 살펴봅니다.
  - `<div class="wrap">` 요소 안에 `<div class="jumbotron">` 과 `<div class="card-columns">` 로 이루어져있구나.

##### ▼ 화면

Elements Console ↗ 1 ⚙ ::

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="wrap">
      <div class="jumbotron"></div>
      <div class="card-columns" style="background-color: #e0f2fd; padding: 10px; margin-bottom: 20px;">
        <div class="card" style="width: 100%; height: 100%; position: relative; background-size: cover; background-position: center; border: none; border-radius: 0; overflow: hidden; margin-bottom: 10px;">
          
          <div class="card-body" style="position: absolute; bottom: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 0; padding: 10px; z-index: 2; font-size: 14px; color: #333; opacity: 0.8; transition: all 0.3s ease; transform: translateY(100%);">
            <h3>Hello, world!</h3>
            <p>This is a simple hero unit, a simplejumbotron-style component for calling out key information or content. It uses utility classes for typography and spacing to space content out within the card body.</p>
            <a href="#" style="color: #007bff; text-decoration: none; font-weight: bold;">Learn more
          </div>
        </div>
        <div class="card" style="width: 100%; height: 100%; position: relative; background-size: cover; background-position: center; border: none; border-radius: 0; overflow: hidden; margin-bottom: 10px;">
          
          <div class="card-body" style="position: absolute; bottom: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 0; padding: 10px; z-index: 2; font-size: 14px; color: #333; opacity: 0.8; transition: all 0.3s ease; transform: translateY(100%);">
            <h3>Card title that wraps to a new line</h3>
            <p>This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
          </div>
        </div>
        <div class="card" style="width: 100%; height: 100%; position: relative; background-size: cover; background-position: center; border: none; border-radius: 0; overflow: hidden; margin-bottom: 10px;">
          
          <div class="card-body" style="position: absolute; bottom: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 0; padding: 10px; z-index: 2; font-size: 14px; color: #333; opacity: 0.8; transition: all 0.3s ease; transform: translateY(100%);">
            <h3>Card title that wraps to a new line</h3>
            <p>This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
          </div>
        </div>
      </div>
    </div>
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKIkVtK3UNzm7KCCR//F5QDg6aAZGJwFDMVNA/GpFFF93XpGSKKN" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1OKtv3Rn7W3npPkhU9K/Sq0SAp7HUbz39J7faKPsKvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6spejhA02d8j0t6vLHfE/JQGrRSQoSxFWp1IMquvdAyUar5+76PVCmYL" crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css?family=Montserrat:400,500,700|Noto+Sans+KR:400,500,700|Roboto:400,500,700&display=swap" rel="stylesheet">
  </body>
</html>
```

html body div.wrap div.card-columns

Styles Computed Event Listeners DOM Breakpoints ↗

Filter :hover .cls +

# Hello, world!

This is a simple hero unit, a simple jumbotron-style component for calling out key information.

It uses utility classes for typography and spacing to space content out within the

[Learn more](#)

**div.jumbotron** 900 x 416  
Color ■ #212529  
Font 16px -apple-system, system-ui, "Segoe... □ E9ECEF  
Background □  
Margin 0px 0px 32px  
Padding 64px 32px

**Card title that wraps to a new line**

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

**Card title that wraps to a new line**

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

```
<!DOCTYPE html>
<html lang="en" class=">
  <head></head>
  <body>
    <div class="wrap">
      <div class="jumbotron"></div>
    ... <div class="card-columns" style="display: flex; flex-wrap: wrap; justify-content: space-around; gap: 10px;"></div>
      <div class="card"></div>
      <div class="card"></div>
      <div class="card"></div>
      <div class="card"></div>
      <div class="card"></div>
      <div class="card"></div>
    </div>
  </body>
</html>


<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3dHkTKKd0ZCfClI+JGnYqD8Q+X3Wq5J8vZgkQ&lt;!-->
  KJ3dHkTKKd0ZCfClI+JGnYqD8Q+X3Wq5J8vZgkQ&lt;!-->
  GopFF93hXpGSKKh" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh8Y1x0jIxiN4xZ4JwTPDfpbBZ8f9eNcJOQ&lt;!-->
  ApNbgh8Y1x0jIxiN4xZ4JwTPDfpbBZ8f9eNcJOQ&lt;!-->
  SCo5AP7hUi1bx39jTfaFpkvXusvfaf0b4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-7TBSNejhU02d8j0t6vHFheJOGlRRSSQx5fFWpi1MquDyAjuar5+76PVCayL" crossorigin="anonymous"></script>
<link href="https://fonts.googleapis.com/css?family=Montserrat:400,500,700|Noto+Sans+KR:400,500,700|Roboto:400,500,700&display=swap" rel="stylesheet">
<div class="liner-mini-tooltip liner-maintain-selection-liner-desktop" style="display: none;"></div>
<div class="liner-mini-button" style="display: none;"></div>
<div class="liner-tooltip-wrap liner-desktop"></div>
<div class="liner-comment-box"></div>
<div class="liner-upgrade-comment-box"></div>
<div class="liner-upgrade-highlight-box liner-maintain-selection"></div>
<div class="liner-mobile-comment-box"></div>
</body>
</html>
```

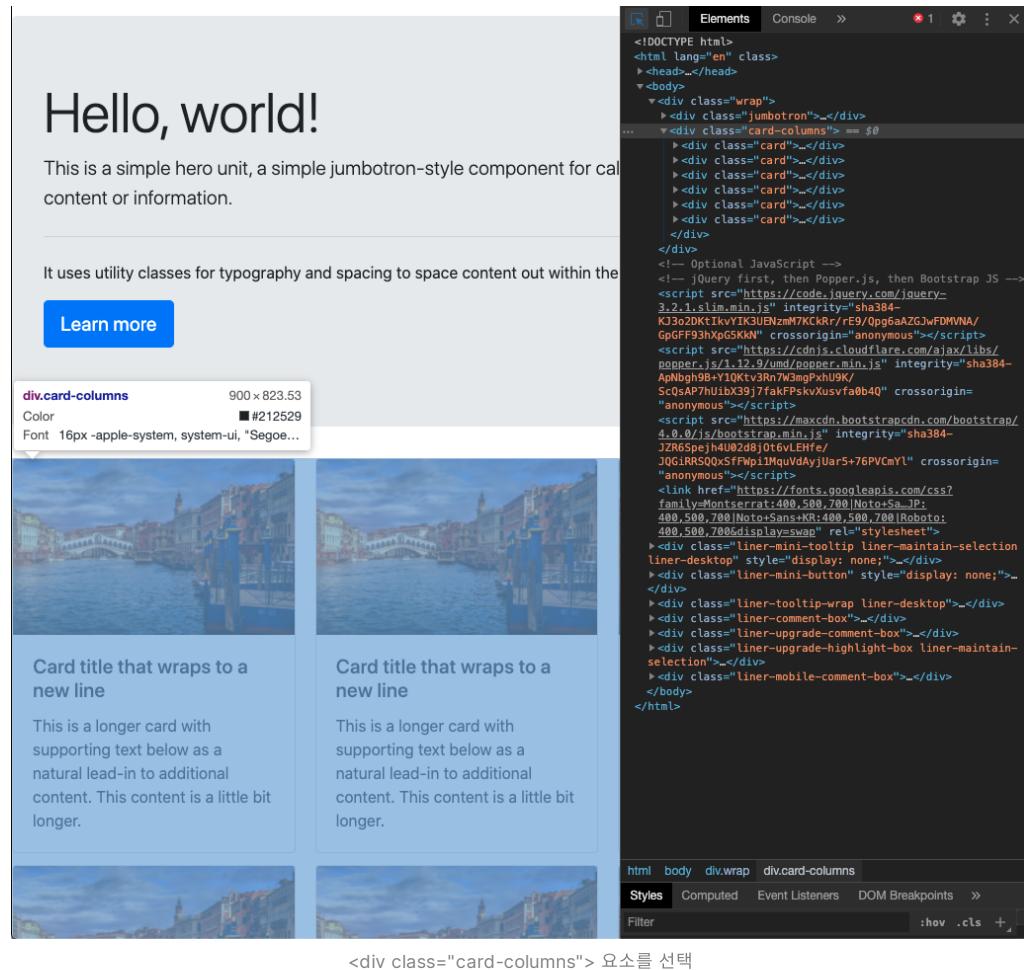
Elements Console >

html body div.wrap div.card-columns

Styles Computed Event Listeners DOM Breakpoints >

Filter :how .cls +

<div class="jumbotron"> 요소를 선택



## 2) 자료 검색 - 부트스트랩 컴퍼넌트 페이지에서 검색해보기

- class 명으로 검색해봅니다. 페이지에 나와있는 화면을 보고 내가 원하는 코드를 찾아봅니다.

### 3) 내 코드에 적용하기

- 부트스트랩 컴퍼넌트 페이지에서 샘플코드를 가져와서 적용해봅니다.
- 부트스트랩 컴퍼넌트 페이지에서 가져올 수 없는 디자인 요소(.wrap)를 샘플 코드에서 확인합니다.

#### ▼ 완성 코드

```
<!doctype html>
<html lang="ko">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
          integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6Jxm" crossorigin="anonymous">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
           integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
           crossorigin="anonymous"></script>

    <title>스파르타코딩클럽 | 부트스트랩 연습하기</title>

    <style>
        .wrap {
            width: 900px;
    
```

```

        margin: 10px auto;
    }

```

```
</head>
```

```
<body>
```

```

<div class="wrap">
    <div class="jumbotron">
        <h1 class="display-4">Hello, world!</h1>
        <p class="lead">This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.</p>
        <hr class="my-4">
        <p>It uses utility classes for typography and spacing to space content out within the larger container.</p>
        <p class="lead">
            <a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a>
        </p>
    </div>

```

```

<div class="card-columns">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```

    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```

    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```

    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```

    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```

    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Card title that wraps to a new line</h5>
            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>
        </div>
    </div>

```

```
</div>
</div>
</body>

</html>
```



앞으로 프로그래밍할 때에도 비슷한 과정을 거치면 됩니다. 문제를 정의/분석 → 자료 검색 → 자료 이해 → 내 코드에 적용

▼ 15) 📝나홀로 메모장 빠대 잡기(1)

▼ Q. 퀴즈설명: 아래 모양을 만들어보세요!

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c09936bb-e669-436a-93d0-688f0d2d7959/memo\\_template.html](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c09936bb-e669-436a-93d0-688f0d2d7959/memo_template.html)

💡 힌트:

1. 모든 글자는 직접 치지 말고 복사해서 붙여넣으면 금방 해결~!
2. "여기에 코멘트가 들어가죠"는 <p>태그로 파란색 굵은 글씨가 하나 더 추가됐고,
3. "여기 기사제목이 들어가죠"는 <a>태그로 하이퍼링크가 걸렸네요!

▼ A. 함께하기

▼ 완성본 코드

```
<!doctype html>
<html lang="ko">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous"

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js"
        integrity="sha384-ApNbghB+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin="anonymous"></script>
    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <title>스파르타코딩클럽 | 부트스트랩으로 나홀로 메모장 만들기</title>

    <!-- style -->
    <style type="text/css">
        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 900px;
            margin: auto;
        }

        .comment {
            color: blue;
            font-weight: bold;
        }
    </style>
</head>

<body>
<div class="wrap">
```

```

<div class="jumbotron">
  <h1 class="display-4">나홀로 링크 메모장!</h1>
  <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
  <hr class="my-4">
  <p class="lead">
    <a class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
  </p>
</div>
<div class="card-columns">
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
</div>
</body>
</html>

```

▼ 16) 🖌 포스팅 박스를 넣어보기(2)

▼ Q. 퀴즈설명: 아래 모양을 만들어보세요!

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c3c28674-a734-4937-b4e3-2bb32ebe64e5/memo\\_posting.html](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c3c28674-a734-4937-b4e3-2bb32ebe64e5/memo_posting.html)

💡 힌트:

1. 부트스트랩 컴포넌트 페이지에서 `forms`를 참고하세요!
2. 가운데로 가져오려면 똑같이 `width`, `margin`을 활용합니다.
3. 경계선을 둑글게 그리는 방법을 찾기 위해 다음을 구글링해봅니다.
  - border
  - border-radius
4. 경계선과 내부 요소의 간격은 `padding`으로 조절하는게 좋습니다.



참고! 일단 해두기!

복사해오는 내용 중 `form` 태그(`<form>...</form>`)가 있다면,  
`form`태그를 `div` 태그(`<div>...</div>`)로 바꿔주세요!

- `<form>`태그 안에 버튼이 있으면, 자동으로 새로고침이 됩니다.
- 우리는 이 기능을 안 쓸거라, 추후 충돌 방지를 위해 미리 바꿔두는 것입니다!

▼ A. 함께하기

▼ 완성본 코드

```
<!doctype html>
<html lang="ko">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5zXp5fGdC6WuZlDUqoAAQbqy6+LwDzJk/2u+eR9jzJLZoJZuHvX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js"
        integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin="anonymous"></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <title>스파르타코딩클럽 | 부트스트랩으로 나홀로 메모장 만들기</title>

    <!-- style -->
    <style type="text/css">
        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 900px;
            margin: auto;
        }

        .comment {
            color: blue;
            font-weight: bold;
        }

        .post-box {
            width: 500px;
            margin: 20px auto;
            padding: 50px;
            border: black solid;
            border-radius: 5px;
        }
    </style>
</head>

<body>
```

```

<div class="wrap">
  <div class="jumbotron">
    <h1 class="display-4">나홀로 링크 메모장!</h1>
    <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
    <p class="lead">
      <a class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
    </p>
  </div>
  <div class="form-post post-box">
    <div>
      <div class="form-group">
        <label for="exampleFormControlInput1">아티클 URL</label>
        <input class="form-control" placeholder="" type="text">
      </div>
      <div class="form-group">
        <label for="exampleFormControlTextarea1">간단 코멘트</label>
        <textarea class="form-control" rows="2"></textarea>
      </div>
      <button type="button" class="btn btn-primary">기사저장</button>
    </div>
  </div>
  <div class="card-columns">
    <div class="card">
      
      <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기에는 코멘트가 들어갑니다.</p>
      </div>
    </div>
    <div class="card">
      
      <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기에는 코멘트가 들어갑니다.</p>
      </div>
    </div>
    <div class="card">
      
      <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기에는 코멘트가 들어갑니다.</p>
      </div>
    </div>
    <div class="card">
      
      <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기에는 코멘트가 들어갑니다.</p>
      </div>
    </div>
    <div class="card">
      
      <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기에는 코멘트가 들어갑니다.</p>
      </div>
    </div>
  </div>
</div>
</body>

```

```
</html>
```

## [0.5시간] 첫 프로그래밍 언어 - Javascript 맛보기

### ▼ 17) Javascript(자바스크립트)란?

- 만약 화면에서 버튼을 클릭할 때마다 색깔을 바꾸고 싶다면? 아니면, div 박스를 하나 감추고 싶다면?
- 이렇게 화면에 동적으로 움직임을 주기 위해 사용하는 것이 Javascript(자바스크립트)입니다. Javascript(자바스크립트)는 프로그래밍 언어로, 웹 브라우저가 해석할 수 있는 언어입니다. 짧게 줄여 JS라고 표기하기도 합니다.

### ▼ 잠깐 상식



Q. 왜 브라우저는 Javascript만 알아들어요? HTML안에다 파이썬, Java 같은 언어를 써서 주면 안되나요?

A. 불가능한 이야기는 아닙니다. 다만, 이 "역사적인 이유 & 이미 만들어진 표준"이기 때문에, 모든 브라우저는 기본적으로 Javascript를 알아듣게 설계되어있고, 모든 웹서버는 HTML+CSS+Javascript를 응답(response) 데이터로 주게 되어있죠.



Q. Java와 Javascript는 어떤 차이가 있나요?

A. 인도와 인도네시아.. 바다와 바다코끼리.. 처럼 아무 관련 없습니다.



이론 설명 때, 클라이언트가 서버에 요청하면, 서버가 응답결과로 클라이언트에게 HTML+CSS+Javascript를 준다고 했던 것, 기억하시나요?

### ▼ 18) [💡튜터와 같이] Javascript - HTML 연결. 버튼을 클릭하면 경고창이 뜨게하기

- 처음 프로그래밍 언어를 배우면 생소한 부분이 있습니다. 다음주에 본격적으로 사용해보기 전에! 오늘 먼저 맛보기를 해보겠습니다.



일단, 따라쓰기 를 통해,  
Javascript가 HTML과 어떻게 연동되는지 알아보겠습니다.

### ▼ 💻코드! 여기서 시작하세요! - [memo.html](#)

- [memo.html](#) 이름으로 새 HTML 파일을 만들고, 아래 내용을 복사([ctrl\(cmd\) + c](#)), 붙여넣기([ctrl\(cmd\) + v](#)) 해주세요.

```
<!doctype html>
<html lang="ko">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
          integrity="sha384-Gn55XoQ1gjKtE7Y07zSF+JwXkC9ZUkXzjZLW4dI7lDfZj7q7dPCi乐" crossorigin="anonymous">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
           integrity="sha384-ApNbgh8B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q">
        crossorigin="anonymous"></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <title>스파르타코딩클럽 | 나홀로 메모장</title>

    <!-- style -->
    <style type="text/css">
```

```

        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 900px;
            margin: auto;
        }

        .comment {
            color: blue;
            font-weight: bold;
        }

        .post-box {
            width: 500px;
            margin: 20px auto;
            padding: 50px;
            border: black solid;
            border-radius: 5px;
        }
    
```

</style>

</head>

<body>

<div class="wrap">

<div class="jumbotron">

<h1 class="display-4">나홀로 링크 메모장!</h1>

<p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>

<hr class="my-4">

<p class="lead">

<a class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>

</p>

</div>

<div class="form-post post-box">

<div>

<div class="form-group">

<label for="exampleFormControlInput1">아티클 URL</label>

<input class="form-control" placeholder="" type="text">

</div>

<div class="form-group">

<label for="exampleFormControlTextarea1">간단 코멘트</label>

<textarea class="form-control" rows="2"></textarea>

</div>

<button type="button" class="btn btn-primary">기사저장</button>

</div>

</div>

<div class="card-columns">

<div class="card">



<div class="card-body">

<a href="#" class="card-title">여기 기사 제목이 들어가죠</a>

<p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리<br>

<p class="card-text comment">여기\_comment>에 코멘트가 들어갑니다.</p>

</div>

</div>

<div class="card">



<div class="card-body">

<a href="#" class="card-title">여기 기사 제목이 들어가죠</a>

<p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리<br>

<p class="card-text comment">여기\_comment>에 코멘트가 들어갑니다.</p>

</div>

</div>

<div class="card">



<div class="card-body">

<a href="#" class="card-title">여기 기사 제목이 들어가죠</a>

<p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리<br>

<p class="card-text comment">여기\_comment>에 코멘트가 들어갑니다.</p>

</div>

</div>

```

        </div>
    </div>
    <div class="card">
        
        <div class="card-body">
            <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
            <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리</p>
            <p class="card-text comment">여기_comment>에 코멘트가 들어갑니다.</p>
        </div>
    </div>
    <div class="card">
        
        <div class="card-body">
            <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
            <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리</p>
            <p class="card-text comment">여기_comment>에 코멘트가 들어갑니다.</p>
        </div>
    </div>
</div>
</body>

</html>

```

- 함수 만들기

- 아래 코드를 통해 간단한 사용방법을 알아봅니다. `<head> ~ </head>` 안에 `<script> ~ </script>` 를 만들고 아래 코드를 넣어 보겠습니다.

```

function sayHello(){
    alert('안녕!');
}

```



- 버튼에 함수를 연결하기. 버튼을 누르면 함수가 불립니다.

```

<button onclick="sayHello()" type="button" class="btn btn-primary">기사저장</button>

```

### ▼ 코드! 완성본

```

<!doctype html>
<html lang="ko">

    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
              integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

        <!-- JS -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" ></script>

```

```

integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous">></script>

<!-- 구글폰트 -->
<link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

<title>스파르타코딩클럽 | 나홀로 메모장</title>

<!-- style -->
<style type="text/css">
    * {
        font-family: 'Stylish', sans-serif;
    }

    .wrap {
        width: 900px;
        margin: auto;
    }

    .comment {
        color: blue;
        font-weight: bold;
    }

    .post-box {
        width: 500px;
        margin: 20px auto;
        padding: 50px;
        border: black solid;
        border-radius: 5px;
    }
</style>
<script>
    function sayHello() {
        alert('안녕!');
    }
</script>

</head>

<body>
<div class="wrap">
    <div class="jumbotron">
        <h1 class="display-4">나홀로 링크 메모장!</h1>
        <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
        <hr class="my-4">
        <p class="lead">
            <a class="btn btn-primary btn-lg" href="#" role="button">포스팅박스 열기</a>
        </p>
    </div>
    <div class="form-post post-box">
        <div>
            <div class="form-group">
                <label for="exampleFormControlInput1">아티클 URL</label>
                <input class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="exampleFormControlTextarea1">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
            <button onclick="sayHello()" type="button" class="btn btn-primary">기사저장</button>
        </div>
    </div>
    <div class="card-columns">
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리</p>
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리</p>
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
</div>
</body>

</html>

```



### 잠깐만!

원래는 JS를 HTML파일 내에서 사용할 때,

- 1) CSS 파일처럼 JS파일을 분리하고 HTML 코드에 추가해서 사용하거나,
- 2) </body> 바로 직전에 <script> ~ </script> 코드를 넣습니다.

하지만 우리는 배우는 과정이므로, JS코드가 파일 앞에 있는게 눈에 잘 띠고, head에 넣어도 코드를 실행하는데 큰 문제가 없기 때문에 head 사이에 두도록 하겠습니다.

## [0.5시간] 첫 프로그래밍 언어 - Javascript 기초 문법

- ▼ 19) 본격적으로, 문법을 배워볼까요?



1주차엔 기본 문법을 배우고, 2주차에 이리저리 연습을 많~이 할거예요!

- ▼ [크롬 개발자도구]를 열어서, console 탭에 작성합니다!



Q. 크롬 개발자도구 콘솔창은 어떤 의미인가요?

A. 띄워놓은 페이지에서 빠르게 자바스크립트를 테스트해볼 수 있게, 개발자들을 위해 만들어둔 도구입니다. 새로 고침하면 모두 사라진다는 사실!

(console 창에서 입력하면 실행에 대한 값이 바로 보이지만, 우리는 연습하기 위해서 console.log를 사용해서 값을 확인해볼거에요.)

- 단축키! Windows: F12 또는 alt + ctrl + i / Mac : option + command + i

- 출력하기 연습

- `console.log(변수)`

- 콘솔 창에 괄호 안의 값을 출력해줍니다. 개발자가 결과값을 보기 편하도록 변수를 출력합니다.

- `console.log(변수1, 변수2)`로 여러 변수를 한 번에 출력할 수도 있어요.

아래 코드를 console에 복사해서 붙여넣고 엔터키를 쳐서 실행해보세요.

```
console.log("Hello World!");
```



출력하기는 값을 확인하거나, 에러를 찾을 때 자주 쓰입니다. 출력해서 버그찾기(Debugging By Printing) 스킬!

- 보통의 프로그래밍 언어는 아래와 같은 구성요소로 되어있어요. 언어가 명사, 동사, 목적어 등으로 구성되어 있는 것과 비슷하답니다.

▼ 변수(Variable) & 기본연산

- 변수 대입 (`a = 2`)의 의미: "오른쪽에 있는 것을 왼쪽에 넣는 것!"

(2를 a라는 변수에 넣는다)

- `let` 으로 변수를 선언합니다.

```
let temp = 20;
console.log(temp);

temp = 'Harry';
console.log(temp);

// 변수는 값을 저장하는 박스예요.
// 한 번 변수를 선언했으면, 다시 선언하지 않고 값을 넣습니다.
```

- 사칙연산, 그리고 문자열 더하기가 기본적으로 가능합니다.

```
let num1 = 1
let num2 = 2

console.log(num1 + num2) // 실행하면 3 출력
console.log(num1 / num2) // 실행하면 0.5 출력
```

```
let firstName = 'Harry'
let lastName = 'Potter'

console.log(firstName + lastName) // 실행하면 'HarryPotter'
console.log(firstName + ' ' + lastName) // 실행하면 'Harry Potter'
console.log(lastName + num1) // 실행하면 Harry1 -> 문자+숫자를 하면, 숫자를 문자로 바꾼 뒤 수행합니다.
```

- 변수명은 아무렇게나 지을까요?

- `5분 개발자 맷 - 파일/폴더 관리`에서 배운 것처럼 이름짓기(naming)는 중요합니다. 담고 있는 정보를 잘 표현해야 합니다.

- 변수명은 숫자로 시작되거나, 일부 특수문자 또는 띄워쓰기를 사용할 수 없습니다

▼ 이름짓기(naming) 형식 - 코딩 컨벤션(코딩 규칙, coding convention)

- 이름을 지을 때에는 데이터를 잘 표현하는 것으로 지어야한다는 거 기억나시나요? 이름을 작성하는 형식에 규칙이 있습니다.

- 이러한 규칙들을 포함해 특정 프로그래밍 언어로 프로그래밍할 때의 일련의 지침을 **코딩 컨벤션(coding convention)**이라고 합니다. 해당 언어로 작성된 프로그램에서의 프로그래밍 스타일, 권장하는 원칙 등을 포함합니다.

- 코딩 컨벤션을 지키면, 코드가 일관된 스타일을 가지게 됩니다. 나만 알아볼 수 있는 코드보다 누구나 알아볼 수 있는 코드가 협업을 위해 중요하겠죠?

- 코딩 컨벤션에는 많은 것들이 있지만, 우리는 일단 이름 작성 형식만 따르겠습니다.

```
// 단어를 _ 로 연결합니다. 뱀 모양 같다고 해서 snake case라고 부릅니다.
let first_name = 'Harry'

// 우리가 사용할 'camel case' 형식
// 단어가 연결되는 부분의 첫 글자를 대문자로 씁니다.
// 낙타의 혹을 굽았다고 해서 camel case라고 합니다.
let firstName = 'Harry'
```

- 숫자, 문자형 외에도, '참 거짓 True False' 를 나타내는 자료형도 있습니다.

```
let wizardStatus = true;

let myTruth = (90 > 50);
let myFalse = (90 < 50);

console.log(myTruth); // true
console.log(myFalse); // false

let age = 30;

myTruth = (age == 30); // 같다 ( == )
console.log(myTruth);

myTruth = (age != 40); // 다르다 ( != ) != 는 not을 의미합니다
console.log(myTruth);

myFalse = (age == 40);
console.log(myFalse);
```

- 주로 '조건문'과 함께 쓰입니다. 어떻게 쓰이는지는 다음주 조건문을 배우며 같이 알아보시죠!

#### ▼ 자료형(Data Type) - 리스트(List) & 딕셔너리(Dictionary)

- 자료(Data)를 담는 형태(type) 입니다. 많은 자료형이 있지만, 가장 많이 쓰이는 두 가지 자료형을 배워보겠습니다.
- 리스트(List): 순서를 지켜서 가지고 있는 형태입니다.

```
// 변수 이름은 의미를 알 수 있게, 원하는 대로 짓기
let sampleList = []; // 리스트를 선언

// 또는,

let multiList = [1,2,'hey',3]; // 로 선언 가능

// 대부분의 프로그래밍 언어는 index가 0 부터 시작합니다.
console.log(multiList[1]); // 2 를 출력
console.log(multiList[2]); // 'hey'를 출력

// 리스트에 요소 넣기
multiList.push(' hey! ');
console.log(multiList); // [1, 2, "hey", 3, " hey! "] 를 출력

// 리스트의 길이 구하기 - 반복문에서 유용하게 쓰여요!
console.log(multiList.length); // 5를 출력
```

- 딕셔너리(Dictionary): 키(key)-값(value) 값 쌍. 사전이 표제어-의미로 쌍을 이루는 걸 연상해보세요.

```
let sampleDict = {};// 딕셔너리 선언. 변수 이름은 역시 아무렇게나 가능!

// 또는,

let wizard = {'name':'Harry Potter','age':40}; // 로 선언 가능
console.log(wizard['name']); // 'Harry Potter'를 출력
console.log(wizard['age']); // 40을 출력

wizard['height'] = 173; // 딕셔너리에 키:밸류 넣기
console.log(wizard); // {name: "Harry Potter", age: 40, height: 173}을 출력
```

- 리스트와 딕셔너리의 조합

```
let wizards = [{name:'Harry Potter','age':40},{'name':'Ron Weasley','age':40}]
```

```

console.log(wizards[0]['name']);
// wizards[0]['name']의 값은? 'Harry Potter'
console.log(wizards[1]['name']);
// wizards[1]['name']의 값은? 'Ron Weasley'

newWizard = {'name':'Albus Potter', 'age':14};
wizards.push(newWizard);

console.log(wizards);
// wizards의 값은? [{name:'Harry Potter', age:40}, {name:'Ron Potter', age:40}, {name:'Albus Potter', age:14}]
console.log(wizards[2]['name']);
// wizards[2]['name']의 값은? 'Albus Potter'

```

- 왜 자료형이 필요할까요?



**순서를 표시할 수 있고, 정보를 묶을 수 있습니다.**

<스파르타과일가게>가 정말 잘 되어서 전국에서 손님이 찾아오고 있습니다. 대기표를 작성하기 위해서 온 순서대로 이름, 휴대폰 번호를 적도록 하였는데요. 변수만을 사용한 모습은 다음과 같습니다.

```

let customer1Name = '김스파';
let customer1Phone = '010-1234-1234';
let customer2Name = '박르탄';
let customer2Phone = '010-4321-4321';
...(휴, 알아보기 힘듭니다.)

```

👉 딕셔너리를 활용한다면 다음과 같이 고객 별로 정보를 묶을 수 있습니다.

```

let customer1 = {'name': '김스파', 'phone': '010-1234-1234'};
let customer2 = {'name': '박르탄', 'phone': '010-4321-4321'};

```

👉 그리고 순서를 나타내기 위해 리스트를 사용하면, 이렇게나 깔끔해집니다.

```

let customers = [
    {'name': '김스파', 'phone': '010-1234-1234'},
    {'name': '박르탄', 'phone': '010-4321-4321'}
]

```

✓ 보기에도 깔끔해지고, 다루기도 쉬워지고, 고객이 새로 한 명 더 오더라도 .push 함수를 이용해 간단하게 대응할 수 있습니다.

## ▼ 기본 함수들

- 사칙연산 외에도, JS에서 기본적으로 제공하는 여러 함수들이 존재합니다.



왠지 이건 있을 것 같은데?(예 - 특정 문자를 바꾸고 싶다 등) 싶으면 직접 만들지 말고 **구글에서 먼저 검색해보세요!**

- 예를 들면, '나눗셈의 나머지'를 구하고 싶은 경우

```

let a = 20;
let b = 7;

console.log(a % b); // 6

```

- 또, 모든 알파벳을 대문자로 바꾸고 싶은 경우

```

let myClub = 'spartacodingclub';
console.log(myClub); //'spartacodingclub'

let upperCase = myClub.toUpperCase();
console.log(upperCase); // SPARTACODINGCLUB

```

- 또, 특정 문자로 문자열을 나누고 싶은 경우

```
let myEmail = 'sparta@gmail.com';

let result = myEmail.split('@'); // ['sparta','gmail.com']

console.log(result[0]); // sparta
console.log(result[1]); // gmail.com

let domain = result[1].split('.'); // ['gmail','com']

console.log(domain[0]); // gmail -> 우리가 알고 싶었던 것!
console.log(domain[1]); // com

let mailDomain = myEmail.split('@')[1].split('.')[0]; // gmail -> 간단하게 쓸 수도 있다!
console.log(mailDomain);
```

- 특정 문자로 나누고 싶은 경우 2

```
let txt = '서울시-마포구-망원동'
// 특정 문자('-')로 나누기
let names = txt.split('-'); // ['서울시', '마포구', '망원동']
console.log(names);

// 특정 문자('>')로 합치고 싶은 경우

let resultAddress = names.join('>'); // '서울시>마포구>망원동' (즉, 문자열 바꾸기!)
console.log(resultAddress);
```

## ▼ 함수 - 내가 만들기

- 기본 생김새

```
// 만들기
function 함수이름(필요한 변수들) {
    내릴 명령어들을 순차적으로 작성
}
// 사용하기
함수이름(필요한 변수들);
```

- 예시

```
// 두 숫자를 입력받으면 더한 결과를 돌려주는(return) 함수
function sum(num1, num2) {
    console.log('num1: ', num1, ', num2: ', num2);
    return num1 + num2;
}

let sumResult;
sumResult = sum(3, 5); // 8
console.log(sumResult);

sumResult = sum(4, -1);
console.log(sumResult); // 3
```

```
// 출생년도와 계산하고 싶은 연도를 입력받아 나이를 계산해 돌려주는(return) 함수
function calculateAge(calYear, birthYear) {
    console.log('출생년도: ', birthYear, ', 계산하고 싶은 연도: ', calYear);
    return calYear - birthYear;
}

let calAge = calculateAge(2020, 1980);
console.log(calAge); // 40

// 잠깐! 값을 입력하는 순서를 바꾸어도 원하는 값이 나오나요?
console.log(calculateAge(1980, 2020)); // -40
// 아니네요! 입력하는 순서도 맞춰줘야하는군요!
```

- 조건문, 반복문은 다음주에 배웁니다!

## [0.5시간] 내 프로젝트 상상하기

### ▼ 20) 내 프로젝트 상상하기

- 목표를 확실해야 힘도 나죠! 같은 내용을 보고 따라하는 것보다 스스로 프로젝트를 만들고 적용해보면서 훨~씬 더 많이 배울 수 있어요! 내가 만들고 싶은 프로젝트를 자유롭게 상상해보세요.
- 1) 수업시간에 만드는 프로젝트에서 아이템을 살짝 바꿔봐도 되고,
- 2) 평소에 관심있는 주제로 새로운 기능을 만들어봐도 되요.

### ▼ 상상의 나래를 펼치기 위한 자료는 여기있어요

#### 수업 시간에 만들 세 개의 프로젝트

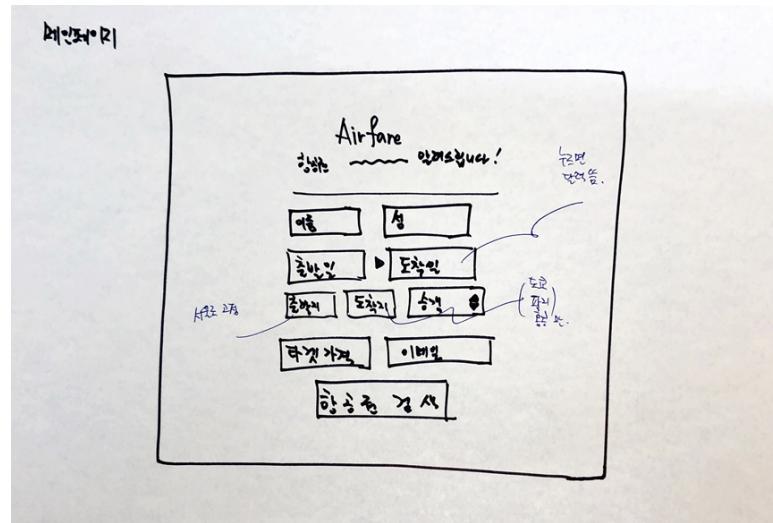
- 나홀로메모장: 링크를 눌러서 확인해보기
- 모두의책리뷰: 링크를 눌러서 확인해보기
- マイペイボリット무비스타: 링크를 눌러서 확인해보기
- 아이디어 창고: 스파르타코딩클럽 포트폴리오 페이지

### ▼ 21) 내가 생각한 프로젝트를 우리반 동료에게 소개하기

- 아래 내용을 간단히 적어 우리반 슬랙 채널에 메시지로 올려주세요. 그리고 3분정도 짧게 발표하기!

- 프로젝트 이름
- 기능 설명
- 참고한 프로젝트/웹사이트가 있다면 링크
- (선택)프로젝트 생김새

### ▼ 프로젝트 생김새(layout) 예시!



잠깐만, 슬랙 채널에만 소개하는 것이 아깝다면 개발일지에 적어보세요.

내가 쓰는 블로그(네이버, velog, Medium 등등 모두 좋아요)에 자유롭게 개발하면서 드는 생각, 배운 것을 적어봅니다. 보여주기 위한 것이 아니라 스스로를 위해 작성해보세요.

TIL (today I learned) 를 검색하면, 많은 분들의 개발일지를 볼 수 있어요.

## 😎 문제뱅크

- JS의 조건문과 반복문을 마저 배운 후에, 다음주에 찾아올게요 😊

## 💡 소화 타임



수업 내용을 배운 후, 소화 타임을 가집니다. 소화타임 = 복습 & 숙제 시작 타임!

### ▼ 소화 타임이 왜 중요할까요?

- 5시간의 내용을 한참 있다가 다시 보려면, 이해도 잘 안가고 깨먹기도 쉬워요.
- 그래서 오늘 이 자리를 뜨기 전에! 내 것으로 만들어보는 시간을 갖습니다.
- 복습 또는 숙제를 시작하고, 튜터에게 질문해서 아직 맞춰지지 않은 퍼즐 을 맞춰가세요.
- **가장 좋은 방법은, 수업 내용을 다시 보는 게 아니라, (의외로) 숙제를 바로 시작하는 것입니다.** 숙제를 하다보면, 수업 내용은 자연스럽게 들춰보게 될거예요!

## 숙제 설명



해당 주차에 배운 내용을 복습할 수 있도록 구성되었습니다. 숙제 파일은 sparta 폴더 밑에 homework 폴더에 넣어주세요.

### ▼ 숙제 구성

- 6~8주차에 팀프로젝트를 진행하려면 1~5주차에 강의내용을 최대한 이해하는 것이 중요합니다. 그리고 이를 위해서는 복습만 한 것이 없습니다!

### ▼ 의지를 사기 위한 벌금 제도

- 강의 시작 후 2만원을 예치금으로 받고, 숙제를 하지 않을 경우 5천 원씩 차감합니다. (숙제는 총 4번 있어요)
- 모아진 예치금은 반 회식비로 사용합니다. 공금은 튜터가 관리합니다.
- 달달한 맛, 짭짤한 맛 두 숙제 중 하나만 해도 숙제한 것!



달달한 맛 vs 짭짤한 맛 중 하나만 골라서 하시면 됩니다! :-)

숙제는 되도록 수업 D-1 까지 하기!

## [숙제 - 달달한 맛 ] - 웹페이지 쇼핑몰에 JS 함수 붙이기 (30분 예상)



주어진 웹페이지 쇼핑몰 코드에

기능: 주문하기 버튼을 클릭했을 때 '주문이 완료되었습니다.'라는 알럿(alert)을 띄워주세요!

- 아래 코드를 복사해서 `index.html` 제목으로 파일을 만들어주세요. 아래 코드에 알럿을 붙이면 끝!

### ▼ 💻 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <!-- Webpage Title -->
    <title>나홀로 쇼핑몰</title>

    <!-- Required meta tags -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
          integrity="sha384-Gn5JXmQD6c3ZTKAQ3+D8TSqIxaodvHcWtP5MqE6L8oL4u0DqF3VWZl9Jw" crossorigin="anonymous">

    <!-- JS -->
```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
       integrity="sha384-ApNbgh8B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
       crossorigin="anonymous"></script>

<!-- 구글폰트 -->
<link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

<style type="text/css">
    * {
        font-family: 'Stylish', sans-serif;
    }

    .wrap {
        width: 500px;
        margin: auto;
    }

    .img {
        background-image: url('https://www.conscious-skincare.com/wp-content/uploads/2016/02/glc-candle-lit-with-new-gi
        background-size: cover;
        background-position: center;
        width: 500px;
        height: 300px;
    }

    .info {
        margin-top: 20px;
        margin-bottom: 20px;
    }

    h1, h5 {
        display: inline;
    }

    .order {
        text-align: center;
    }

    .orders {
        margin-top: 100px;
    }
</style>

<script>
    function order() {
        // 여기에 코드를 작성하세요.
    }
</script>
</head>
<body>
<div class="wrap">
    <div class="img"></div>
    <div class="info">
        <h1>양초를 팝니다</h1>
        <h5>가격: 3,000원/개</h5>
        <p>이 양초는 사실 특별한 힘을 가지고 있어요. 양초를 켜고 소원을 빌면 짜자잔 뭐든지 이뤄지게 된답니다. 하나 사가세요! 대나무 향이 아주 좋아요</p>
    </div>
    <div class="info">
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">주문자 이름</span>
            </div>
            <input type="text" class="form-control" id="order-name" aria-describedby="basic-addon3">
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <label class="input-group-text" for="inputGroupSelect01">수량</label>
            </div>
            <select class="custom-select" id="order-count">
                <option selected value=""> -- 수량을 선택하세요 --</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
            </select>
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">주소</span>
            </div>
            <input type="text" class="form-control" id="order-address" aria-describedby="basic-addon3">
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">전화번호</span>
            </div>
    </div>
</div>

```

```

        <input type="text" class="form-control" id="order-phone" aria-describedby="basic-addon3">
    </div>
    <div class="order">
        <button onclick="order()" type="button" class="btn btn-primary">주문하기</button>
    </div>
    <div class="orders">
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">이름</th>
                    <th scope="col">수량</th>
                    <th scope="col">주소</th>
                    <th scope="col">전화번호</th>
                </tr>
            </thead>
            <tbody id="orders-box">
                <tr>
                    <td>박근태</td>
                    <td>10000</td>
                    <td>르탄시 코딩구 열심동</td>
                    <td>010-1234-5678</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
</body>
</html>

```

### [숙제 - 짭짤한 맛 🍜] - 나만의 원페이지 쇼핑몰 만들기 (2시간 예상)



아래 기획서를 보고, 부트스트랩 또는 템플릿을 활용해서 원페이지 쇼핑몰의 메인 페이지를 완성해주세요. (아이템은 가상으로, 아무거나 파셔도 좋습니다.^^;;)

기능: 주문하기 버튼을 클릭했을 때 '주문이 완료되었습니다.'라는 얼럿을 띄워주세요.

▼ 기획서(레이아웃) 보기

상품 사진

## 상품제목

가격: 1,000,000원 / 개

이 상품으로 말할 것 같으면 용왕이 찾지 못한 토끼의 간을 찾아 빨간망토의 늑대와 아기 돼지 삼형제의 숨결이 고루 녹아 있는 먹으면 120세까지 살 수 있다는 이 상품으로 말할 것 같으면 용왕이 찾지 못한 토끼의 간을 찾아 빨간망토의 늑대와 아기 돼지..

## 주문하기

주문자 성함:

수량:

▼

주소:

전화 번호:

주문하기

▼ 예시 화면



## 양초를 팝니다 가격: 3,000원/개

이 양초는 사실 특별한 힘을 가지고 있어요. 양초를 켜고 소원을 빌면 짜자잔 뿐만 아니라 지게 된답니다. 하나 사가세요! 대나무 향이 아주 좋아요

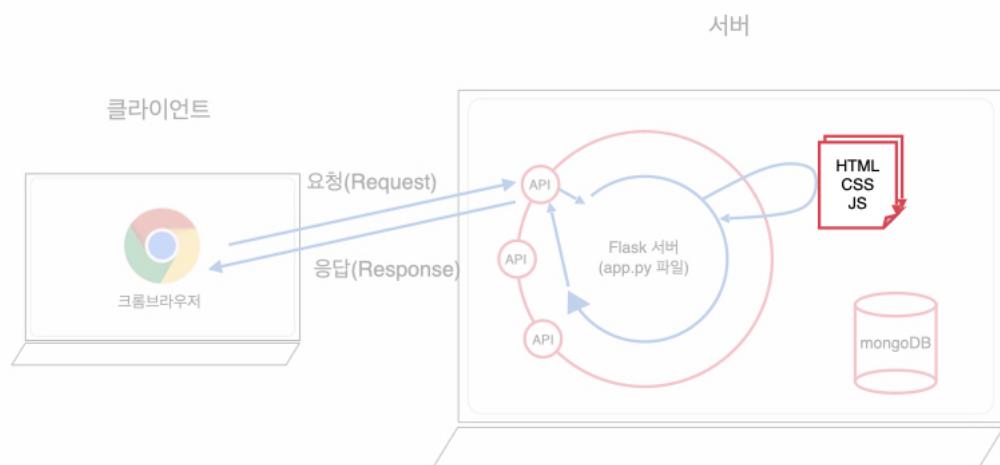
주문자 이름		
수량	-- 수량을 선택하세요 --	▼
주소		
전화번호		
<a href="#">주문하기</a>		

### [숙제 제출]

- 숙제 파일을 슬랙에 올리면 끝!

### 복습 도우미

- 이번주 배운 내용을 빠르게 복습할 수 있게 준비했어요!
- ▼ 웹 기초 동작 원리
  - 클라이언트의 요청(Request)에 서버가 응답(response)해서 데이터를 보내줍니다.
  - 보내줄 응답(Response) 데이터인 HTML,CSS 를 배우고, Javascript 기초를 조금 배워봤습니다.



### ▼ 전반부 주요 키워드

- 웹 기초 동작 원리 그림
- HTTP: 웹은 HTTP라는 규약(규칙)을 따릅니다. url에서 `http://` 가 바로 HTTP 라는 통신규약을 따른다는 표시예요.
- 클라이언트 : HTTP에서 요청을 하는 쪽
- 서버 : HTTP에서 요청을 받아 응답하는 쪽
- HTML, CSS, JS(Javascript) 는 화면을 구성합니다.
- HTML 은 뼈대, CSS 는 예쁘게 꾸미는 디자인을 담당합니다

### ▼ 후반부 주요 키워드

- 부트스트랩(Bootstrap): 웹사이트 화면을 쉽게 만들 수 있도록 도와주는 일종의 디자인 키트
- JS(JavaScript, 자바스크립트): 프로그래밍 언어로, 웹 브라우저가 해석할 수 있는 언어입니다. JS 의 더 많은 기능은 2주차에 배웁니다.
- JS 프로그래밍 언어 문법 : 변수, 자료형, 함수(요기까지 배웠습니다), 조건문, 반복문

## [0.5시간] 스스로 소화 타임 시작

- 오늘 배운 것을 복습하고, 본격적으로 숙제를 시작합니다. 모르는 것이 있으면 튜터에게 질문합니다.

## 체크아웃

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! ([링크](#)).
- 체크인처럼, 현재 본인의 감정상태와 수업후기에 관해 이야기합니다.
- 자유롭게 해주셔도 좋고, **KPT**에 맞춰해주셔도 좋아요.
  - Keep : 오늘 수업을 하면서 좋았던 것, 앞으로도 할 행동 / Problem : 아쉬워서 고쳐보면 좋을 것 / Try : 고치기 위해 내가 할 시도
  - 예) (Keep)오늘 다같이 자기소개를 해서 좋았어요. 친해진 기분! (Problem) 그런데 0주차 사전과제 중에 안한 게 있어서 쉬는 시간에 했어요. (Try)수업 전날 지난주 강의자료 확인하고 숙제하할게요!
- 튜터님은 체크아웃과 함께 출석 체크([링크](#))를 진행해주세요!
- 출석체크 - (변동이 있다면 다시 제출해주세요!)

스파르타코딩클럽 출석체크  
<http://spartacodingclub.shop/attendance>