

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with





공지 사항



Swiper

The Most Modern Mobile Touch Slider


[Get Started](#) [API](#) [Element](#) [React](#) [Vue](#) [Demos](#)

MIT Licensed, v9.0.5 released on February 13, 2023 [Changelog](#)

 34,064 stars

Top Notch Features

- ✓ Library Agnostic
- ✓ Flexbox Layout
- ✓ Multi Row Slides Layout
- ✓ Two-way Control
- ✓ Rich API
- ✓ Parallax Transitions
- ✓ Virtual Slides
- ✓ Mutation Observer
- ✓ Full True RTL Support
- ✓ 3D Effects
- ✓ Full Navigation Control
- ✓ Most Flexible Slides Layout Grid
- ✓ Images Lazy Loading
- ✓ And many more

 Adobe Stock

Get 10 Free Images From
Adobe Stock. Start Now.

<https://swiperjs.com/>

Swiper 라이브러리 추가



```
<!-- Icons -->
<link rel="stylesheet"
      href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />

<!-- SWIPER -->
<link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-bundle.min.css" />
<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>

<!-- main -->
<link rel="stylesheet" href="./css/main.css">
<script defer src="./js/main.js"></script>
```



```
<div class="inner">
  <div class="inner__left">
    <h1>공지사항</h1>
    <div class="swiper">
      <div class="swiper-wrapper">
        <div class="swiper-slide">My DT Pass 이용 안내</div>
        <div class="swiper-slide">My DT Pass 관련 서비스 점검 안내</div>
        <div class="swiper-slide">시스템 개선 및 점검 안내</div>
        <div class="swiper-slide">전자 영수증 서비스 서비스 점검 안내</div>
        <div class="swiper-slide">스타벅스 e-Gift Card 구매 가능 금액 안내</div>
      </div>
    </div>
    <a href="#"><span class="material-symbols-outlined">add_circle</span></a>
  </div>
  <div class="inner__right">
    <h1>스타벅스 프로모션</h1>
    <a href="#"><span class="material-symbols-outlined">expand_circle_down</span></a>
  </div>
</div>
```

JS에 Swiper 생성자 함수 추가



- 변수 하나를 만들고 Swiper 생성자 함수 추가
- 설정 값 입력
 - Direction: "vertical"
 - Autoplay: true
 - Loop: true

```
// SWIPER
// SWIPER NOTICE
const swiperNotice = new
Swiper(".notice .inner .inner__left .swiper", {
  direction: "vertical",
  loop: true,
  autoplay: true,
});
```



프로모션

슬라이드



공지사항 시스템 개선 및 점검 안내



스타벅스 프로모션



스타벅스와 함께
하는 것을 즐기는 나만의 여름!

기간 : 2022년 6월 14일(화) - 7월 25일(월)

자세히 보기

STARBUCKS RESERVE™ SUMMER

스타벅스 리저브 서머 원두 & 커피

기간 : 2022년 4월 5일 ~ 2022년 8월 25일

자세히 보기



회원 계정에 등록된 스타벅스 카드
1만원당 별★1개를 즉시 추가로

기간 : 2022년 1월 1일 ~ 2022년 12월

자세히 보기



```
<div class="promotion">
  <div class="swiper">
    <div class="swiper-wrapper">
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
      <div class="swiper-slide">
        
        <a href="#" class="btn btn--black">자세히 보기</a>
      </div>
    </div>
  </div>
</div>
```



```
// SWIPER PROMOTION
const swiperPromotion = new Swiper(".promotion .swiper", {
  direction: "horizontal", // 기본 값
  slidesPerView: 3, // 한번에 보여줄 아이템 수
  spaceBetween: 10, // 아이템간 거리
  centeredSlides: true, // 슬라이드 센터 여부
  loop: true, // 루프 여부
  autoplay: {
    // 자동 재생, 변경 시간 설정
    delay: 1000,
    disableOnInteraction: false,
  },
});
```



```
.notice .promotion .swiper .swiper-wrapper {  
}  
  
.notice .promotion .swiper .swiper-wrapper .swiper-slide {  
  opacity: 0.5;  
  transition: 0.2s;  
}  
  
.notice .promotion .swiper .swiper-wrapper .swiper-slide-active {  
  opacity: 1;  
}  
  
.notice .promotion .swiper .swiper-wrapper .swiper-slide .btn {  
  width: 150px;  
  position: absolute;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  margin: auto;  
}
```

페이지네이션 / 이동 버튼 추가하기



- Swiper 에서 제공하는 pagination / prev / next 버튼 추가하기

```
<div class="swiper-pagination"></div>  
<div class="swiper-button-prev"></div>  
<div class="swiper-button-next"></div>
```

- Swiper 생성자에 옵션 추가하기!



```
// SWIPER PROMOTION
const swiperPromotion = new Swiper(".promotion .swiper", {
  direction: "horizontal", // 기본 값
  slidesPerView: 3, // 한번에 보여줄
  spaceBetween: 10, // 아이템간 거리
  centeredSlides: true, // 슬라이드 센터 여부
  loop: true, // 루프 여부
  autoplay: {
    // 자동 재생, 변경 시간 설정
    delay: 5000,
    disableOnInteraction: false,
  },
  pagination: {
    el: ".promotion .swiper-pagination", // pagination을 할 엘리먼트 클래스 설정
    clickable: true, // 클릭 가능 여부 설정
  },
  navigation: {
    prevEl: ".promotion .swiper-button-prev", // 이전 버튼 클래스 설정
    nextEl: ".promotion .swiper-button-next", // 이후 버튼 클래스 설정
  },
});
```

Autoplay?



- <a> 태그의 특성 상 href 로 인해 페이지를 새로 고침
- 이를 해결하기 위해 href="javascript:함수명();" 을 사용
- A 태그의 효과는 쓰고 싶지만, 페이지 새로 고침을 막으려면
 - 사용

```
<a href="javascript:controlAutoPlay();" ><span class="material-symbols-outlined"> check_circle </span></a>
```

Promotion Toggle



- Promotion section 토글 기능 추가!
- Toggle 용 버튼에 클래스 주기!
 - Javascript:void(0); 처리
- JS 에서 버튼에 addEventListener 추가
 - Promotion 에 hide 클래스 추가 / 삭제 처리
- CSS 에서 promotion 처리



```
<div class="inner_right">
  <h1>스타박스 프로모션</h1>
  <a href="javascript:controlAutoPlay();" class="toggle-promotion">
    <span class="material-symbols-outlined"> check_circle </span>
  </a>
</div>
```

HTML

```
.notice .promotion {
  position: relative;
  height: 658px;
  background-color: #f6f5ef;
  overflow: hidden;
  transition: height 0.4s;
}

.notice .promotion.hide {
  height: 0px;
}
```




```
const promotionEl = document.querySelector(".promotion");
const promotionToggleBtn = document.querySelector(".toggle-promotion");
promotionToggleBtn.addEventListener("click", function () {
  if (promotionEl.classList.contains("hide")) {
    promotionEl.classList.remove("hide");
    promotionToggleBtn.classList.add("show");
  } else {
    promotionEl.classList.add("hide");
    promotionToggleBtn.classList.remove("show");
  }
});
```

Javascript

```
.notice .inner__right .toggle-promotion {
  transition: 0.4s;
}
.notice .inner__right .toggle-promotion.show {
  transform: rotate(180deg);
}
```

CSS



스크롤에 따른
애니메이션 처리!



페이지가 로드되면 바로!

- Visual 의 애니메이션 재생 필요
- Visual 의 inner 클래스에 :active 로 처리 했던 것으로 클래스로 변경 → :active → animate
- 페이지가 로드 되면 바로 실행되는 함수
 - Window.onload() 사용

```
window.onload = () => {  
  const visualInner = document.querySelector(".visual .inner");  
  visualInner.classList.add("animate");  
};
```



엘살바도르 애니메이션!

- 스크롤 위치 찾기! → 300 정도가 적당해 보이네요!
- active 로 처리해 두었던 애니메이션을 animate 클래스로 변경!

```
if (scrollYpos > 300) {  
    const elsalvadorAnimate = document.querySelector(".elsalvador");  
    elsalvadorAnimate.classList.add("animate");  
}
```

JavaScript

```
.elsalvador.animate .inner .img_product {  
    transform: translate(0px, 0);  
    opacity: 1;  
    transition: 2.5s;  
}
```

CSS





코딩

테스트!



회원 가입부터 가시죠!

- 다양한 방법으로 회원 가입을 하시면 됩니다!
- https://programmers.co.kr/account/sign_in?referer=https%3A%2F%2Fprogrammers.co.kr%2F

다른 계정으로 로그인 하기



Facebook



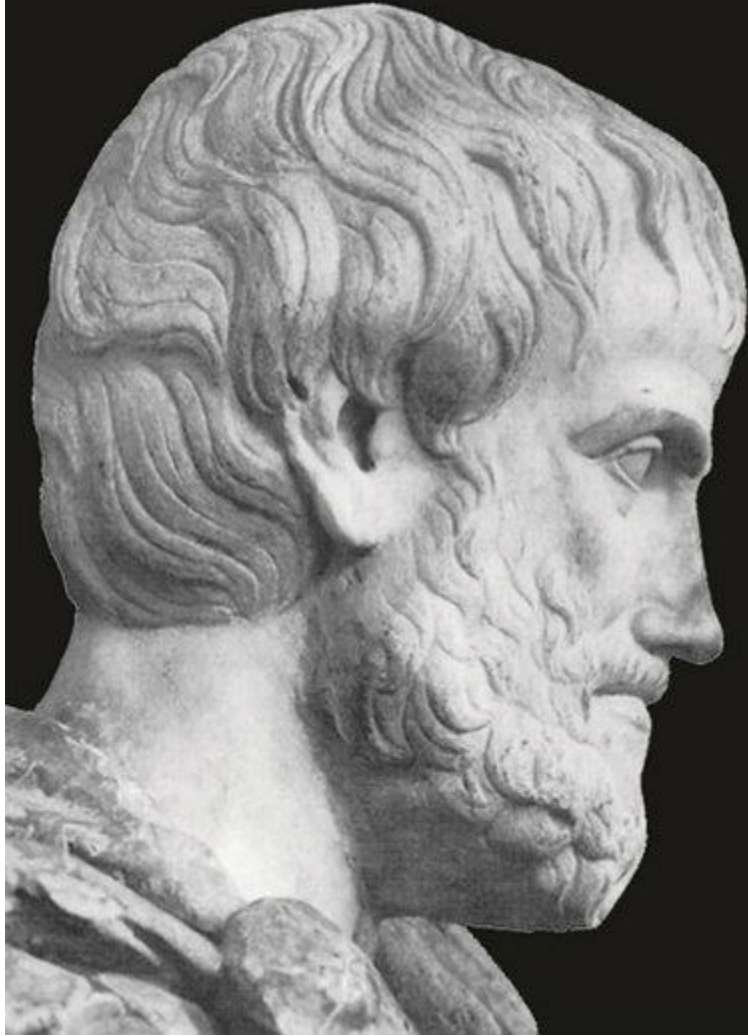
Github



Google



Kakao



Well Begun
is Half Done.

-Aristotle

프로그래머스 익히기!



- <https://school.programmers.co.kr/learn/courses/30/lessons/12937>
- <https://school.programmers.co.kr/learn/courses/30/lessons/120813?language=javascript>



지금 시작합니다



호텔 대실 연습문제	Lv. 2	1,099명	35%
무인도 여행 연습문제	Lv. 2	1,148명	35%
뒤에 있는 큰 수 찾기 연습문제	Lv. 2	1,100명	45%
숫자 변환하기 연습문제	Lv. 2	915명	44%
시소 짝꿍 연습문제	Lv. 2	807명	30%
인사고과 연습문제	Lv. 3	511명	29%
자동차 대여 기록 별 대여 금액 구하기 String, Date	Lv. 4	527명	28%
대여 횟수가 많은 자동차들의 월별 대여 횟수 구하기 GROUP BY	Lv. 3	1,000명	54%
자동차 대여 기록에서 장기/단기 대여 구분하기 String, Date	Lv. 1	1,675명	53%
자동차 종류 별 특정 옵션이 포함된 자동차 수 구하기 GROUP BY	Lv. 2	1,612명	77%
평균 일일 대여 요금 구하기 SELECT	Lv. 1	3,264명	72%
개인정보 수집 유효기간 2023 KAKAO BLIND RECRUITMENT	Lv. 1	3,761명	33%



문자 관련

JS 기능들!

문자열 길이, `.length`



- 문자열 `.length()`;

```
const str = "Hello, world";  
  
console.log(str.length);  
  
// 12
```

문자열 위치 찾기, indexOf



- 문자를 찾을 문자열.indexOf("찾을 문자열");

```
const str = "Hello, world";  
  
console.log(str.indexOf("world"));  
  
// 7
```

문자열 자르기, slice



- 문자열.slice(시작위치, 종료위치);

```
const str = "Hello, world";  
  
console.log(str.slice(0, 5));  
  
//Hello
```

문자열 바꾸기, **replace**



- 문자열.**replace**("찾을 문자" , "바꿀 문자");

```
const str = "Hello, world";  
  
console.log(str.replace("world", "뽀로로"));  
  
// Hello, 뽀로로
```


문자열 반복하기, repeat



- 문자열.repeat(반복 횟수);

```
let str = "우영";  
  
let longStr = str.repeat(10) + "우";  
  
console.log(longStr);
```

앞 뒤 공백 문자 제거하기, trim



- 문자열.trim();

```
const str = "      Hello, world      ";  
  
console.log(str.trim());  
  
// Hello, world
```

수박수박수박수박



- <https://school.programmers.co.kr/learn/courses/30/lessons/12922>



숫자 + 수학 관련

JS 기능들!

수학 함수들



```
console.log("abs: ", Math.abs(-999));      // 절대값

console.log("min: ", Math.min(10, 33));     // 작은 값
console.log("min: ", Math.min(10, 33, 6, 2, -1)); // 작은 값

console.log("max: ", Math.max(10, 33));     // 큰 값
console.log("max: ", Math.max(10, 33, 2, 1, 77)); // 큰 값

console.log("ceil: ", Math.ceil(3.14));     // 올림

console.log("floor: ", Math.floor(3.14));   // 버림

console.log("round: ", Math.round(3.6));    // 반올림
console.log("round: ", Math.round(3.4));    // 반올림

console.log("random: ", Math.random());     // 랜덤
```

실습



- 주어진 배열에서 가장 큰 수와 작은 수의 찾아서 소수점을 버림 처리
- 해당 수 절대값의 평균을 구하기

```
let nums = [-1.23, 13, 14.52, -33.53, 30];
```

- 0 ~ 100 까지의 숫자 중에서 랜덤한 정수가 나오도록 만들기

약수의 합



- <https://school.programmers.co.kr/learn/courses/30/lessons/12928>



배열 관련

기능들!



배열, `arr.map`

- 기존 배열에 특정 작업을 한 배열을 리턴하는 메소드
- `arr.map(function(배열의값, 인덱스, 원본배열) {});`

```
const fruits = ["사과", "파인애플", "수박", "포도", "아원지"];

let obj = fruits.map(function (item, index) {
  return {
    id: index,
    name: item,
  };
});

console.log(obj);
```

화살표 함수!



```
const fruits = ["사과", "파인애플", "수박", "포도", "아원지"];

let obj = fruits.map((item, index) => {
  return {
    id: index,
    name: item,
  }
})

console.log(obj);
```

- 요런 식으로 줄이기가 가능합니다!

화살표 함수!



// 1단계

```
const numbers = [1, 2, 3, 4, 5, 6];  
numbers.map(function (item) {  
  return item * 2;  
});
```

// 2단계

```
const numbers = [1, 2, 3, 4, 5, 6];  
numbers.map((item) => {  
  return item * 2;  
});
```

// 3단계

```
const numbers = [1, 2, 3, 4, 5, 6];  
numbers.map((item) => item * 2);
```

실습



- 1 부터 100 까지의 숫자가 들어 있는 배열을 for 문으로 만들기
- Map 메소드를 이용하여 해당 배열의 합산을 구하기!

문자열도 사실 배열이다!



- `apple = ["a", "p", "p", "l", "e"];`

```
let string = "apple";

for (let i = 0; i < string.length; i++) {
  console.log(string[i]);
}

for (letter of string) {
  console.log(letter);
}

for (index in string) {
  console.log(letter);
}
```



배열 함수, **filter**

- 조건에 부합하는 배열 요소만을 반환
- **arr.filter**(function (매개변수) { return 조건});

```
let numbers = [1, 2, 3, 4, 5, 6];  
let arr;  
  
arr = numbers.filter((num) => num > 3);  
  
console.log(arr);  
  
const words = ['spray', 'limit', 'elite', 'exuberant',  
  'destruction', 'present'];  
  
const result = words.filter(word => word.length > 6);  
  
console.log(result);
```



배열 함수, **includes**

- 해당 배열에 지정한 요소가 있는지 확인하는 메소드
- **arr.includes(요소);**

```
let numbers = [1, 2, 3, 4, 5, 6];

console.log(numbers.includes(3));
console.log(numbers.includes(7));

const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];

console.log(words.includes('elite'));
console.log(words.includes('pororo'));
```

실습



```
let fruits1 = ["사과", "딸기", "파인애플", "수박", "참외", "오렌지", "자두", "망고"];  
let fruits2 = ["수박", "사과", "참외", "오렌지", "파인애플", "망고"];
```

- 두 배열에서 동일한 요소만을 가지는 배열 sameArr 만들기
- 두 배열에서 서로 다른 요소만을 가지는 배열 diffArr 만들기



배열 함수, **find** / **findIndex**

- **find** : 배열에 특정 값이 있는지 찾고 반환
- **findIndex** : 배열에 특정 값이 있는지 찾고 위치를 반환

```
const fruits = ["Apple", "Banana", "Cherry"];

const result1 = fruits.find(item => {
  return /^A/.test(item);
})

const result2 = fruits.findIndex(item => {
  return /^C/.test(item);
})

console.log(result1);
console.log(result2);
```

배열, `arr.reduce`



- Reduce 는 가장 많은 기능을 할 수 있습니다!
- `arr.reduce`(function(누산기, 배열의값, 인덱스, 원본배열), 누산기 초기 값);



```
const arr = [1, 2, 3, 4, 5];  
const result = arr.reduce((acc, item, index) => {  
  return (acc += item);  
}, 0);  
console.log(result); // 15
```

```
const arr2 = [1, 2, 3, 4, 5];  
const result2 = arr2.reduce((acc, item, idx) => {  
  return (acc += item);  
}, 10);  
console.log(result2); // 25
```



```
const numbers = [2, -5, -123, 59, -5480, 24, 0, -69, 349, 3];
const result = numbers.reduce(
  (acc, item, index) => {
    if (item < 0) {
      // 처리할 현재 요소가 음수일 경우
      acc[0]++;
    } else if (item > 0) {
      // 처리할 현재 요소가 양수일 경우
      acc[1]++;
    }
    return acc;
  },
  [0, 0]
);
console.log(result); // [4, 5]
```

idx	cur	acc
0	2	[0, 1]
1	-5	[1, 1]
2	-123	[2, 1]
3	59	[2, 2]
4	-5480	[3, 2]
5	24	[3, 3]
6	0	[3, 3]
7	-69	[4, 3]
8	349	[4, 4]
9	3	[4, 5]

실습



- 1 부터 100 까지의 숫자가 들어 있는 배열을 for 문으로 만들기
- Reduce 메소드를 이용하여 해당 배열의 합산을 구하기! 단, reduce 메소드의 누산기를 이용하기

평균 구하기!



- <https://school.programmers.co.kr/learn/courses/30/lessons/12944>



Object

객체!

Object.assign()



- 객체의 병합에 사용되는 메소드

```
const obj1 = { a: 1, b: 2 };  
const obj2 = { b: 3, c: 4 };  
  
const returnObj = Object.assign(obj1, obj2);  
  
console.log(obj1);  
console.log(returnObj);
```

```
▶ {a: 1, b: 3, c: 4}  
▶ {a: 1, b: 3, c: 4}
```

- 기존에 선언된 객체에 사용하지 않고 가상의 Object 라는 것을 불러와서 사용 → 정적 메소드!



구조 분해 할당

(비구조화 할당)

구조 분해 할당



- 객체 또는 배열의 각각의 값을 분해하여 변수에 넣어 사용하는 방법

```
const user = {
  id: 1,
  name: "tetz",
  email: "xenosign@naver.com",
};

const { id, name, email, address } = user;
// 기본값 설정 const { id, name, email, address = "KOREA" } = user;
// 특정 변수에 넣기 const { id, name: tetz, email, address = "KOREA" } = user;

console.log(id);
console.log(name);
console.log(email);
console.log(address);

const fruits = ["사과", "딸기", "망고", "수박"];
const [a, b, c, d] = fruits;
console.log(a, b, c, d);
```



전개 연산자

전개 연산자



- 배열의 값을 , 단위로 구분 하여 전개시켜주는 연산자

```
const fruits = ["사과", "바나나", "수박"];
console.log(fruits);
console.log(...fruits);
// console.log("사과", "바나나", "수박");

function conLog(a, b, c) {
  console.log(a, b, c);
}

conLog(fruits[0], fruits[1], fruits[2]);
conLog(...fruits);
```



나머지 연산자(매개 변수일 때)

- 매개 변수에 너무 많은 값이 들어올 때 나머지 연산자를 사용하여 한꺼번에 처리 가능

```
const fruits = ["사과", "바나나", "수박", "망고", "딸기"];
```

```
function conLog(a, b, ...c) {  
  console.log(a, b, c);  
}
```

```
conLog(...fruits);
```

```
const fruits = ["사과", "바나나", "수박", "망고", "딸기"];
```

```
function conLog(...rest) {  
  rest.forEach((element) => {  
    console.log(element);  
  });  
}
```

```
conLog(...fruits);
```



문자열을 사실 진짜 배열로!

```
let string = "apple";  
let strToArr = [...string];  
console.log(strToArr);  
  
let strToArr2 = str.split("");  
console.log(strToArr2);
```

- 전개 연산자(...) 사용으로 "apple" 을 "a", "p", "p", "l", "e" 로 변환
- 그것을 [] 안에 넣어서 배열로 변환!

번호 가리기!



- <https://school.programmers.co.kr/learn/courses/30/lessons/12948>



자릿수 더하기



- <https://school.programmers.co.kr/learn/courses/30/lessons/12931>

모의 고사



- <https://school.programmers.co.kr/learn/courses/30/lessons/42840>

✓	모의고사 완전탐색	Lv. 1	56,847명	60%
---	--------------	-------	---------	-----

옹알이



- <https://school.programmers.co.kr/learn/courses/30/lessons/120956>

✓	옹알이 (1) 코딩테스트 입문	Lv. 0	8,006명	28%
---	---------------------	-------	--------	-----



<https://www.acmicpc.net/problem/1000>





넥슨 나이트 워커

열기

1000번

제출

맞힌 사람

숫코딩

재채점 결과

채점 현황

질문 게시판

A+B 다국어

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	780064	318713	223433	40.889%

문제

두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 A와 B가 주어진다. ($0 < A, B < 10$)



그런데 말입니다.

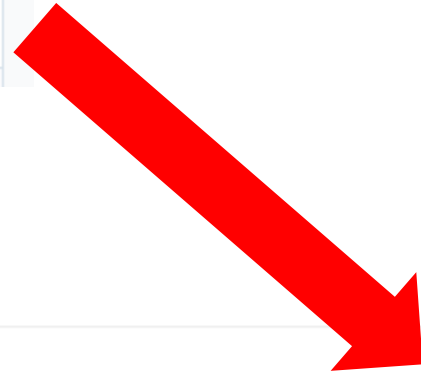


두 수의 합
코딩테스트 입문

Lv. 0

25,595명

92%



A+B 다국어

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	780064	318713	223433	40.889%



solution.js

```
1 function solution(num1, num2) {  
2     var answer = -1;  
3     return answer;  
4 }
```


A+B



언어

node.js ▼

[언어 설정](#)

소스 코드 공개

- ☒ 공개
☐ 비공개
☐ 맞았을 때만 공개

소스 코드

1

제출



응~ 어쩔티비~ 안궁 티비~
 뇌절티비~ 루뱅뱅
 지금 화났죠? 개 내가 사는
 곳 모르죠? 무것도 모르
 죠? 아무것도 알겠습니대
 무도 안물 인 극한 사 응 근데
 어쩔티비죠 어 안궁 zzz



```
const fs = require('fs');  
  
const input = fs.readFileSync('/dev/stdin').toString().split(' ');  
  
const A = parseInt(input[0], 10);  
const B = parseInt(input[1], 10);  
  
console.log(A + B);
```





예제 입력 1 복사

```
1 2
```

```
◀
```

```
≡ stdin ×
```

```
dev > ≡ stdin
```

```
1 1 2
```



```
stdin x
dev > stdin
1 1 2
```

```
JS test.js x stdin
JS test.js > ...
1 const fs = require("fs");
2
3 const input = fs.readFileSync("./dev/stdin").toString().split(" ");
4
5 const A = parseInt(input[0], 10);
6 const B = parseInt(input[1], 10);
7
8 console.log(A + B);
9
```



```
const fs = require("fs");  
  
const input = fs.readFileSync("./dev/stdin");  
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"  
<Buffer 31 20 32>
```



```
const fs = require("fs");  
  
const input =  
fs.readFileSync("./dev/stdin").toString();  
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"  
1 2
```



```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin").toString().split
(" ");
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test.js"
[ '1', '2' ]
```




```
const fs = require("fs");

const input = fs.readFileSync("./dev/stdin").toString().split(" ");
console.log(input);

const A = parseInt(input[0], 10);
const B = parseInt(input[1], 10);

console.log(A + B);
```





<https://www.acmicpc.net/problem/1037>

예제 입력 1 복사

```
2
4 2
```

예제 입력 2 복사

```
1
2
```

예제 입력 3 복사

```
6
3 4 2 12 6 8
```





```
stdin2 x JS test2.js
dev > stdin2
1 14
2 14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596
```

```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin2").toString().split("\n");
console.log(input);
```

```
[Running] node "c:\Users\lhs\Desktop\bj\test2.js"
[ '14\r', '14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596' ]
185192
```



New Line

CRLF / LF / CR

CR : Carriage Return (\r)

LF : Line Feed (\n)

해당 용어(CR, LF)와 이 용어들의 조합(CRLF)은 새로운 줄 (New line) 으로 바꾸는 방식을 의미한다.

CR 과 LF 는 타자기 시절 부터 줄바꿈을 위해 사용하던 방식인데 각각의 의미는 다음과 같다.

CR : 현재 커서를 줄 올림 없이 가장 앞으로 옮기는 동작

LF : 커서는 그 자리에 그대로 둔 상황에서 종이만 한 줄 올려 줄을 바꾸는 동작

대표적으로 Microsoft 사의 Windows 는 CRLF (\r\n) 을 기본으로 사용하는 반면

Unix/Linux 에서는 LF (\n) 만으로 줄바꿈을 하고 있다.



줄 2, 열 1 공백: 2 UTF-8 LF 일반 텍스트



```
[Running] node "c:\Users\lhs\Desktop\bj\test2.js"
```

```
[ '14', '14 26456 2 28 13228 3307 7 23149 8 6614 46298 56 4 92596' ]  
185192
```

```
const fs = require("fs");

const input =
fs.readFileSync("./dev/stdin2").toString().split("\n");
// 배열 값을 " " 로 나눈 배열로 변경
const arr1 = input[1].split(" ");
const arr2 = input[1].split(" ");

// 문자를 다시 10진법 숫자로 전환
const numArr1 = arr1.map((el) => parseInt(el, 10));
const numArr2 = arr2.map((el) => parseInt(el, 10));
```



그럼 풀어봅시다!!



1037번

제출


맞힌 사람

숫코딩

재채점 결과

채점 현황

내 제출

난이도 기여 

강의 ▾

질문 게시판

약수

언어

node.js ▾

[언어 설정](#)

소스 코드 공개

☒ 공개

☐ 비공개

☐ 맞았을 때만 공개

소스 코드

1

제출





영상 기반 랜딩 페이지

KDT

Team Tetz 1st

😊 Codign with fun

👥 Coding with us

2022 KDT Tetz Team 1st

영상 기반 랜딩 페이지

<https://movie-main-page.netlify.app/>



영상부터 구해봅시다

<https://www.pexels.com/ko-kr/>

<https://pixabay.com/videos/>



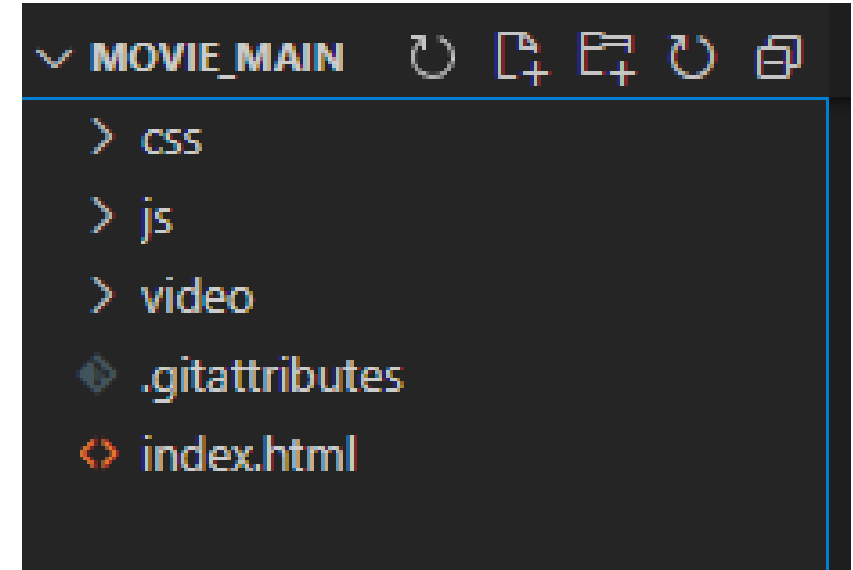
뼈대를 세우자

HTML



HTML 만들기 전에 세팅하기

- Index.html 문서 만들기
- Fontawesome 키트 추가하기
- JS 폴더 → 메뉴 처리를 위한 main.js 파일
추가 및 링크
- CSS 폴더 → style.css 링크
- Video 폴더 → 사용할 비디오 클립 삽입





기본 CSS 설정

기본 CSS 세팅



- 구글 폰트 가져오기
 - Architects Daughter
 - Fredericka the Great
- 마진, 패딩 X / 보더 박스 설정
- 리스트 스타일 X
- A 태그 속성 X



```
/* FONT */
@import
url("https://fonts.googleapis.com/css2?family=Architects+Daughter&family=Fredericka+the+Great&display=swap");

/* COMMON */
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}
ul,
ol {
  list-style: none;
}

a {
  text-decoration: none;
}

body {
  background-color: black;
}
```




HEADER

HEADER, HTML



- 로고 역할을 하는 H1
- GNB(Global Navigation Bar)
 - 메뉴는 마음대로
 - about / team / news / community
- SNS 파트
 - 여기도 마음대로
 - Search / Mail / Facebook
- 반응형에 사용할 햄버거 메뉴



```
<!-- HEADER -->
<header>
  <h1><a href="#">TetzKDT</a></h1>

  <ul id="gnb">
    <li><a href="#">ABOUT</a></li>
    <li><a href="#">TEAM</a></li>
    <li><a href="#">NEWS</a></li>
    <li><a href="#">COMMUNITY</a></li>
  </ul>

  <ul class="sns">
    <li><a href="#"><i class="fas fa-search"></i></a></li>
    <li><a href="#"><i class="far fa-envelope"></i></a></li>
    <li><a href="#"><i class="fab fa-facebook-square"></i></a></li>
  </ul>

  <a href="#" class="btn_menu"><i class="fas fa-bars"></i></a>
</header>
```

HEADER, CSS



- Header

- Width: 100% / border-bottom: 1px solid #999;

- 가로로 배치 해야겠죠?

- 각각 아이টে을 양끝에 붙여서 균등 배치 → justify-content: space-between;
 - 수직 중앙 정렬 → align-items: center;
 - 너무 양끝에 붙어 있으니 여백 주기 → padding : 5vh 5vw;



```
/* HEADER */  
header {  
  width: 100%;  
  border-bottom: 1px solid #999;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 5vh 5vw;  
}
```

HEADER, CSS



- 버거 메뉴는 반응형에서 보일 요소이므로 → `display: none;`
- 로고 폰트 설정
 - `font: bold 24px/1 "Architects Daughter";`
 - `color: #555;`

```
header .btn_menu {  
  display: none;  
}  
  
header h1 a {  
  font: bold 24px/50px "Architects Daughter";  
  color: #555;  
}
```



GNB

GNB, CSS



- 역시나 가로 배치 해야겠쥬? → `display: flex;`
- Li 태그
 - 각각 여백 주기 → `margin: 0px 20px;`
- A 태그
 - Font: 16px/1 "Architects Daughter";
 - Color: #555;
 - Letter-spacing: 1px;



```
header #gnb {  
  display: flex;  
}  
  
header #gnb li {  
  margin: 0px 20px;  
}  
  
header #gnb li a {  
  font: 16px/1 "Architects Daughter";  
  color: #333;  
  letter-spacing: 1px;  
}
```



SNS

실습, SNS 파트 작업하기!



- Gnb 파트를 참고하여 SNS 파트 구현!



```
header .sns {  
  display: flex;  
}  
  
header .sns li {  
  margin: 0px 10px;  
}  
  
header .sns li a {  
  font-size: 16px;  
  color: #333;  
}
```



Video Frame

Video Frame, HTML



- 멀티 미디어 콘텐츠의 경우 figure 태그를 사용
- Video 태그로 영상 삽입
 - 속성
 - Autoplay : 자동 재생 여부
 - Muted : 음소거 여부
 - Loop : 반복 여부
 - Controls : 컨트롤 패널 삽입 여부 (기본 속성 : X)
- 영상의 경우 muted 속성이 없으면 자동 재생이 안됩니다!



```
<!-- BACK VIDEO -->  
  <figure>  
    <video src="./video/video.mp4" autoplay muted loop></video>  
  </figure>
```

Video Frame, CSS



- **Figure**

- Width: 100%; / height: 100vh;
- position: absolute; / top: 0px / left: 0px;
- Overflow: hidden;
- Z-index: 1;

- **Video**

- Width: 100%;
- Height: 100%;
- Object-fit: cover; (background-size 와 동일)



```
/* VIDEO */
figure {
  width: 100%;
  height: 100vh;
  position: absolute;
  top: 0px;
  left: 0px;
  overflow: hidden;
  z-index: 1;
}

figure video {
  width: 100%;
  height: 100%;
  object-fit: cover;
}
```

Video Frame, Text HTML



- 특정 섹션의 내용이므로 article 사용, class → text
- P 태그로 텍스트 삽입

```
<figure>
  <video src="./video/video.mp4" autoplay muted loop></video>
  <article class="text">
    <p>KDT<br>Team Tetz 1st</p>
  </article>
</figure>
```

Video Frame, Text CSS



- 텍스트 영역 설정
 - Position: absolute;
 - Left: 6vw; / top: 50%; / translate(0, -50%);
- P 태그
 - Font: normal 8vw/1 "Fredericka the Great";
 - Color: rgb(218, 218, 218);



```
figure .text {  
  position: absolute;  
  left: 6vw;  
  top: 45%;  
  transform: translate(0, -50%);  
}  
  
figure .text p {  
  font: normal 8vw/1 "Fredericka the Great";  
  color: rgb(218, 218, 218);  
}
```

Video Frame, Text HTML



- 같은 요소가 2번 반복!? → list 사용

```
<ul>
  <li>
    <a href="#">
      <i class="fa-solid fa-face-laugh-wink"></i>
      <strong>Codign with fun</strong>
    </a>
  </li>
  <li>
    <a href="#">
      <i class="fa-solid fa-people-group"></i>
      <strong>Coding with us</strong>
    </a>
  </li>
</ul>
```

Video Frame, Text CSS



- UI
 - 가로 배치? Display → flex;
 - Margin-top: 30px;
- Li
 - Margin: 0px 30px;
- i 태그
 - Font-size: 20px / color: rgb(218, 218, 218);
 - Margin-right: 10px;
- Strong
 - Font: 14px/1 "Architects Daughter"; / color: rgb(218, 218, 218);
 - Letter-spacing: 1px;



```
figure .text ul {  
  display: flex;  
  margin-top: 30px;  
}  
figure .text ul li {  
  margin: 0px 30px;  
}  
  
figure .text ul li i {  
  font-size: 20px;  
  color: rgb(218, 218, 218);  
  margin-right: 10px;  
}  
  
figure .text ul li strong {  
  font: 14px/1 "Architects Daughter";  
  color: rgb(218, 218, 218);  
  letter-spacing: 1px;  
}
```



FOOTER

FOOTER, HTML



```
<!-- FOOTER -->  
<footer>  
    2022 KDT Tetz Team 1st  
</footer>
```

FOOTER, CSS



```
/* FOOTER */
footer {
  width: 100%;
  position: absolute;
  bottom: 0px;
  left: 0px;
  font: 12px/1 "arial";
  color: rgb(218, 218, 218);
  letter-spacing: 2px;
  text-align: center;
  padding: 5vh 5vw;
  z-index: 2;
}
```



반응형

작업하기

메뉴가 잘리는 영역 파악 or 데이터로!



- 개발자 도구를 사용하여 메뉴가 잘리기 시작하는 넓이 값을 바탕으로 반응형 작업 or 일반적 크기(태블릿, 핸드폰)에 맞춰서 작업
- 우리는 잘리기 시작하는 넓이로 설정!

```
/* MOBILE */  
@media screen and (max-width: 968px) {}
```

반응형 CSS 작업



- 안보이게 처리 할 것
 - Gnb
 - SNS
- 보이게 처리할 것
 - Btn_menu
- 반응형에서 텍스트 크기 키우기!



```
@media screen and (max-width: 968px) {  
  header #gnb {  
    display: none;  
  }  
  
  header .sns {  
    display: none;  
  }  
  
  header #snb {  
    display: block;  
  }  
  
  header .btn_menu {  
    display: block;  
  }  
  
figure .text p {  
  font-size: 15vw;  
}
```



Snb

추가하기

Snb, HTML



- Snb 영역을 따로 만들기 → div
- btn_menu 는 snb 안으로
- Gnb 의 메뉴를 그대로 옮길 것이므로 UL 사용
- GNB 의 메뉴 그대로 가져오기
- 마지막 Li 는 SNS 항목 추가



```
<div id="snb">
  <a href="#" class="btn_menu"><i class="fas fa-bars"></i></a>
  <ul class="sub_menu hide">
    <li><a href="#">ABOUT</a></li>
    <li><a href="#">TEAM</a></li>
    <li><a href="#">NEWS</a></li>
    <li><a href="#">COMMUNITY</a></li>
    <li>
      <a href="#"><i class="far fa-envelope"></i></a>
      <a href="#"><i class="fab fa-facebook-square"></i></a>
    </li>
  </ul>
</div>
```

Snb, CSS



- Sub_menu

- 문서 흐름에서 탈출 + 위치 내 맘대로 → `position: absolute;`
- 배경을 영상 배경과 비슷한 색으로 설정
- 오른쪽 여백은 햄버거 메뉴와 동일하게 → `right: 5vw;`
- 적당한 padding 주기
- 헤더의 border 와 동일한 border 주기
- Border-radius 취향대로 설정
- 글자 오른쪽 정렬!
- 버튼 클릭에 대한 반응을 `opacity` 로 할 것이므로 `opacity: 1;`
- `Transition: 0.4s;`



```
header #snb .sub_menu {  
  position: absolute;  
  padding: 10px 25px;  
  background-color: rgb(173, 179, 164);  
  border: 1px solid #999;  
  border-radius: 10px;  
  right: 5vw;  
  text-align: right;  
  opacity: 1;  
  transition: 0.4s;  
}  
  
header #snb .sub_menu.hide {  
  opacity: 0;  
}
```

Snb, CSS



- Sub_menu li
 - 적절한 상하 여백
 - 폰트 설정
- Sub_menu li a i
 - 여백 주기



```
header #snb .sub_menu li {  
  padding: 10px 0px;  
  font: 16px/1 "Architects Daughter";  
}
```

```
header #snb .sub_menu li a i {  
  margin-left: 8px;  
}
```



JS

기능 추가

Btn_menu 기능 추가



- Sub_menu의 hide class 존재 여부에 따라 클래스 추가 → 삭제 구현

```
const snbBtn = document.querySelector("#snb .btn_menu");
const subMenu = document.querySelector("#snb .sub_menu");

snbBtn.addEventListener("click", function () {
  if (subMenu.classList.contains("hide")) {
    subMenu.classList.remove("hide");
  } else {
    subMenu.classList.add("hide");
  }
});
```



과제!!



블로그!

- 블로그 최소 게시물 1개 업로드 하기!
- [포스코 x 코딩온] 제목 붙이기!
- 주차 회고 및 배운 내용 정리 내용 포함
- 일요일 10시까지 '반드시' 제출 하셔야 합니다!

영상 기반 랜딩 페이지 제작!



- 오늘 배운 영상 기반 랜딩 페이지를 제작하시면 됩니다!
- 최대한 예쁘게 만들어 주세요~!
- 깃에 올리시고, netlify 로 배포 까지 하신 다음 해당 링크를 일요일 밤 10시 까지 제출해 주세요!
- 다음주 월요일날 제출한 결과물을 다 같이 본 다음 → 투표!
- 1등, 2등에게 시상할 예정입니다! 😊

