

[개인 과제] 프론트엔드 기초 4문제

훈련과정명	[하나은행] Digital hana 路 금융서비스개발
강의과목명	UI/UX 디자인과 구현(HTML/CSS), 프론트엔드 개발 기초 (JavaScript)
과 제 명	(개인-기초과제1) 프론트엔드 기초
제출 기한	02월14일(수) 23:00까지
제출 방법	학습사이트→ 과제 제출

1.html/css 를 이용하여 다음 테이블 구조를 완성하세요. (총 25점 - 각 5점)

- 1) 테이블 전체크기는 화면의 80%를 차지하고 중앙정렬 되어야 합니다.
 객실구성은 h1태그입니다.
- 2) table의 제목은 caption 태그를 사용합니다. TABLE태그에 속한 속성입니다
- 3) 패딩은 bottom과 top에 10px를 지정하고 텍스트 자체는 중앙정렬 하세요 .
- 4) 표의 타이틀이나 border 색상은 자유롭게 하셔도 됩니다.
 단, 동일한 색상으로 제목줄과 내용을 다르게 하셔야 합니다.
- 5) 전체적인 완성도를 봅니다.

객실구성

요안도라 객실 구성

방이름	대상	크기	가격
유채방	여성 도미토리	4인실	1인 20,000원
동백방	동성 도미토리		
	가족1팀		
천혜향방	-	2인실	
바깥채 전체를 렌트합니다			

과제 설명

[개인 과제] 프론트엔드 기초 4문제

훈련과정명	[하나은행] Digital hana 路 금융서비스개발
강의과목명	UI/UX 디자인과 구현(HTML/CSS), 프론트엔드 개발 기초 (JavaScript)
과 제 명	(개인-기초과제1) 프론트엔드 기초
제출 기한	02월14일(수) 23:00까지
제출 방법	학습사이트→ 과제 제출

2. json 객체 배열이 다음처럼 주어졌을때 지정한 연산들을 수행하세요.
페이지는 html 페이지에 작성하면 됩니다. (총 25점 - 각 5점)

```
{
  "base_time":20,
  "list":
  [
    {"id":1, "username":"홍길동", "work_time":40, "per_pay":10000},
    {"id":2, "username":"송민헌", "work_time":20, "per_pay":20000},
    {"id":3, "username":"장길산", "work_time":30, "per_pay":30000},
    {"id":4, "username":"임꺽정", "work_time":10, "per_pay":40000},
    {"id":4, "username":"강민구", "work_time":30, "per_pay":10000}
  ]
}
```

과제 설명

- 위 자료를 변수에 대입하고, 주급을 계산하여 테이블 태그에 채우기 바랍니다.
- 1) 테이블의 행과 셀은 동적으로 생성하세요,
window.onload 에서 변수에 값을 할당하고 함수를 작성하세요
 - 2) 주급 계산은 base_time에 전달된 시간을 기준으로 그 시간을 초과하는 부분에 대해서는 초과 수당을 지급하기 바랍니다. 초과수당은 per_pay의 50%입니다.
 - 3) 그리고 3.3% 세금을 공제하면 됩니다.
 - 4)숫자는 세자리마다 , 를 찍으시고 오른쪽 정렬을 해서 자리수를 맞춰주세요.
 - 5)총지불금액합계도 구하세요.

번호	이름	근무시간	시간당급여액	기본액	초과수당	합계	세액	실지불액
1	홍길동	40	10,000	400,000	100,000	500,000	16,500	483,500
2	송민헌	20	20,000	400,000	0	400,000	13,200	386,800
3	장길산	30	30,000	900,000	150,000	1,050,000	34,650	1,015,350
4	임꺽정	10	40,000	400,000	0	400,000	13,200	386,800
5	강민구	30	10,000	300,000	50,000	350,000	11,550	338,450
		총지불액	2,610,900					

[개인 과제] 프론트엔드 기초

훈련과정명	[하나은행] Digital hana 路 금융서비스개발
강의과목명	UI/UX 디자인과 구현(HTML/CSS), 프론트엔드 개발 기초 (JavaScript)
과 제 명	(개인-기초과제1) 프론트엔드 기초
제출 기한	02월14일(수) 23:00까지
제출 방법	학습사이트→ 과제 제출
과제 설명	<div>3. TypeScript - 객체 정의 (다음 오렌지 박스 부분을 파일 또는 소스로 제공하면 됨) (총 25점)</div> <div><pre>interface Naver { userid: number; username: string; email: string; } interface Kakao { userid: number; userName: string; kakaotalk: string; email: string; } interface SnsUser { <이 부분을 작성하시오!> email: string; } // 다음 코드에 오류가 없으면 통과! const naverUser: SnsUser = {userid: 1, username: 'HH', email: 'abc@naver.com'}; const kakaoUser: SnsUser = {userid: 1, userName: 'HH', kakaotalk: 'HH', email: 'abc@hanmail.net'};</pre></div>

[개인 과제] 프론트엔드 기초

훈련과정명	[하나은행] Digital hana 路 금융서비스개발
강의과목명	UI/UX 디자인과 구현(HTML/CSS), 프론트엔드 개발 기초 (JavaScript)
과 제 명	(개인-기초과제1) 프론트엔드 기초
제출 기한	02월14일(수) 23:00까지
제출 방법	학습사이트→ 과제 제출

과제 설명

4. JS - 객체지향 프로그래밍 (feat. Stack & Queue)
(다음 오렌지 박스 부분을 파일 또는 소스로 제공하면 됨) (총 25점)

```
import assert from 'assert'; // ESM
// → const assert = require('assert'); // CJS

class Collection {
  constructor() {}
  get _arr() {}
  push(value) {}

  // 가장 (Stack:나중, Queue:먼저) 들어간 요소 반환 (요소 삭제 없음!)
  get peek() {}

  // 가장 (Stack:나중, Queue:먼저) 들어간 요소 반환 & 삭제
  get poll() {}

  // 모든 원소 지우기
  clear() {}

  // array 타입 반환
  toArray() {}

  // 가장 (Stack:나중, Queue:먼저) 들어간 요소 삭제(skip)
  remove() {}

  // 원소가 하나도 없으면 true
  get isEmpty() {}

  // 현재 원소의 개수
  get size() {}
}

class Stack extends Collection {
  pop() { return this._arr.pop(); }
}

class Queue extends Collection {
  enqueue(value) { this.push(value); return this; }
  dequeue() { return this._arr.shift(); }
}

// 아래 코드가 통과되도록 Collection 클래스의 method를 작성하시오!
const stack = new Stack();
stack.push(1).push(2).push(3).push(5);
assert.deepStrictEqual(stack.toArray(), [1, 2, 3, 5]);
stack.pop();
assert.strictEqual(stack.peek, 3);
stack.remove();
assert.strictEqual(stack.poll, 2);
assert.deepStrictEqual(stack.toArray(), [1]);

const queue = new Queue();
queue.enqueue(1).enqueue(3).enqueue(5);
queue.dequeue();
assert.deepStrictEqual(queue.toArray(), [3, 5]);
assert.strictEqual(queue.poll, 3);
assert.deepStrictEqual(queue.toArray(), [5]);

if (!stack.isEmpty) stack.clear();
if (queue.size) queue.clear();
assert.deepStrictEqual(stack.toArray(), []);
assert.deepStrictEqual(queue.toArray(), []);
```