

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 1) 다음 세 화면을 TailwindCSS를 이용하여 작성하시오.

색상, 스타일은 자유! (단, 위치 준수)

파일명: book-edit.html

BookMark

https://naver.com

☐ 이동 시 자동 삭제

↺

📄

There is no marks.

+ Add Book

"로그인 전 표현"

BookMark

Github Repo. | SignUp | SignIn

Sign Up

Nickname

nickname...

Email

email...

Password

password...

Password Confirm

password confirm...

Register

파일명: register.html

"로그인 후 표현"

BookMark

Github Repo. |

BookMark BookMar...

55BookMark B...

55나만의 북마크나만의...

네이버

네이버 메인에서 다양한...

Kakao

기술과 사람으로 더 나...

YouTube

YouTube에서 마음에 ...

+ Add Mark

Study

Tailwind CSS -...

Tailwind CSS is a ut...

Create Next A...

Generated by creat...

+ Add Mark

파일명: index.html

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 1) 추가 설명

서비스 로고(텍스트)

본인 github repository

본인의 프로필 이미지 자유사용

Bookmark 등록/수정

og:image, og:title, og:description

Bookmark 그룹 명

There is no marks.

채점 가이드

- Live Server로 테스트 했을 때 화면이 깨지거나 JS 오류가 없는가
- HTML 문법과 Tailwind의 올바른 사용
- 요구한 구성요소와 위치가 올바른가
- UI(스타일)에 얼마나 정성을 쏟았는가 (추가 1점)

© Hanaro 2024

Tailwind를 ex1 폴더에 설치 및 설정해서 작성하세요!

BookMark

Github Repo. SignUp | SignIn

https://naver.com

☐ 이동 시 자동 삭제

↺ ↻

BookMark BookMar...

55BookMark B...  
55나만의 북마크나만의...

네이버  
네이버 메인에서 다양한...

Kakao  
기술과 사람으로 더 나...

YouTube  
YouTube에서 마음에 ...

+ Add Mark

Study

Tailwind CSS -...  
Tailwind CSS is a ut...

^NEXT.js^ Create Next A...  
Generated by creat...

+ Add Mark

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 2) 정수 배열을 생성하는 range 함수를 작성하시오.

(요구 조건: 테스트 통과, 코드 가독성)

**\* 규칙 f(s, e, step)**

- step 기본값 =  $s > e ? -1 : 1$
  - $step === 0 \ || \ s === e ? [s]$
  - e 가 없다면,
    - ⇒  $s > 0 ? e = s, s = 1$
    - ⇒  $s < 0 ? e = -1$
    - ⇒  $s === 0 ? [0]$
  - 비정상(예외)
    - ⇒  $s > e \ \&\& \ step > 0 ? []$
    - ⇒  $s < e \ \&\& \ step < 0 ? []$
- 즉,  $(s - e) * step > 0$

```

range(1, 10, 1); // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
range(1, 10, 2); // [1, 3, 5, 7, 9]
range(1, 10);    // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
range(10, 1);    // [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
range(10, 1, -2); // [10, 8, 6, 4, 2]
range(5);        // [1, 2, 3, 4, 5]
range(100);      // [1, 2, 3, 4, 5, ..., 99, 100]
range(-5);       // [-5, -4, -3, -2, -1]
range(5, 5);     // [5]
range(5, 5, 0);  // [5]
range(5, 5, -1); // [5]
range(5, 1, 1);  // []
range(1, 5, -1); // []
range(1, 5, 6);  // [1]
range(0);        // [0]
range(2, 1, -5); // [2]
range(1, 5, 0); // [1]
range(0, 5);    // [0, 1, 2, 3, 4, 5]
range(0, -1);   // [0, -1]
range(0, -3);   // [0, -1, -2, -3]
range(-3, 0);   // [-3, -2, -1, 0]
range(5, 1);    // [5, 4, 3, 2, 1]
range(0, 0);    // [0]
range(0, 0, 5); // [0]
range(0, -1, -5); // [0]
    
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 3) sortBy 함수를 작성하시오.

(요구 조건: 테스트 통과, 코드 가독성)

```
Array.prototype.sortBy = function(...  
  
assert.deepEqual(users.sortBy('id'), [hong, kim, lee]);  
assert.deepEqual(users.sortBy('name:desc'), [lee, kim, hong]);  
assert.deepEqual(users.sortBy('dept:desc,city:asc'), [lee, kim, hong]);  
assert.deepEqual(users.sortBy('dept:desc,city:desc'), [kim, lee, hong]);  
assert.deepEqual(users.sortBy('dept:desc,id'), [kim, lee, hong]);
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 4) 깊은 복사를 하는 deepCopy 함수를 작성하시오.

(요구 조건: 테스트 통과, 코드 가독성)

```
const kim = {
  nid: 3,
  addr: 'Pusan',
  arr: [1, 2, 3, { aid: 1 }, [10, 20]],
  oo: { id: 1, name: 'Hong', addr: { city: 'Seoul' } },
  xx: null,
  yy: function() { console.log(this.oo); },
  yyy() { console.log(this.oo); },
  [Symbol()]: 9,
  [Symbol()]: Symbol('symbol2'),
  zs: new Set([arr, hong]),
  zws: new WeakSet([arr, hong]),
  zm: new Map([[hong, arr]]),
  zwm: new WeakMap([[hong, arr]]),
};
const newKim = deepCopy(kim);
assert.deepStrictEqual(newKim, kim, 'deepCopy equal fail!');
newKim.addr = 'Daegu';
newKim.oo.name = 'Kim';
assert.notDeepStrictEqual(newKim, kim, 'Not Valid Deep Copy!');
newKim.arr[0] = 100;
newKim.arr[3].aid = 200;
newKim.arr[4][1] = 300;
newKim.oo.addr.city = 'Daejeon';
assert.notStrictEqual(kim.arr[4][1], newKim.arr[4][1], 'pass2: 다르지 않아요!');
assert.notStrictEqual(kim.oo.addr.city, newKim.oo.addr.city, 'Not Pass3: city가 다르지 않아요!');
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 5) 초성 검색 함수를 정규식을 이용하여 작성하시오.  
 (요구 조건: 코드 가독성 및 간결함, 올바른 정규식 사용, 결과)

```
s = ['강원도 고성군', '고성군 토성면', '토성면 북면', '북면', '김1수']

assert.deepStrictEqual(searchByKoreanInitialSound(s, 'ㄱㅇ'), ['강원도 고성군']);
assert.deepStrictEqual(searchByKoreanInitialSound(s, 'ㄱㅅㄱ'), ['강원도 고성군', '고성군 토성면']);
assert.deepStrictEqual(searchByKoreanInitialSound(s, 'ㅌㅅㅁ'), ['고성군 토성면', '토성면 북면']);
assert.deepStrictEqual(searchByKoreanInitialSound(s, 'ㅃㅁ'), ['토성면 북면', '북면']);
assert.deepStrictEqual(searchByKoreanInitialSound(s, '표ㅁ'), []);
assert.deepStrictEqual(searchByKoreanInitialSound(s, 'ㄱ1ㅅ'), ['김1수']);
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 6) Promise.allSettled와 동일한 기능을 하는 promiseAllSettled 함수를 TS로 작성하시오.

만약 Promise.all 함수가 필요할 경우 promiseAll 함수도 직접 TS로 작성하여 사용하세요.

(요구조건: 테스트 통과, 올바른 Type 정의 및 TypeScript 사용)

```
const randTime = <T>(val: T): Promise<T> =>
  new Promise((resolve) => setTimeout(resolve, Math.random() * 1000, val));

assert.deepStrictEqual(
  await Promise.allSettled([randTime(1), randTime(2),randTime(3)]),
  await promiseAllSettled([randTime(1), randTime(2),randTime(3)])
);

assert.deepStrictEqual(
  await Promise.allSettled([randTime(11), Promise.reject('REJECT'), randTime(33)]),
  await promiseAllSettled([randTime(11), Promise.reject('REJECT'), randTime(33)])
);
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 7) 특정 `userId`의 댓글(`comments`)이 담긴 게시글 목록을 반환하는 `getPosts` 비동기 함수를 TypeScript로 작성하시오. (요구 조건: 테스트 통과 및 올바른 추가 테스트 코드 작성, 코드 가독성 및 간결함, 병렬로 실행되는 올바른 비동기 코딩)

- 예시) 1번 유저의 글목록: <https://jsonplaceholder.typicode.com/posts?userId=1>
- 댓글 목록: <https://jsonplaceholder.typicode.com/posts/<postId>/comments>

`getPosts(1)`를 호출하면 아래와 같은 **JSON Object** 반환

```
[
  {
    postId: 게시글ID,
    title: 게시글 제목,
    comments: [댓글 목록]
  },
  { ... }
]
```

```
const userId = 1;
const posts = await getPosts(userId);
...
assert.strictEqual(posts?.length, 10);
assert.strictEqual(posts?.at(-1)?.comments?.length, 5);
assert.deepStrictEqual(posts[0], {
  ...
  // 추가 테스트 코드를 작성하시오.
```

`postId`: 게시글ID,  
`id`: 댓글ID,  
`email`: 작성자 이메일 주소,  
`body`: 내용,



훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 8) 다음 debounce와 throttle 함수를 TypeScript로 작성하시오.

(요구조건: 범용성, 올바른 TypeScript 사용)

```
// test
const debo = debounce((a: number) => console.log(a + 1), 500);
for (let i = 10; i < 15; i++) debo(i);    // 15 출력

const thro = throttle((a: number) => console.log(a + 1), 500);
for (let i = 10; i < 15; i++) thro(i);    // 11 출력
```

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 9) 다음과 같이 작동하는 bill 함수를 클로저를 이용하여 작성하시오.

(요구 조건: 코드 가독성 및 간결함, 올바른 클로저/템플릿 태그 함수 작성, 결과화면)

참고) 부가세 = `Math.round((금액/1.1) * 0.1)` 또는 `Math.round(금액 / 11)`

```
const MENU = {
  짜장: { price: 7000 }, 짬뽕: { price: 9900 }, 탕숙: { price: 25000, taxfree: 1 },
};

function bill() {}

const table1 = bill();
table1.order('짜장');
table1.order('짬뽕');
table1.printBill();

const table2 = bill();
table2.order('짜장');
table2.printBill();

table1.order('탕숙');
table1.printBill();

table2.order('짬뽕');
table2.printBill();
```

**Table. 1**

---

\* 짜 장  
공 급 가 액 : 7,000원  
부 가 세 액 : 636원

---

\* 짬 뽕  
공 급 가 액 : 9,900원  
부 가 세 액 : 900원

---

주 문 합 계 : 16,900원  
주 문 합 계 : 1,536원

---

**Table. 1**

---

\* 짜 장  
공 급 가 액 : 7,000원  
부 가 세 액 : 636원

---

\* 짬 뽕  
공 급 가 액 : 9,900원  
부 가 세 액 : 900원

---

\* 탕 숙  
공 급 가 액 : 25,000원  
부 가 세 액 : 0원

---

주 문 합 계 : 41,900원  
주 문 합 계 : 1,536원

---

**Table. 2**

---

\* 짜 장  
공 급 가 액 : 7,000원  
부 가 세 액 : 636원

---

주 문 합 계 : 7,000원  
주 문 합 계 : 636원

---

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

문제 10) 다음 Collection을 상속받은 ArrayList class를 작성하고, 검증하는 테스트 코드를 TypeScript로 작성하시오.  
 (요구 조건: 올바른 테스트 코드 작성 및 통과, 올바른 타입스크립트 작성, 코드 가독성 및 간결함, 올바른 객체지향 코딩)

```

ArrayList.listToArray({ value: 1, rest: { value: 2 } }) ⇒ [1,2]
ArrayList.arrayToList([1,2]) ⇒ { value: 1, rest: { value: 2 } }

const alist = new ArrayList([1, 2]); // alist.toString() ⇒ { value: 1, rest: { value: 2 } }
alist.add(3); // { value: 1, rest: { value: 2, rest: { value: 3 } } }
alist.add(5, 1); // { value: 1, rest: { value: 5, rest: { value: 2, rest: { value: 3 } } } }
alist.remove(2); // { value: 1, rest: { value: 5, rest: { value: 3 } } }
alist.add(22, 1); // { value: 1, rest: { value: 22, rest: { value: 5, rest: { value: 3 } } } }
alist.add(33, 1);
alist.toString(); // ArrayList(4) { value: 1, rest: { value: 33, rest: { value: 22, rest: { value: 5, rest: ...} } } }
alist.set(1, 300); // { value: 1, rest: { value: 300, rest: { value: 22, rest: { ... } } } }
alist.get(2); alist.size; // 22, 5
alist.indexOf(300); // 1
alist.contains(300); alist.contains(301); // true, false
alist.isEmpty; alist.peek; // false, 3
alist.toArray(); // [1, 300, 22, 5, 3]
alist.iterator().next(); // { value: 1, done: false }
alist.clear(); // all clear
    
```

ArrayList Methods

- **add(value), add(value, index)**  
값 추가,            특정 index에 값 추가
- **get(index)** : 특정 index의 값 읽기
- **remove(value), removeByIndex(index)**
- **set(index, value)** : 변경
- **contains(value)**    cf. includes()
- **indexOf(value)**
- **size()**                    cf. length
- **iterator()**
- **toArray()** : values
- **isEmpty**
- **clear()**
- **print(), toString()**

훈련과정명	[하나은행] Digital hana 路 금융서비스개발 4기/5기	제출 기한	9월13일(금) 16:00까지
과 제 명	[개인 과제] 프론트엔드 개발 기초	제출 방법	학습사이트 → 과제 제출

## 채점 방법

1. 각 항목 당 10점 \* 10 = 100 점 만점
2. **각 항목 별 기준**에 맞춰 수(9~10), 우(7~8), 미(5~6), 양(3~4), 가(0~2) 로 등급 나눔
3. 테스트코드가 준비된 문제는 테스트 통과 시 일단 **수(9점)**에서 시작하여 점수 가감.
4. 해당 등급 수준으로 구현되었지만 코드 수준이나 미흡한 부분이 있다면 1점 감점  
예를들어, 우(8)를 받았지만 코드 수준 등 미흡한 부분이 보인다면 7점으로 결정.
5. 교과서(수업내용)를 원칙으로 하나 교과서 보다 더 우수하다면 등급 인정  
만약 구현은 잘 했지만 교과서와 다르면서 비 효율적이면 1점 감점.
6. 기본 기능 구현이 전혀 안됨(0), 구현하려고 시도(함수명 정도)는 했지만 내용 없음(1~2),  
구현했지만 **완전히** 틀림(3), 구현 했지만 **대부분** 틀림(4)
7. 그 외 수 ~ 미 등급은 항목 별 채점 가이드 준수하여 채점
8. 기타 애매한 부분은 강사가 학생에게 납득시킬 수 있는 근거를 피드백에 명확히 제시하고 점수 결정
9. 모든 테스트 코드와 기본 코드는 Github Repository에 기반하여 작성.