Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with

페이지

클론!

https://radiant-maamoul-80f681.netlify.app/

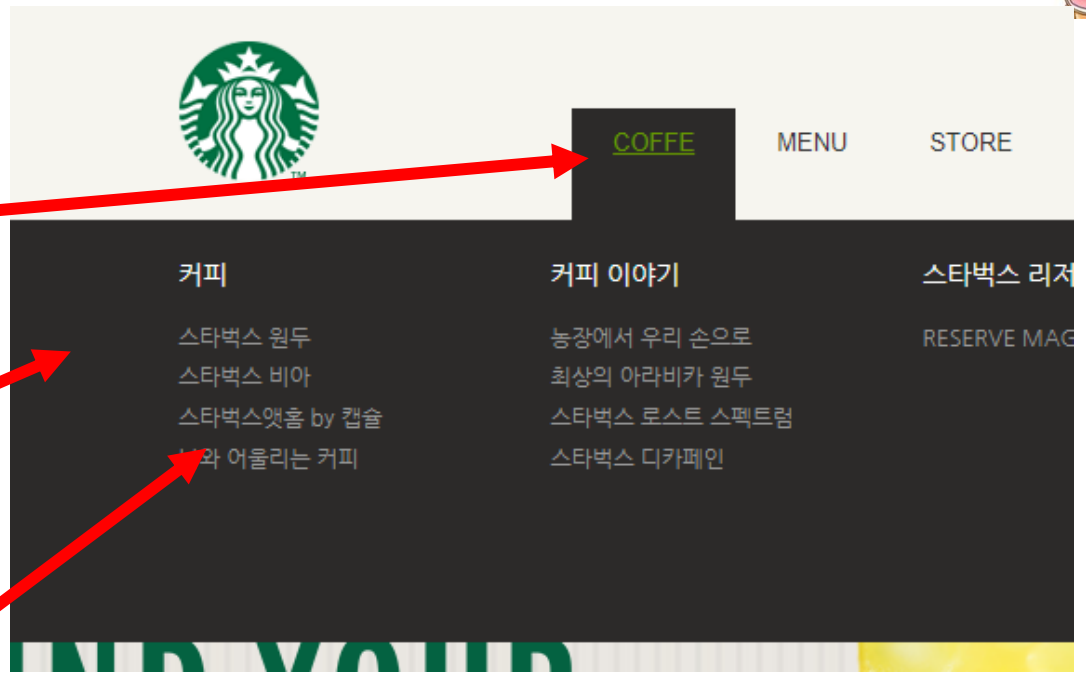
메인 메뉴

item

Item__name

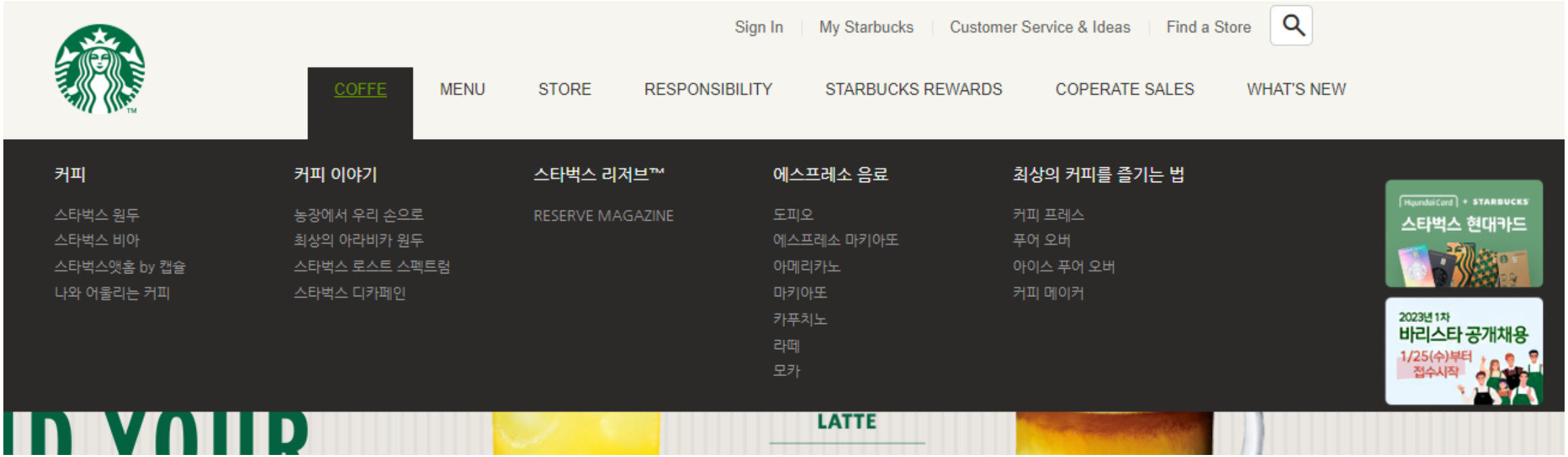
Item__contents

Item__contents_
_menu



```css
header .main-menu .item .item__contents {
  position: absolute;
  width: 10%;
}
```

```css
header .main-menu .item .item__contents {
  display: none;
  position: fixed;
  width: 100%;
  left: 0;
  top: 120px;
}
```



메인 메뉴 컨텐츠, 상단 컨텐츠 CSS

- 먼저 가로 배치를 해야겠죠?
- 배경색 적용: #2c2a29
- 넓이는 .inner 1100px 의 공간에 5개의 메뉴가 들어가므로? → 220px 씩 사용이 가능
  - Width: 200px
  - Padding: 20px 20px 0 0;
- 제목 <a> 태그
  - 흰색 / font-size: 15px / bottom 간격 주기

```css
header .main-menu .item .item__contents .item__contents__menu {
  background-color: #2c2a29;
  padding: 20px 0;
}

header .main-menu .item .item__contents .item__contents__menu .inner {
  display: flex;
  flex-wrap: wrap;
}

header .main-menu .item .item__contents .item__contents__menu .inner > li {
  width: 200px;
  padding: 20px 20px 0 0;
}

header .main-menu .item .item__contents .item__contents__menu .inner > li >
a {
  display: block;
  margin-bottom: 15px;
  color: white;
  font-size: 14px;
}
```

메뉴

Animation?

```css
header .main-menu .item .item__contents {
  position: fixed;
  width: 100%;
  left: 0;
  top: 120px;
  transform: scaleY(0);
  transform-origin: center top 0;
}

header .main-menu .item:hover .item__contents {
  transition: 0.5s;
  transform: scaleY(1);
}
```


JS

Badges

Sign In | My Starbucks | Customer Service & Ideas | Find a Store

COFFEE MENU STORE RESPONSIBILITY STARBUCKS REWARDS CORPORATE SALES WHAT'S NEW

– HAPPY 2023 –

FIND YOUR LUCK

자세히 보기

GOLDEN MIMOSA GREEN TEA

골든 미모사 그린 티

BLACK RICE LATTE

블랙 햅쌀 고봉 라떼

STARBUCKS® TURMERIC LATTE

스타벅스 튜메릭 라떼

```css
/* BADGES */
header .badges {
  position: absolute;
  top: 150px;
  left: calc(1100px + (100vw - 1100px) / 2);
}

header .badges .badge {
  cursor: pointer;
}

header .badges .badge img {
  width: 140px;
}

@media screen and (max-width: 1400px) {
  header .badges {
    left: calc(100vw - 150px);
  }
}
```

Visual

Sign In | My Starbucks | Customer Service & Ideas | Find a Store

COFFEE MENU STORE RESPONSIBILITY STARBUCKS REWARDS CORPORATE SALES WHAT'S NEW

– HAPPY 2023 –

FIND YOUR LUCK

자세히 보기

GOLDEN MIMOSA GREEN TEA

골든 미모사 그린 티

BLACK RICE LATTE

블랙 햅쌀 고봉 라떼

STARBUCKS® TURMERIC LATTE

스타벅스 튜메릭 라떼

Notice

공지사항 탄소 중립 포인트제 실천 안내

스타벅스 프로모션

Reward

STARBUCKS®
REWARDS

스타벅스만의 특별한 혜택, 스타벅스 리워드

스타벅스 회원이세요? 로그인을 통해 나만의 리워드를 확인해보세요.
스타벅스 회원이 아니세요? 가입을 통해 리워드 혜택을 즐기세요.

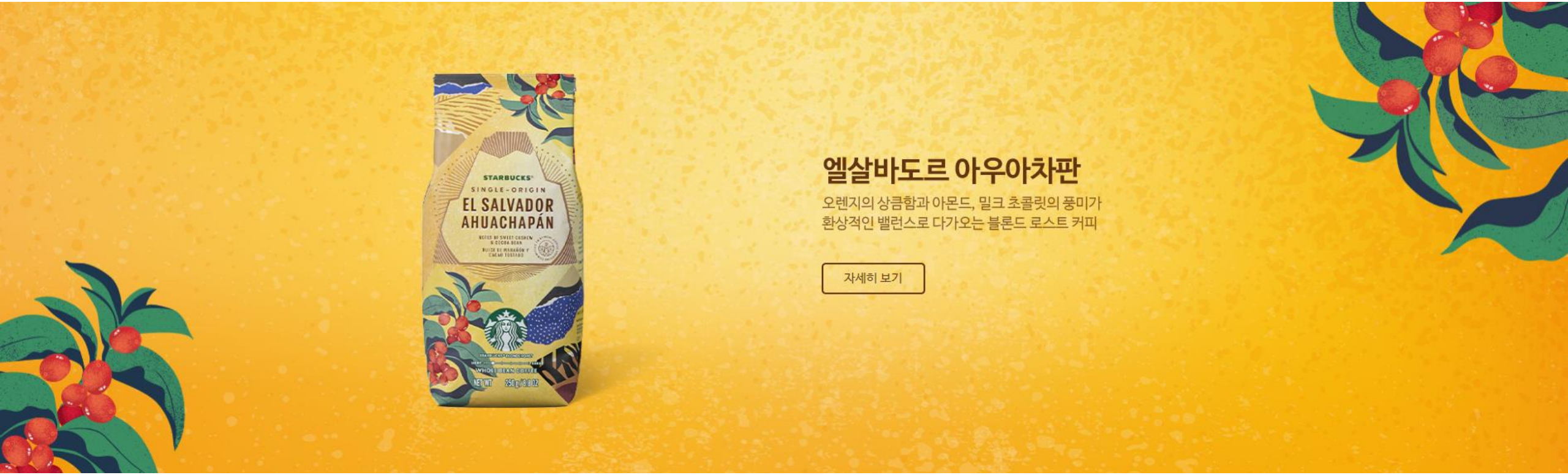
회원가입 | 로그인

회원 가입 후, 스타벅스 e-Gift Card를 "나에게 선물하기"로 구매하시고, 편리하게 등록하세요!
카드를 등록하여 스타벅스 리워드 회원이 되신 후, 첫 구매를 하시면 무료 음료 쿠폰을 드립니다!

e-Gift Card 선물하기


지금 시작합니다

애니메이션 비쥬얼


STARBUCKS
SINGLE-ORIGIN
EL SALVADOR
AHUACHAPÁN
엘살바도르 아우아차판
오렌지의 상큼함과 아몬드, 밀크 초콜릿의 풍미가
환상적인 밸런스로 다가오는 블론드 로스트 커피
자세히 보기

HTML

구성하기

```
<!-- ELSALVADOR -->
    <section class="elsalvador">
      <div class="inner">
        <img
          src="./images/elsalvador_item.png"
          alt="ELSALVADOR"
          class="elsalvador__item"
        />
        <div class="elsalvador__contents">
          <img src="./images/elsalvador_text.png" alt="" class="img_text" />
          <a href="#" class="btn btn--brown">자세히 보기</a>
        </div>
      </div>
    </section>
```

CSS

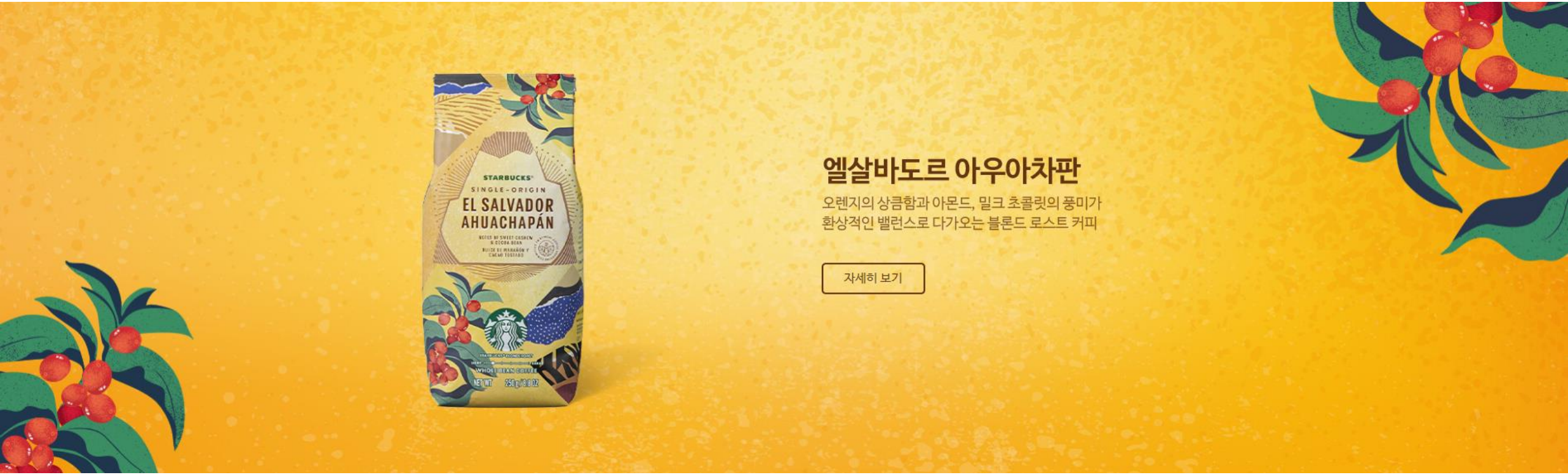
선언하기

Elsalvador, 백그라운드 지정!

- 백그라운드 지정
  - 이미지가 있으니 부르면 땡이겠죠?
  - 이미지 & 백그라운드 이미지 구별 방법
- 일단 기본 세팅
  - Height: 570px
  - Overflow: hidden;
  - Postion: relative;
  - Background-size: 105%;

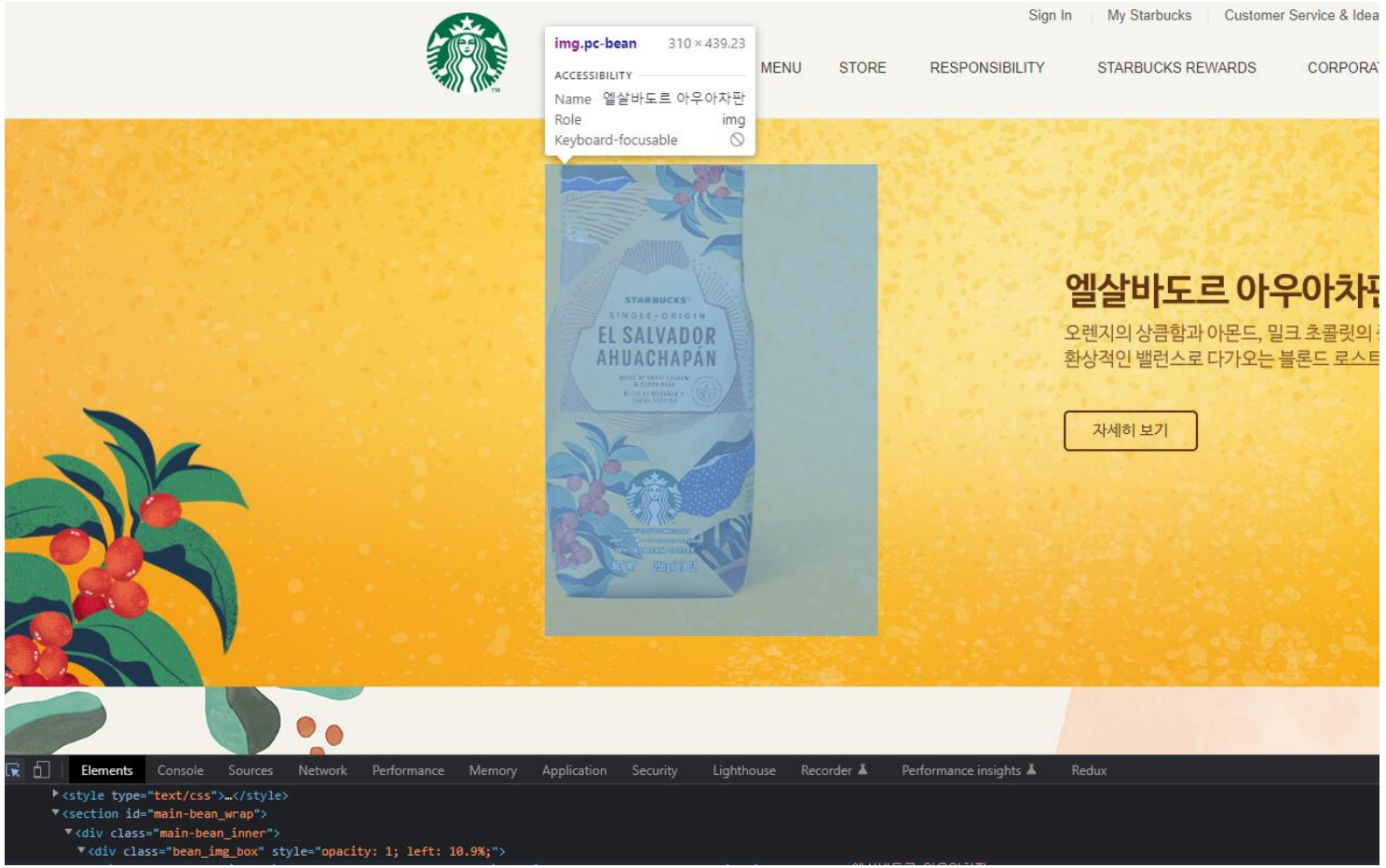
```
/* ELSALVADOR */
.elsalvador {
  position: relative;
  background-image:
URL("../images/elsalvador_bg.jpg");
  background-size: 105%;
  background-position: center;
  overflow: hidden;
  height: 570px;
}
```

실습, 이미지 위치 정하기!


STARBUCKS
SINGLE-ORIGIN
EL SALVADOR
AHUACHAPÁN
엘살바도르 아우아차판
오렌지의 상큼함과 아몬드, 밀크 초콜릿의 풍미가
환상적인 밸런스로 다가오는 블론드 로스트 커피
자세히 보기

실습, 이미지 위치 정하기!

- 스타벅스 페이지를 참고해서 이미지 크기 및 위치를 정해 주세요!




```css
.btn--brown {
  border-color: #633510;
  color: #633510;
  transition: 0.6s;
  cursor: pointer;
}

.btn--brown:hover {
  background-color: #633510;
  color: white;
  text-decoration: underline;
}
```

Elsalvador, 애니메이션 주기!

- transform: translate() 를 이용하여 애니메이션 효과 주기
- JS로 구현 하지만 일단 active 를 사용해서 구현해 봅시다!
- 좀 더 편하게 구현하는 방법!
  - 이미지를 밖에 두고 translate() 로 이동 시키는 방법 → X
  - 이미지를 원래 위치에 두고 translate() 로 밖으로 이동 시킨 다음 → active 시에 translate(0) 을 주기!

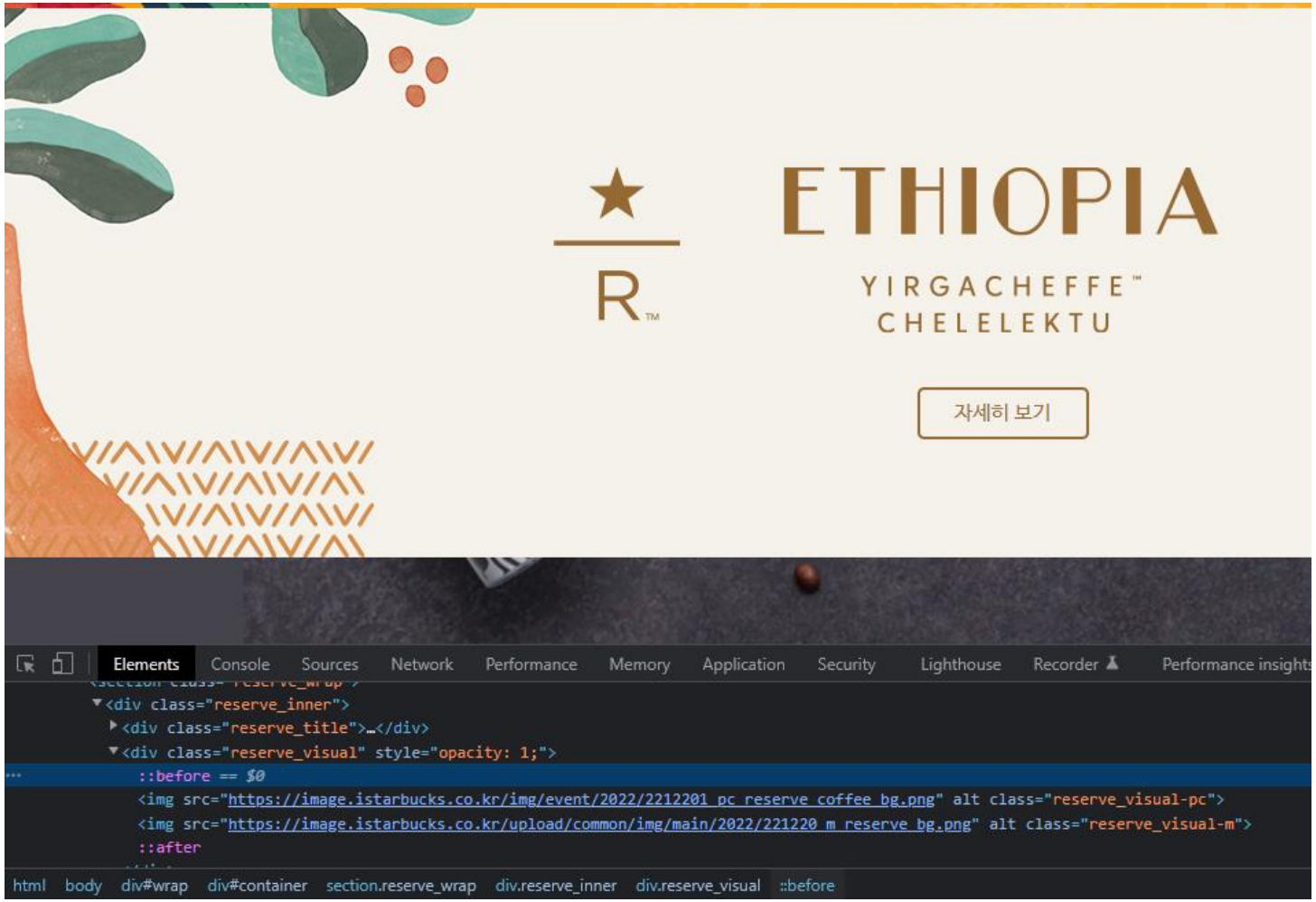
```css
.elsalvador .inner .elsalvador__item {
  transform: translate(-700px, 0);
  opacity: 0;
  transition: 2.5s;
}

.elsalvador:active .inner .elsalvador__item {
  transform: translate(0px, 0);
  opacity: 1;
 }

.elsalvador .inner .elsalvador__contents {
  transform: translate(900px, 0);
  transition: 2.5s;
}

.elsalvador:active .inner .elsalvador__contents {
  transform: translate(0px, 0);
}
```

실습, 이디오피아 구현하기

실습, 이디오피아 구현하기

- 양 옆의 요 이미지는 따로 넣어 주셔야 합니다!

실습, 이디오피아 구현하기




```css
.btn--gold {
  border-color: #966932;
  color: #966932;
  transition: 0.6s;
  cursor: pointer;
}

.btn--gold:hover {
  background-color: #966932;
  color: white;
  text-decoration: underline;
}
```

```
<!-- ETHIOPIA -->
<section class="ethiopia">
  <div class="inner">
    <img
      src="./images/ethiopia_text.png"
      alt="ETHIOPIA"
      class="ethiopia__contents"
    />
    <img
      src="./images/ethiopia_item.png"
      alt="ETHIOPIA"
      class="ethiopia__item"
    />
    <a href="#" class="btn btn--gold">자세히 보기</a>
  </div>
</section>
```

실습, Favorite 구현하기

실습, Favorite 구현하기

```css
.btn--white {
  border-color: #fff;
  color: #fff;
  transition: 0.6s;
  cursor: pointer;
}

.btn--white:hover {
  background-color: #fff;
  color: black;
  text-decoration: underline;
}
```

실습, Magazine 구현하기

RESERVE MAGAZINE

리저브 매거진과 함께 깊어가는 가을을 즐겨보세요.

자세히 보기

실습, Magazine 구현하기

```css
.btn--black {
  border-color: #000;
  color: #000;
  transition: 0.6s;
  cursor: pointer;
}

.btn--black:hover {
  background-color: #000;
  color: white;
  text-decoration: underline;
}
```

실습, FIND STORE 구현하기

FOOTER

FOOTER 구현하기

COMPANY

- 한눈에 보기
- 스타벅스 사명
- 스타벅스 소개
- 국내 뉴스룸
- 세계의 스타벅스
- 글로벌 뉴스룸

CORPORATE SALES

- 단체 및 기업 구매 안내
- 단체 주문 배달 안내

PARTNERSHIP

- 신규 입점 제의
- 협력 고객사 등록신청

ONLINE COMMUNITY

- 페이스북
- 트위터
- 유튜브
- 인스타그램

RECRUIT

- 채용 소개
- 채용 지원하기

STARBUCKS

개인정보처리방침 | 영상정보처리기기 운영관리 방침 | 홈페이지 이용약관 | 위치정보 이용약관 | 스타벅스 카드 이용약관 | 비회원 이용약관 | My DT Pass 서비스 이용약관 | 윤리경영 핫라인

찾아오시는 길 | 신규입점제의 | 사이트 맵

FOOTER

Menu

```
<footer>
  <div class="inner">
    <!-- FOOTER MENU -->
    <div class="footer__menu">
      <div class="left">
        <ul class="menu">
          <li>
            <ul>
              <a href="#">COMPANY</a>
              <li>한눈에 보기</li>
              <li>스타벅스 사명</li>
              <li>스타벅스 소개</li>
              <li>국내 뉴스룸</li>
              <li>세계의 스타벅스</li>
              <li>글로벌 뉴스룸</li>
            </ul>
          </li>
        </ul>
      </div>
    <div class="right">
      <img src="./images/footer_logo.png" alt="FOOTER LOGO" />
      </div>
    </div>
```

FOOTER, menu 구현하기

- Flex 를 사용해서 메뉴 파트(left)와 로고 파트(right) 나누기
- 메뉴 파트는 넓이를 90%, 로고 파트는 10%로 부여

```
/* FOOTER MENU */
footer .inner .footer__menu {
  display: flex;
}

footer .inner .footer__menu .left {
  width: 90%;
}

footer .inner .footer__menu .right {
  width: 10%;
}
```

실습, FOOTER 구현하기

COMPANY
- 한눈에 보기
- 스타벅스 사명
- 스타벅스 소개
- 국내 뉴스룸
- 세계의 스타벅스
- 글로벌 뉴스룸

CORPORATE SALES
- 단체 및 기업 구매 안내
- 단체 주문 배달 안내

PARTNERSHIP
- 신규 입점 제의
- 협력 고객사 등록신청

ONLINE COMMUNITY
- 페이스북
- 트위터
- 유튜브
- 인스타그램

RECRUIT
- 채용 소개
- 채용 지원하기

STARBUCKS

- .menu 의 li 크기를 20%으로 부여하여 left 90% 공간을 1/5로 분리하여 적용

FOOTER

Sub-Menu

실습, sub_menu 구현하기

개인정보처리방침 | 영상정보처리기기 운영관리 방침 | 홈페이지 이용약관 | 위치정보 이용약관 | 스타벅스 카드 이용약관 | 비회원 이용약관 | My DT Pass 서비스 이용약관 | 윤리경영 핫라인

```
<div class="footer__sub-menu">
  <ul>
    <li><a href="#">개인정보처리방침</a></li>
    <li class="contour">|</li>
    <li><a href="#">영상정보처리기기 운영관리 방침</a></li>
    <li class="contour">|</li>
    <li><a href="#">홈페이지 이용약관 </a></li>
    <li class="contour">|</li>
    <li><a href="#">위치정보 이용약관</a></li>
    <li class="contour">|</li>
    <li><a href="#">스타벅스 카드 이용약관</a></li>
    <li class="contour">|</li>
    <li><a href="#">비회원 이용약관</a></li>
    <li class="contour">|</li>
    <li><a href="#">My DT Pass 서비스 이용약관</a></li>
    <li class="contour">|</li>
    <li><a href="#">윤리경영 핫라인</a></li>
  </ul>
</div>
```

FOOTER

button

실습, button 영역 구현하기

찾아오시는 길 | 신규입점제의 | 사이트 맵

```
<!-- FOOTER BTN -->
        <div class="footer__btn">
          <a href="#" class="btn btn--white">찾아오시는 길</a>
          <a href="#" class="btn btn--white">신규입점제의</a>
          <a href="#" class="btn btn--white">사이트 맵</a>
        </div>
```

실습, copylight 영역 구현하기

Cloned by **tetz** from Starbucks Korea, © 2023 Starbucks Coffee Company. All Rights Reserved.

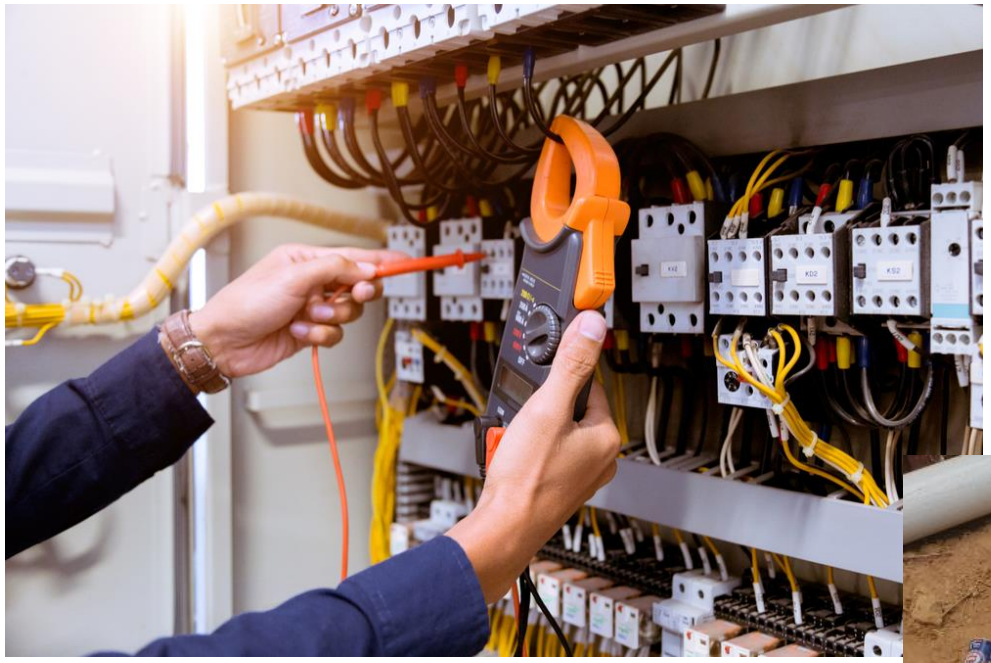
```
<!-- FOOTER COPYLIGHT -->
        <div class="footer__copylight">
          <p>
            Cloned by <span class="strong">tetz</span> from Starbucks Korea, ©
            2023 Starbucks Coffee Company. All Rights Reserved.
          </p>
        </div>
```


수고하셨습니다!


This is the moment 지금 의 순간

JS

javascript!



JS

WITHOUT

매일 쓰는 브라우저 보안이 걱정된다면, 안전하고 빠른 최신 브라우저 웨일로 업데이트 하세요. 다운로드 3일 동안 보지 않기

네이버를 시작페이지로 | 쥬니어네이버 해피빈

NAVER

메일 카페 블로그 지식iN 쇼핑 쇼핑LIVE Pay TV 사전 뉴스 증권 부동산 지도 VIBE 책 웹툰 더보기 32.3° 맑음 25.0° / 34.0° 북아현동

GRAND OPEN
2022년 7월
기다리다 지치겠어요

네이버를 더 안전하고 편리하게 이용하세요
NAVER 로그인
아이디 · 비밀번호찾기 회원가입

연합뉴스 > 15년 묵은 소득세, 전면 개편 검토...월급쟁이 세 부담... 뉴스홈 · 연예 스포츠 경제

이슈 코로나바이러스감염증-19 현황

뉴스스탠드 > 구독한 언론사 · 전체언론사

아이뉴스24	아시아경제	SBS	BLOTER	세계일보	한국일보
Chosun Biz	朝鮮日報	뉴스타파	YTN	시사iN	ZDNet Korea
OSEN	스포츠서울	디지털타임스	Abl	NEWSJOY	
Digital Today	NextDaily		RBS 한국농어촌방송	IT Chosun	보안뉴스

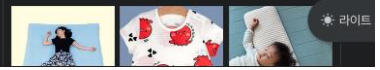
트렌드쇼핑 > 상품 쇼핑몰 MEN

쿠팡 · G마켓 · 옥션 · 11번가 · 티몬 · SSG닷컴
올리브영 · 위메프 · GS샵 · CJ온스타일 · 패션플러스

1 / 18

오늘 읽을만한 글 주제별로 분류된 다양한 글 모음 674 개의 글 관심주제 설정

엔터 스포츠 자동차 웹툰 경제 추천·구독 레시피 리빙



라이트

매일 쓰는 브라우저 보안이 걱정된다면, 안전하고 빠른 최신 브라우저 웨일로 업데이트 하세요. 다운로드 3일 동안 보지 않기

네이버를 시작페이지로 | 쥬니어네이버 해피빈

NAVER

메일 카페 블로그 지식iN 쇼핑 쇼핑LIVE Pay TV 사전 뉴스 증권 부동산 지도 VIBE 책 웹툰 더보기 32.3° 맑음 25.0° / 34.0° 북아현동

네이버를 더 안전하고 편리하게 이용하세요
NAVER 로그인
아이디 · 비밀번호찾기 회원가입

연합뉴스 > 15년 묵은 소득세, 전면 개편 검토...월급쟁이 세 부담... 뉴스홈 · 연예 스포츠 경제

이슈 코로나바이러스감염증-19 현황

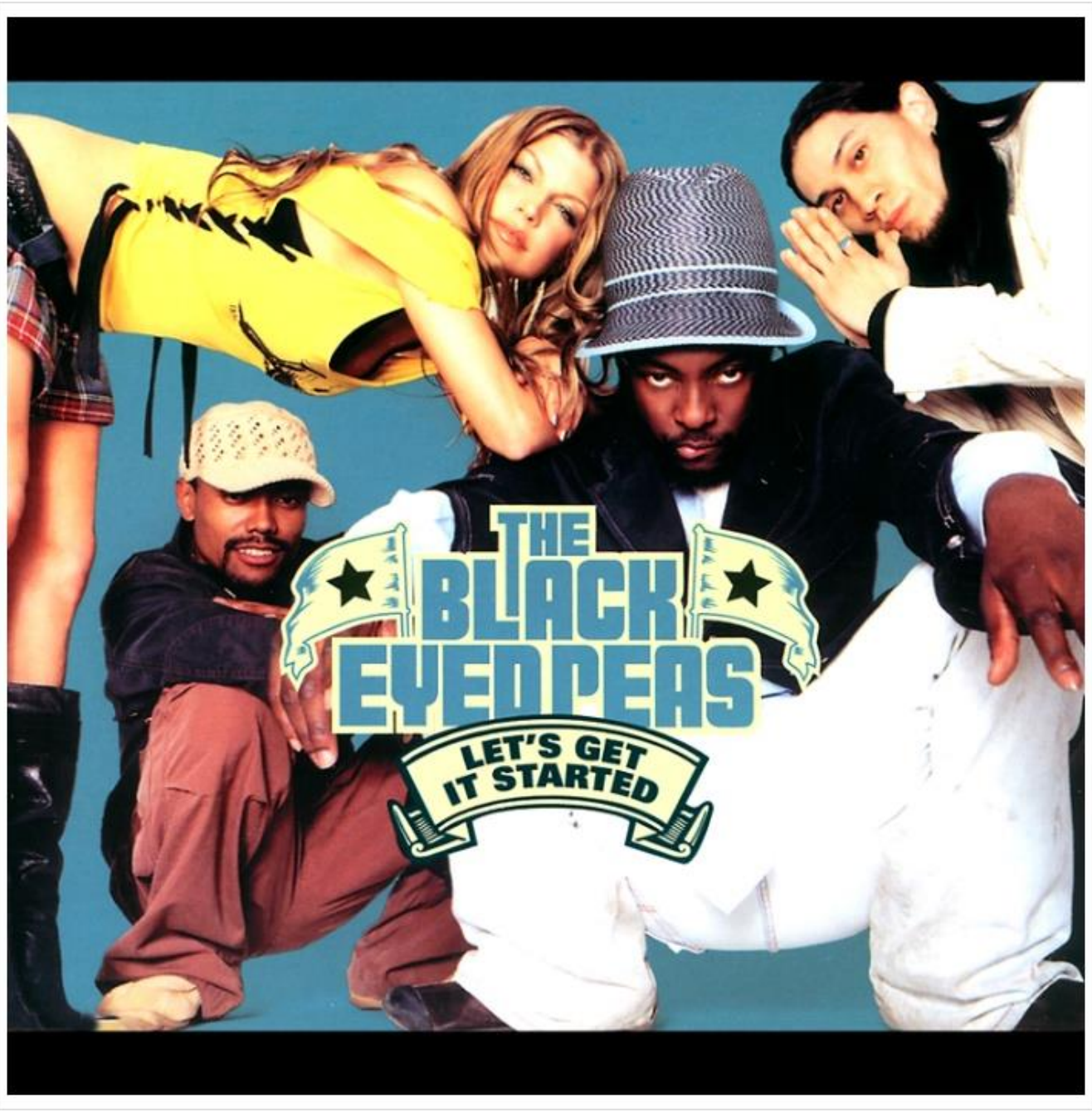
뉴스스탠드 > 구독한 언론사 · 전체언론사

한국일보	Korea JoongAng Daily	朝鮮日報	서울경제	MBN	SBS
The JoongAng	한겨레	전자신문	뉴스타파	MBC	스포츠서울
세계일보	시사iN	동아일보	소년한국일보	조이스경제	EBS
YONHAP NEWS AGENCY	디지털타임스	주간조선	ECONOMY Chosun	매경헬스	데일리한국

오늘 읽을만한 글 주제별로 분류된 다양한 글 모음 674 개의 글 관심주제 설정

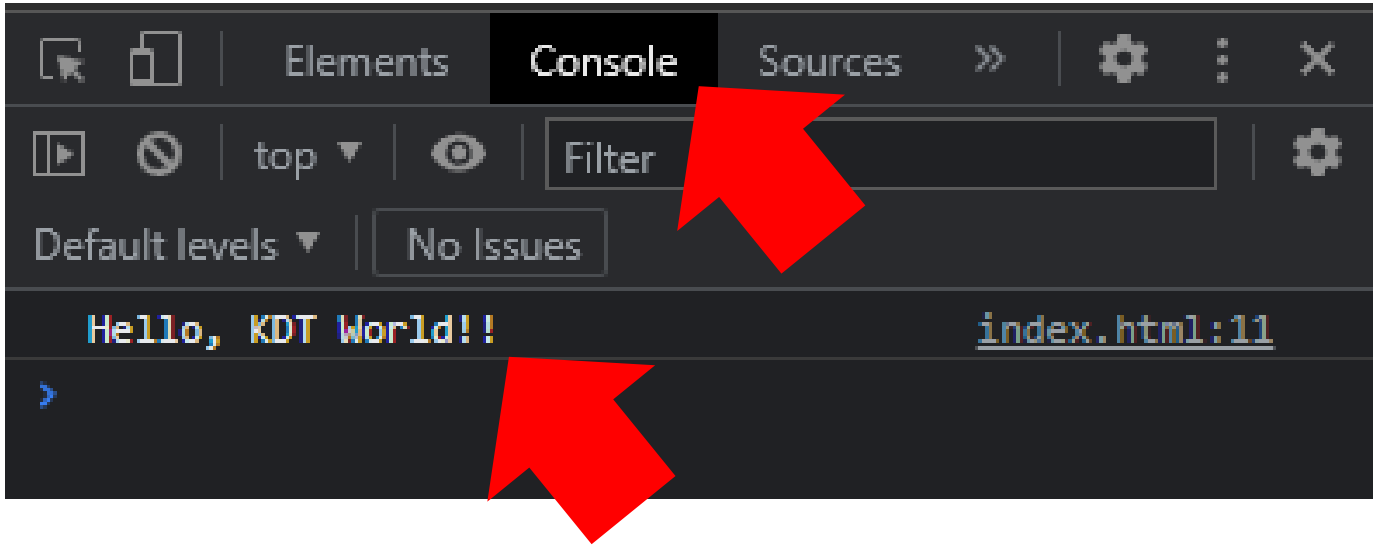
엔터 스포츠 자동차 웹툰 경제 추천·구독 레시피 리빙

라이트

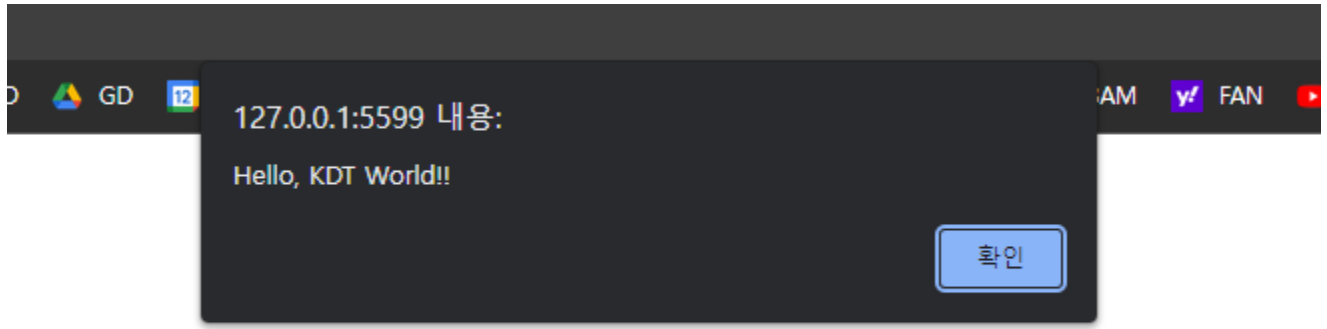

THE
BLACK
EYED PEAS
LET'S GET
IT STARTED

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <script>
      console.log("Hello, JS World!");
    </script>
  </head>
  <body></body>
</html>
```

개발자 도구 실행!


Elements
Console
Sources
top
Filter
Default levels
No Issues
Hello, KDT World!!
index.html:11

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <script>
      alert("Hello, JS World!");
    </script>
  </head>
  <body></body>
</html>
```


GD
AM
FAN
127.0.0.1:5599 내용:
Hello, KDT World!!
확인

3가지 CSS 참조 방식

인라인 방식 > 내장 방식 ? 링크 방식

JS

내장
방식

?

링크
방식

내장 방식!

```
<script>
        alert("Hello, JS World!");
</script>
```

- 위치는 어디서나 사용이 가능합니다!
  - Head 태그 내부
  - Body 태그 내부
  - Head 와 body 사이
  - Body 아래 등

링크 방식!

- Java Script 파일을 따로 만들어서 링크하는 방식 like CSS

```
<script src="./main.js"></script>
```

- 위치는 어디서나 사용이 가능합니다!
  - Head 태그 내부
  - Body 태그 내부
  - Head 와 body 사이
  - Body 아래 등

각각의 장단점이 존재 하겠죠?

- 내장 방식
  - 간단하게 만들 수 있음
  - 특정 페이지에서만 작동하는 기능일 경우 내장 방식으로만 따로 구현 가능
- 링크 방식
  - JS 코드의 양이 많아지면 파일로 관리하는 편이 편함
  - 같은 기능을 다른 페이지에서 사용하고 싶을 때 JS 파일 링크만 걸어서 사용 가능
  - 유지 보수 용이성이 편리

일단 해봅시다!

- Console.log("TEXT");
  - 우리의 디버깅 친구!
  - 물론 이걸 쓰는건 안 좋은 방식입니다!
- Alert("TEXT");
  - 화면에 뭐든 떠야 재미 있으니까!
  - 자매품 confirm("TEXT");
    - 얘는 취소 버튼이 있어요! → 값을 리턴한다 & IF문에 사용 가능!

실습

- 안녕하세요! Console.log() 로 띄우기
  - 단, 파일 링크 방식 사용
- 수업 언제 끝나냐!? Alert() 로 띄우기
  - 단, 내장 방식 사용

읽기 순서?

- CSS의 방식 또는 선언 위치에 따라서 읽기 순서가 달려졌어요?
- JS 에서도 확인해 봅시다!

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script>
        alert("헤드 그런데 js 파일 링크 위")
    </script>
    <script src="./index.js"></script>
    <script>
        alert("헤드 그런데 js 파일 링크 아래")
    </script>
</head>

<script>
    alert("헤드랑 바디 사이");
</script>

<body>
    hello, HTML World!
    <script>
        alert("바디 안쪽");
    </script>
</body>

<script>
    alert("바디 아래");
</script>


</html>
```

Index.html

Index.js

```js
alert("JS 파일 안쪽!");
```

읽기 순서!

- 말 그대로 읽히는 순서에 따라서 작동 합니다!!

JS 기초!

표기법

dash-case(kebab-case)

snake_case

camelCase

ParcelCase

thequickbrownfoxjumpsoverthelazydog

HTML CSS

dash-case(kebab-case)

the-quick-brown-fox-jumps-over-the-lazy-dog

HTML CSS

snake_case

the_quick_brown_fox_jumps_over_the_lazy_dog

JS

camelCase

theQuickBrownFoxJumpsOverTheLazyDog

JS

PascalCase

TheQuickBrownFoxJumpsOverTheLazyDog

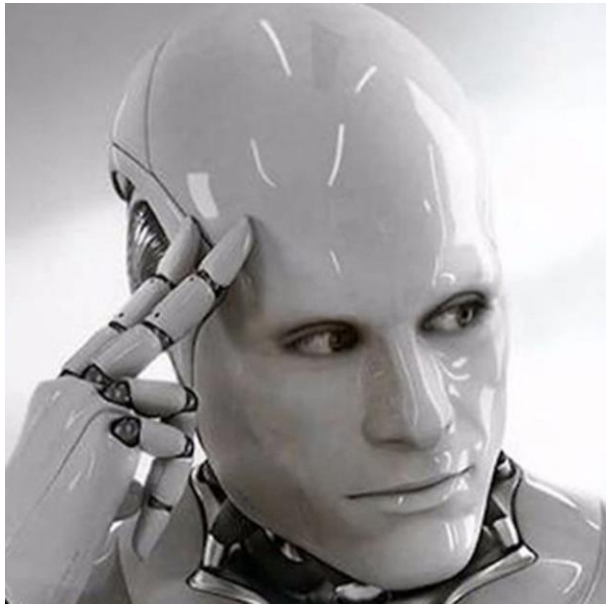
생성자!

Zero-based Numbering

0 기반 번호 매기기!

특수한 경우를 제외하고 0부터 숫자를 시작합니다.

12345678910~



0123456789~

```
let fruits = ['Apple', 'Banana', 'Cherry'];

console.log(fruits[0]); // 'Apple'
console.log(fruits[1]); // 'Banana'
console.log(fruits[2]); // 'Cherry'

console.log(new Date('2021-01-30').getDay()); // 6, 토요일
console.log(new Date('2021-01-31').getDay()); // 0, 일요일
console.log(new Date('2021-02-01').getDay()); // 1, 월요일
```

주석

Comments

```
// 한 줄 메모
/* 한 줄 메모 */

/**
 * 여러 줄
 * 메모1
 * 메모2
 */
```

단축키만 잘 외우시면 됩니다!

- 윈도우
  - Ctrl + /
  - Ctrl + K + C / Ctrl + K + U
- 맥
  - Cmd + /
  - Cmd + K + C / Cmd + K + U
- 사용하시기 편한 방법으로 쓰시면 됩니다!

JS 의

데이터 종류!

데이터 종류(자료형)

String

Number

Boolean

Undefined

Null

Object

Array

String

문자형 데이터

```
// String 문자형 데이터
// "" 따옴표를 사용

let myName = "TETZ";
var email = "xenosign@naver.com";
let hello = `Hello ${myName}?!`;

console.log(myName);
console.log(email);
console.log(hello);
```

```
TETZ                                    index.js:10
xenosign@naver.com                      index.js:11
Hello TETZ?!                            index.js:12
>
```

Number

숫자형 데이터

```
// Number 숫자형 데이터
// 정수 및 부동소수점 숫자를 나타냄
let number = 123;
let opacity = 0.7;

console.log(number);
console.log(opacity);
```

```
123                                                    index.js:17
0.7                                                    index.js:18
```

Boolean

참, 거짓 데이터

```
// Boolean 참, 거짓 데이터
// true, false 두 가지 값만 가지는 데이터
let checked = true;
let isShow = false;

console.log(checked);
console.log(isShow);
```

```
true                                    index.js:25
false                                   index.js:26
```

Undefined

미할당 데이터

```
// Undifined
// 값이 할당되지 않은 상태를 표기
let undef;
let obj = {
  abc: 123,
};

console.log(undef);
console.log(obj.abc);
console.log(obj.efg);
```

```
undefined                                    index.js:35
123                                          index.js:36
undefined                                    index.js:37
```

Null

의도된 빈 데이터

```
// Null
// 어떤 값이 "의도적"으로 비어 있음을 의미할 때 사용
let empty = null;

console.log(empty);
```

```
null                                                    index.js:43
```

Array

배열 데이터

```
// Array 배열 데이터
// 여러 데이터를 순차적으로 저장
let fruits = ["Pen", "Pineapple", "Apple", "Pen"];

console.log(fruits[0], fruits[1], fruits[2], fruits[3]);
```

```
Pen Pineapple Apple Pen
```

```
let data = [1, "Apple", false, null, undefined];

console.log(data[0], data[1], data[2], data[3], data[4]);
```

```
1 'Apple' false null undefined
```

Object

여러 데이터 꾸러미

```
// Object 객체형 데이터
// 여러 데이터를 key : value 의 형태로 저장

let tetz = {
  name: "이효석",
  age: "??",
  isOld: true,
  isMarried: false,
  hoby: ["Bicycle", "LOL", "Drink", "Travel"],
  brain: null,
  girlFriend: undefined,
};

console.log(tetz.name);
console.log(tetz.age);
console.log(tetz.isOld);
console.log(tetz.isMarried);
console.log(tetz.hoby);
console.log(tetz.brain);
console.log(tetz.girlFriend);
```

```
이효석
??
true
false
▶ (4) ['Bicycle', 'LOL', 'Drink', 'Travel']
null
undefined
```

실습

- 각자를 소개할 수 있는 Object 형태의 변수 선언
- 지금 까지 배운 모든 데이터를 사용하여 자신을 소개하는 console.log 만들기!

문자와

변수를 동시에!

문자 + 변수를 동시에 쓰고 싶을 때!

- 메소드의 매개 변수로 넣어서 사용
  - Console.log("문자", 변수, "문자);
- + 연산자를 사용해서 변수를 문자로 변환 후 더하여 사용
  - Console.log("문자" + 변수 + "문자);
- 백틱 문자 사용
  - `문자를 쓰다가 변수를 쓰고 싶으면 ${variable} 처럼 쓰면 됩니다`

변수!

(~~Variable~~)

변수

데이터를 저장하고 참조(사용)하는 **데이터의 이름**

var, let, const

변수, let

```
// 재사용이 가능!
// 변수 선언!
let a = 2;
let b = 5;

console.log(a + b); // 7
console.log(a - b); // -3
console.log(a * b); // 10
console.log(a / b); // 0.4
```

변수, let

```
// 값(데이터)의 재할당 가능!
let a = 12;
console.log(a); // 12

a = 999;
console.log(a); // 999
```

변수, const

```
// 값(데이터)의 재할당 불가!
const a = 12;
console.log(a); // 12

a = 999;
console.log(a); // TypeError: Assignment to constant variable.
```

예약어

특별한 의미를 가지고 있어, 변수나 함수 이름 등으로 사용할 수 없는 단어

Reserved Word

```
let this = 'Hello!'; // SyntaxError
let if = 123; // SyntaxError
let break = true; // SyntaxError
```

break, case, catch, continue, default, delete, do, else, false, fina
lly, for, function, if, in, instanceof, new, null, return, switch, this,
throw, true, try, typeof, var, void, while, with,
abstract, boolean, byte, char, class, const, debugger, double, e
num, export, extends, final, float, goto, implements, import, int,
interface, long, native, package, private, protected, public, shor
t, static, super, synchronized, throws, transient, volatile, as, is,
namespace, use, arguments, Array, Boolean, Date, decodeURI,
decodeURIComponent, encodeURI, Error, escape, eval,
EvalError, Function, Infinity, isFinite, isNaN, Math, NaN,
Number, Object, parseFloat, parseInt, RangeError,
ReferenceError, RegExp, String, SyntaxError, TypeError,
undefined, unescape, URIError ...

~~Variable~~

~~Variable~~

지맘대로

```
// var
var name = "tetz";
var name = "LHS";

console.log(name);
```

```
LHS
```

```
// let
let name = "tetz";
let name = "LHS";

console.log(name);
```

```
Uncaught SyntaxError: Identifier 'name' has index.js:78
already been declared (at index.js:78:5)
```

```
// const
const name = "tetz";
const name = "LHS";

console.log(name);
```

```
Uncaught SyntaxError: Identifier 'name' has index.js:78
already been declared (at index.js:78:5)
```


“이불 밖은 위험해”

```
// var
var tetz = "LHS";

if (tetz == "LHS") {
  var result = true;
} else {
  var result = false;
}

console.log(result);
```

```
true
```

```
// let
var tetz = "LHS";

if (tetz == "LHS") {
  let result = true;
} else {
  let result = false;
}

console.log(result);
```

```
▶Uncaught ReferenceError: result is not defined    index.js:91
    at index.js:91:13
>
```

var 의 문제점

- 중간의 같은 이름의 변수를 다시 선언해도 기존의 변수에 덮어 씌움
- 변수를 선언 했다는 건 분명이 다른 데이터를 넣으려는 것인데, 그것을 기존의 데이터에 덮어 씌우면!? → 문제 발생!
- 그리고 변수가 {블록 단위} 에서 끝나는 것이 아니라, 자기 맘대로 전역으로 돌아다니고 영향력을 행사함 → 의도치 않은 문제 발생!
- 게다가 Debug 도 무지하게 어려움!
- 따라서 ES6 문법 부터는 var 대신 let 사용을 권장

함수

Function!

함수

특정 동작(기능)을 수행하는 일부 코드의 **집합**(부분)

function

```
// 함수 선언
function helloFunc() {
  // 실행 코드
  console.log(1234);
}

// 함수 호출
helloFunc(); // 1234
```

```
function returnFunc() {
  return 123;
}

let a = returnFunc();

console.log(a); // 123
```

```
// 함수 선언!
function sum(a, b) { // a와 b는 매개변수(Parameters)
  return a + b;
}

// 재사용!
let a = sum(1, 2); // 1과 2는 인수(Arguments)
let b = sum(7, 12);
let c = sum(2, 4);

console.log(a, b, c); // 3, 19, 6
```

```
// 기명(이름이 있는) 함수
// 함수 선언!
function hello() {
  console.log('Hello~');
}

// 익명(이름이 없는) 함수
// 함수 표현!
let world = function () {
  console.log('World~');
}

// 함수 호출!
hello(); // Hello~
world(); // World~
```

```
// Object 데이터
const tetz = {
  // 데이터 파트
  name: "이효석",
  age: "??",
  isMarried: false,
  hoby: ["Bicycle", "LOL", "Drink", "Travel"],

  // 메소드(Method) 파트
  getName: function () {
    return this.name;
  },
  doesHeMarried: function () {
    return this.isMarried;
  },
};

let hisName = tetz.getName();

console.log(hisName);
console.log(tetz.doesHeMarried());
```

```
이효석
false
```

실습

- 두 수의 곱을 구하는 함수 만들고 출력 시키기!
- 실습 2의 Object 데이터의 값을 return / 출력하는 메소드를 2개 추가하기!

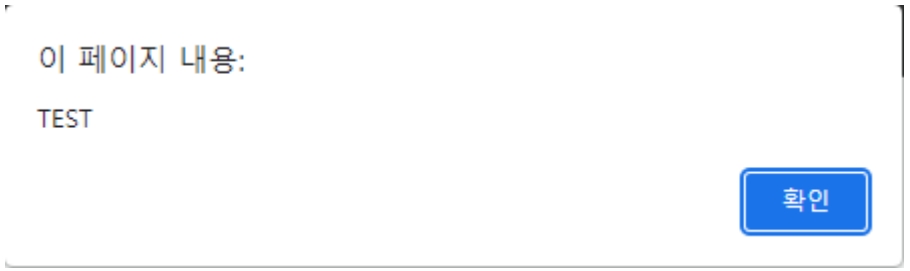
Onclick!

onclick

- 각각의 HTML 요소에 속성 값으로 JS 함수를 연결

```
<body>
    <div class="box"
onclick="test();">click</div>
</body>
```

```
function test() {
    alert("TEST");
}
```



실습

- 두 수의 곱을 구하는 함수 만들고 출력 시키기!
- 실습 2의 Object 데이터의 값을 return / 출력하는 메소드를 2개 추가하기!

조건문

IF!

비교 연산자!

비교연산자

a == b : a와 b가 동일하면 True

a != b : a와 b가 동일하지 않으면 True

a < b : a가 b보다 작으면 (b가 a보다 크면) True

a <= b : a가 b보다 작거나 같으면 True

a >= b : a가 b보다 크거나 같으면 True

논리연산자

a && b : a, b 전부가 true 일 때만 true

a || b : a, b 둘 중 하나만 true 면 true

IF / ELSE

```
if ( 조건1 ) {
        // 조건1이 참이라면 실행
} else {
        // 조건1dl  거짓이라면 실행
}
```

IF / ELSE IF / ELSE

```
if ( 조건1 ) {
        // 조건1이 참이라면 실행
} else if ( 조건2 ) {
        // 조건2가 참이라면 실행
} else {
        // 조건 1과 2가 모두 참이 아닐 때 실행
}
```

IF 중첩

```
if ( 조건1 ) {
        if ( 조건2 ) {
                //실행
        } else {
                //실행2
        }
}
```

```
let isShow = true;
let checked = false;

if (isShow) {
  console.log('Show!'); // Show!
}

if (checked) {
  console.log('Checked!');
}
```

```
let isShow = true;

if (isShow) {
  console.log('Show!');
} else {
  console.log('Hide?');
}
```

조건문

Switch

```
switch ( 변수 ) {
        case 값1:
                // 변수와 값1이 일치하면 실행
                break;
        case 값2:
                // 변수와 값2가 일치하면 실행
                break;
        default:
                //일치하는 값이 없을 때 실행
                break;
}
```

```
// Switch
let day;
switch (new Date().getDay()) {
  case 0:
    day = "일요일";
    break;
  case 1:
    day = "월요일";
    break;
  case 2:
    day = "화요일";
    break;
  case 3:
    day = "수요일";
    break;
  case 4:
    day = "목요일";
    break;
  case 5:
    day = "금요일";
    break;
  case 6:
    day = "토요일";
    break;
}

console.log(day);
```

```
월요일
>
```

3항 연산자

IF 문을 간단하게 표현하는 방법

- 조건식 ? 조건이 참인 경우 실행 : 조건이 거짓인 경우 실행;
- 한 줄로 간단히 표현 가능!

```
// 3항 연산자
let tetzName = "LHS";

if (tetzName == "LHS") {
  console.log("맞았어요 🤩");
} else {
  console.log("틀렸어요 😟");
}

tetzName != "LHS" ? console.log("맞았어요 🤩") : console.log("틀렸어요 😟");
```

실습

- If 문을 이용해서 오늘 요일을 alert 으로 출력하는 프로그램 작성
- 힌트
  - 오늘의 날짜 얻는 방법 : new Date().getDay(); 를 사용하면 오늘의 요일을 숫자로 받을 수 있다
  - 0 : 일요일 / 1 : 월요일 / 2 : 화요일 / 3 : 수요일 / 4 : 목요일 / 5 : 금요일 / 6 : 토요일

반복문

For, while

For 문

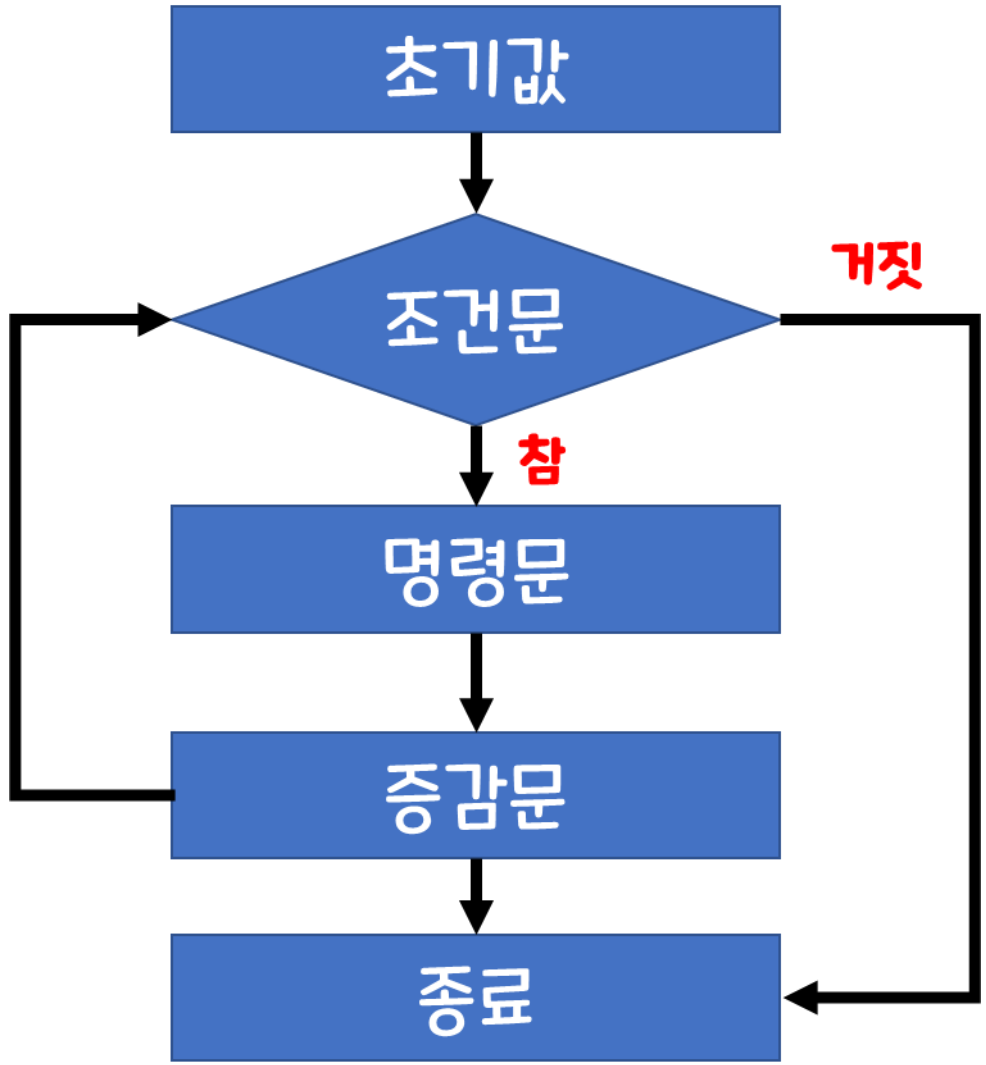
- for(초기 상태 설정; 조건문; 증감문) {

  실행할 코드

}

```
// for 문
for(let index = 0; index < 10; index++) {
  console.log("인사를 ", index+1, "번째 드립니다!
😆");
}
```

```
인사를 1 번째 드립니다! 😆
인사를 2 번째 드립니다! 😆
인사를 3 번째 드립니다! 😆
인사를 4 번째 드립니다! 😆
인사를 5 번째 드립니다! 😆
인사를 6 번째 드립니다! 😆
인사를 7 번째 드립니다! 😆
인사를 8 번째 드립니다! 😆
인사를 9 번째 드립니다! 😆
인사를 10 번째 드립니다! 😆
```


초기값
조건문
거짓
참
명령문
증감문
종료

while 문

- while(조건문) {

  실행할 코드(명령문)

}

- For 문과는 달리 조건을 변경하는 구문이 기본적으로 포함이 되어 있지 않기 때문에 무한 루프의 가능성이 농후!
- 주의하여 사용 필요

```
// while 문

// 1번 타입, 조건문을 사용
let index = 0;

while (index < 10) {
  console.log("인사를 ", index + 1, "번째 드립니다! 😆");
  index++;
}

// 2번 타입, 조건문을 사용하지 않고 if 문 + break 사용
let index2 = 0;

while (true) {
  console.log("절을 ", index2 + 1, "번째 드립니다! 😆");
  index2++;
  if (index2 == 10) {
    break;
  }
}
```

```
인사를 1 번째 드립니다! 😆
인사를 2 번째 드립니다! 😆
인사를 3 번째 드립니다! 😆
인사를 4 번째 드립니다! 😆
인사를 5 번째 드립니다! 😆
인사를 6 번째 드립니다! 😆
인사를 7 번째 드립니다! 😆
인사를 8 번째 드립니다! 😆
인사를 9 번째 드립니다! 😆
인사를 10 번째 드립니다! 😆
절을 1 번째 드립니다! 😆
절을 2 번째 드립니다! 😆
절을 3 번째 드립니다! 😆
절을 4 번째 드립니다! 😆
절을 5 번째 드립니다! 😆
절을 6 번째 드립니다! 😆
절을 7 번째 드립니다! 😆
절을 8 번째 드립니다! 😆
절을 9 번째 드립니다! 😆
절을 10 번째 드립니다! 😆
```

실습

- 구구단을 반복문을 이용해서 console.log 로 출력해 보자!
- 변수를 console.log 로 출력하는 방법

```
let number = 10;
console.log("숫자는", number, "입니다");
console.log("숫자는" + number + "입니다");
```

```
숫자는 10 입니다
숫자는10입니다
```

- "콤마"를 사용하면 각각의 변수의 데이터 타입을 지키면서 출력
- "+" 를 사용하면 숫자를 자동으로 문자로 변경 후, 문자열로 더하여 출력

실습

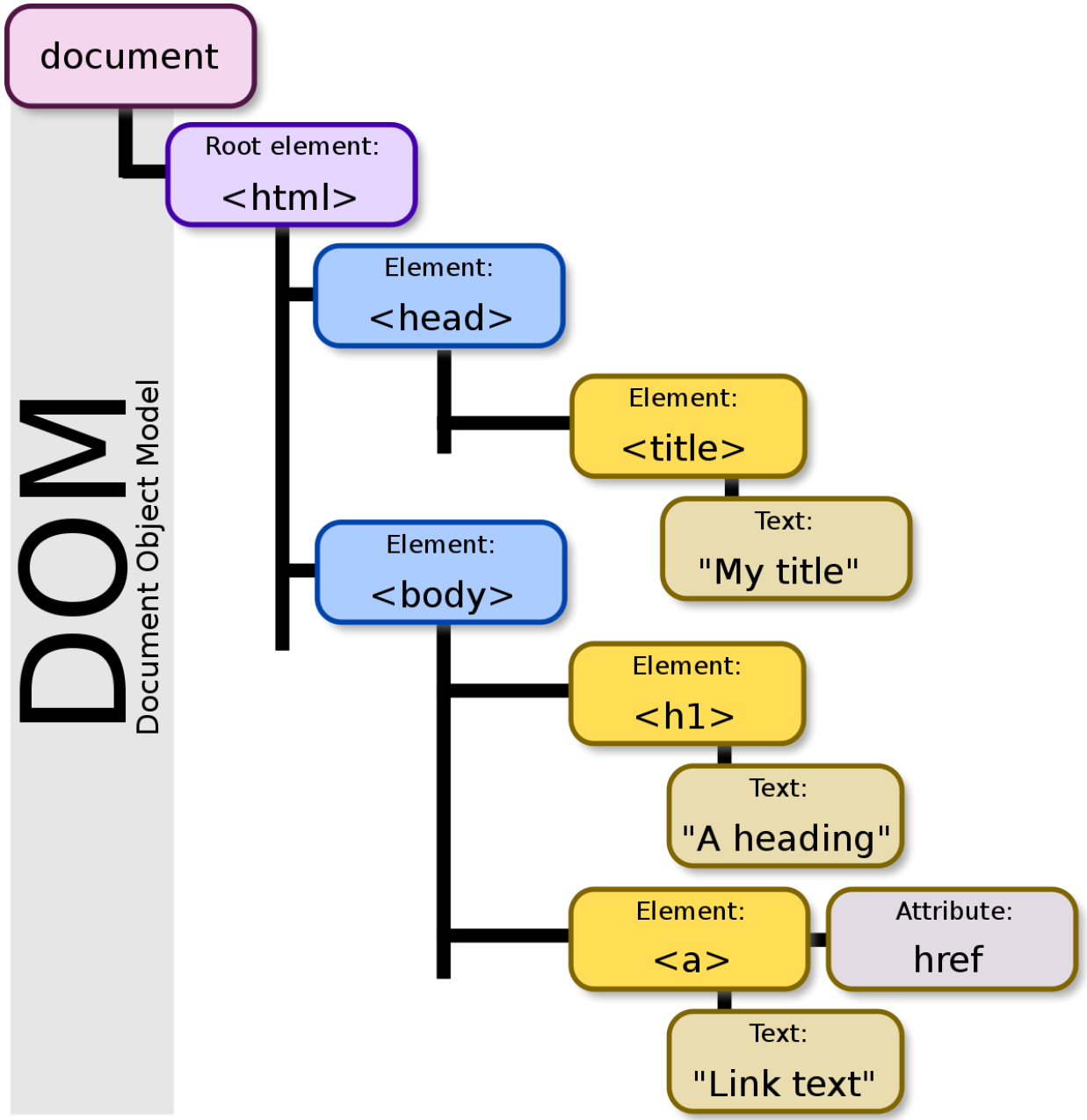
- 0 ~ 100의 숫자 중에서 2 또는 5의 배수 총합 구하기!
- 힌트
  - 나머지 연산자 % 를 사용
  - 5 % 3 → 2 (5를 3으로 나눈 나머지인 2의 값을 반환)

DOM

(Document Object Model)

DOM(Document Object Model)

- HTML 문서 요소의 집합!
- HTML 문서는 각각의 node 와 object 의 집합으로 문서를 표현
- 따라서 각각 node 또는 object 에 접근하여 문서 구조 / 스타일 / 내용 등을 변경 할 수 있도록 하는 것!


DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute:
href
Text:
"Link text"

DOM API

Document Object Model, Application Programming Interface

dom.html

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Document</title>
    <script src="./dom.js"></script>
</head>

<body>
    <div class="box">Box!!</div>
</body>

</html>
```

```
null
```

dom.js

```js
let boxEl = document.querySelector(".box");

console.log(boxEl);
```

dom.html

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Document</title>
</head>

<body>
    <div class="box">Box!!</div>
    <script src="./dom.js"></script>
</body>

</html>
```

```html
<div class="box">Box!!</div>
```

dom.js

```js
let boxEl = document.querySelector(".box");

console.log(boxEl);
```

dom.html

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Document</title>
    <script defer src="./dom.js"></script>
</head>

<body>
    <div class="box">Box!!</div>
</body>

</html>
```

```html
<div class="box">Box!!</div>
```

dom.js

```js
let boxEl = document.querySelector(".box");

console.log(boxEl);
```


불편한 진실
KBS2
왜 이러는 걸까요?

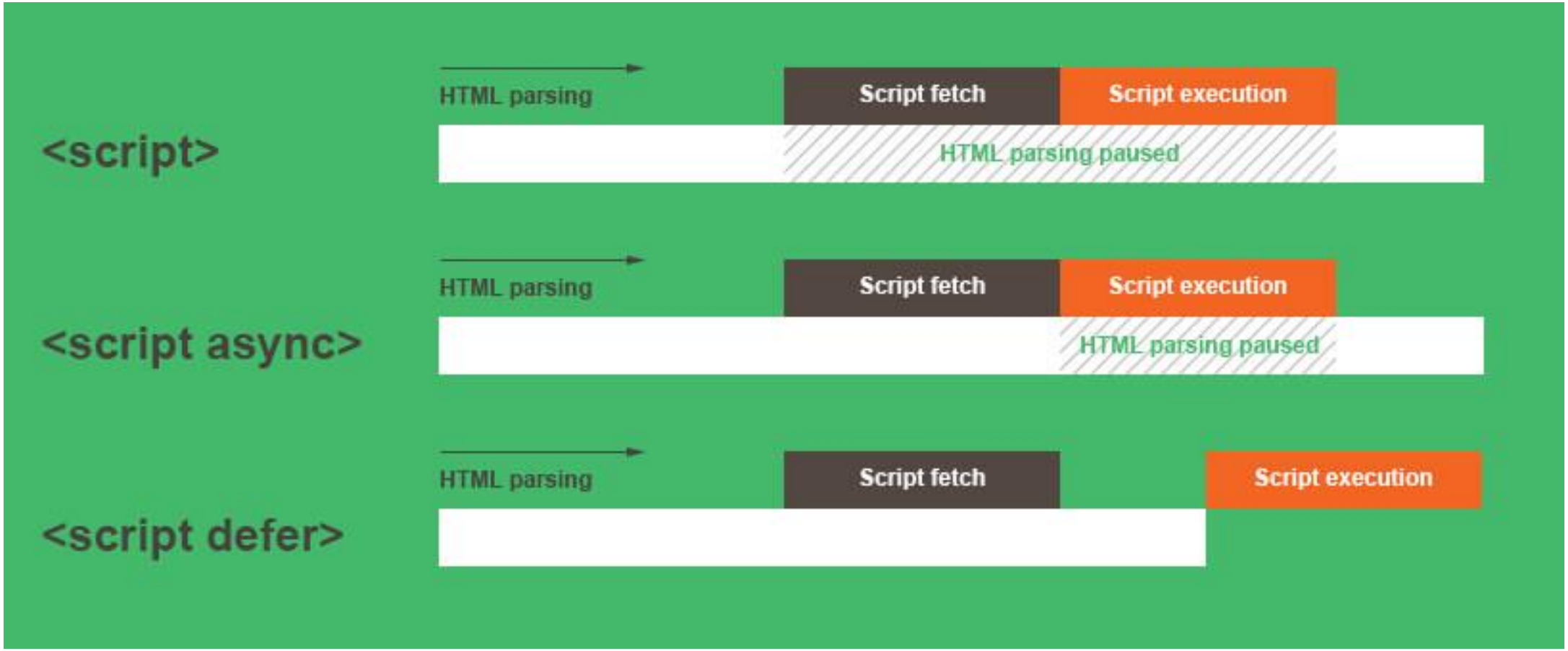
```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script src="./main.js"></script>
</head>
<body>
  <div class="box">Box!!</div>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="box">Box!!</div>
  <script src="./main.js"></script>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script defer src="./main.js"></script>
</head>
<body>
  <div class="box">Box!!</div>
</body>
</html>
```

Defer,

Async


<script>
HTML parsing
Script fetch
Script execution
HTML parsing paused
<script async>
HTML parsing
Script fetch
Script execution
HTML parsing paused
<script defer>
HTML parsing
Script fetch
Script execution

DOM API

Document Object Model, Application Programming Interface

```
// HTML 요소(Element) 1개 검색/찾기
const boxEl = document.querySelector('.box');

// HTML 요소에 적용할 수 있는 메소드!
boxEl.addEventListener();

// 인수(Arguments)를 추가 가능!
boxEl.addEventListener(1, 2);

// 1 - 이벤트(Event, 상황)
boxEl.addEventListener('click', 2);

// 2 - 핸들러(Handler, 실행할 함수)
boxEl.addEventListener('click', function () {
  console.log('Click~!');
});
```

querySelector("요소 선택자")

- 요소 선택자를 사용해서 자신이 가져오고 싶어하는 요소를 가져오는 메소드
- 문서에서 만나는 제일 첫번째 요소를 반환 합니다!

```
let boxEl = document.querySelector(".box");

console.log(boxEl);
```

addEventListener(이벤트, 명령)

- 선택 요소에 지정한 이벤트가 발생하면, 약속 된 명령어를 실행시키는 메소드

```
let boxEl = document.querySelector(".box");

console.log(boxEl);

boxEl.addEventListener("click", function() {
    alert("click!");
})
```

```
document.querySelector(".box").addEventListener("click", function() {
    alert("click");
})
```

```
// HTML 요소(Element) 검색/찾기
const boxEl = document.querySelector('.box');

// 요소의 클래스 정보 객체 활용!
boxEl.classList.add('active');
let isContains = boxEl.classList.contains('active');
console.log(isContains); // true

boxEl.classList.remove('active');
isContains = boxEl.classList.contains('active');
console.log(isContains); // false
```

classList.add / remove / contain

- 선택 요소에 class 를 더하거나, 빼거나, 클래스가 존재하는지 체크하는 메소드
- 해당 기능과 CSS 를 잘 활용하면 변화무쌍(?)한 웹페이지 구성이 가능

```
let boxEl = document.querySelector(".box");
console.log(boxEl);
console.log(boxEl.classList.contains("orange"));

boxEl.addEventListener("click", function() {
    boxEl.classList.add("orange");
    console.log(boxEl);
    console.log(boxEl.classList.contains("orange"));
})
```

```
    <div class="box">Box!!</div>
  false
    <div class="box orange">Box!!</div>
  true
```

실습

- .box 를 최초 클릭하면 배경을 orange 색으로 변경하기
- .box 를 다시 클릭 했을 때 배경이 orange 색이면 skyblue 로 변경 하거나 skyblue 면 orange 로 변경하는 페이지 만들기!

```
// HTML 요소(Element) 모두 검색/찾기
const boxEls = document.querySelectorAll('.box');
console.log(boxEls);

// 찾은 요소들 반복해서 함수 실행!
// 익명 함수를 인수로 추가!
boxEls.forEach(function () {});

// 첫 번째 매개변수(boxEl): 반복 중인 요소.
// 두 번째 매개변수(index): 반복 중인 번호
boxEls.forEach(function (boxEl, index) {});

// 출력!
boxEls.forEach(function (boxEl, index) {
  boxEl.classList.add(`order-${index + 1}`);
  console.log(index, boxEl);
});
```

querySelectorAll("요소 선택자")

- 문서에 존재하는 모든 요소를 찾아주는 메소드
- 모든 요소를 가져와서 배열(같은) 데이터로 만들어 줍니다!

```
<body>
    <div class="box">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
    <div class="box">4</div>
    <div class="box">5</div>
    <div class="box">6</div>
    <div class="box">7</div>
</body>
```

```
let boxEls = document.querySelectorAll(".box");
console.log(boxEls);
```

```
▼NodeList(7) [div.box, div.box, div.box, div.box, div.box, div.box, div.box]
  ▶0: div.box
  ▶1: div.box
  ▶2: div.box
  ▶3: div.box
  ▶4: div.box
  ▶5: div.box
  ▶6: div.box
    length: 7
  ▶[[Prototype]]: NodeList
```

forEach(function (반복중인 요소, 인덱스) {})

- 찾은 요소 전부에게 명령을 반복적으로 실행해 주는 메소드

```
let boxEls = document.querySelectorAll(".box");
console.log(boxEls);

boxEls.forEach(function (boxEl, index) {
    boxEl.classList.add(`box_${index + 1}`);
})

console.log(boxEls);
```

```
▼NodeList(7) [div.box, div.box,
  ▶0: div.box.box_1
  ▶1: div.box.box_2
  ▶2: div.box.box_3
  ▶3: div.box.box_4
  ▶4: div.box.box_5
  ▶5: div.box.box_6
  ▶6: div.box.box_7
    length: 7
  ▶[[Prototype]]: NodeList

▼NodeList(7) [div.box.box_1, di
, div.box.box_6, div.box.box_7
  ▶0: div.box.box_1
  ▶1: div.box.box_2
  ▶2: div.box.box_3
  ▶3: div.box.box_4
  ▶4: div.box.box_5
  ▶5: div.box.box_6
  ▶6: div.box.box_7
    length: 7
  ▶[[Prototype]]: NodeList
```

```
let boxEl = document.querySelector(".box");
console.log(boxEl.textContent);

boxEl.textContent = "KDT";
console.log(boxEl.textContent);
```

```
1
KDT
```

실습

- 최 상단에 버튼(아무 요소나 가능) 하나 만들기
- 100 개의 box 요소 만들기
- 버튼을 클릭하면 각각의 박스의 배경색이 변경 되는 페이지 만들기
- 첫번째 박스, 두번째 박스, 세번째 박스 ~ 열번째 박스 색상이 다르게 만들기
- 10개를 기준으로 색상이 반복 되도록 만들기


수고하셨습니다!