

2024년 상반기 K-디지털 트레이닝

VueJS Composition

[KB] IT's Your Life

Q1 - Composition API

- 프로젝트를 생성하고, **Option API**를 **Composition API**로 변환하시오.
- **name**, **cal**을 재사용 가능한 함수로 분리하여 **setup()** 함수내에서 참조하시오.
- **style**을 설정하시오.
 - 외부파일 또는 내부 파일
 - bs5설치 후 사용

```
<script>
export default {
  name : "OptionAPI",
  data(){
    return {
      name : "",
      x: 0,
      y: 0
    }
  },
  computed : {
    result() {
      return parseInt(this.x, 10) + parseInt(this.y, 10)
    }
  },
  mounted() {
    this.name = "john",
    this.x = 10,
    this.y = 20;
  },
  methods : {
    changeX(strX) {
      let x = parseInt(strX, 10);
      this.x = isNaN(x) ? 0 : x;
    },
    changeY(strY) {
      let y = parseInt(strY, 10);
      this.y = isNaN(y) ? 0 : y;
    },
    changeName(name) {
      this.name = name.trim().length < 2 ? "" : name.trim();
    }
  }
}
```



name : john

x : 10

y : 20

name:

honggildong

x:

109

y:

20

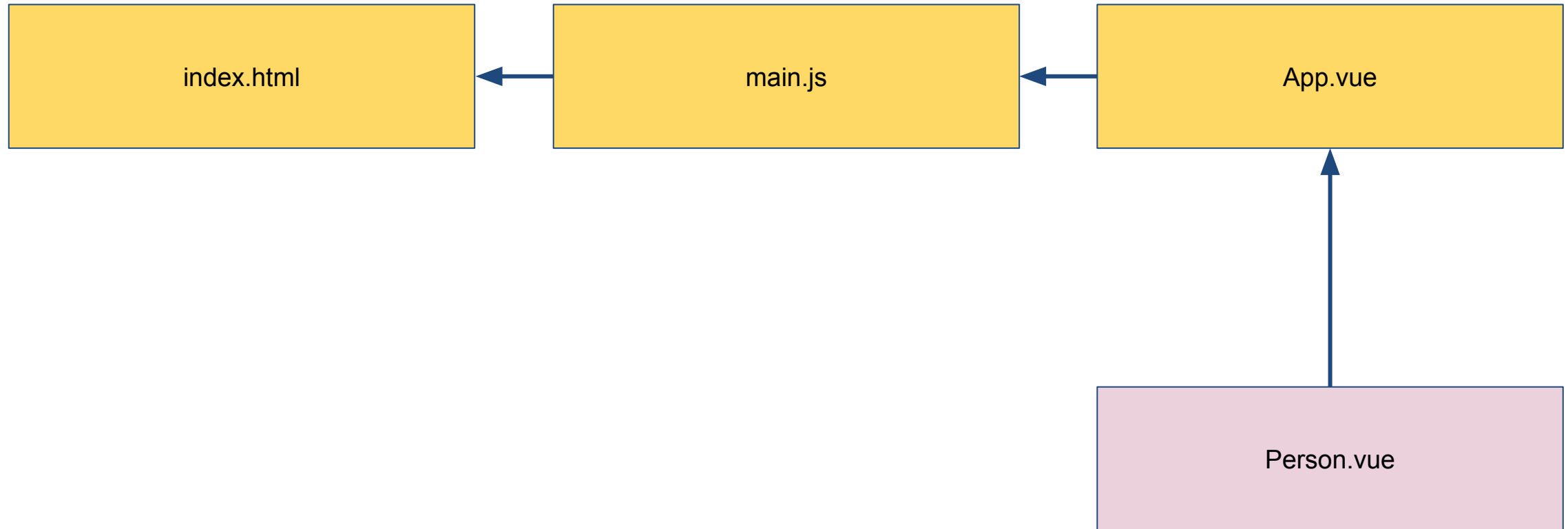
name : honggildong

x : 109

y : 20

Q1 - Composition API

- 다음 프로젝트를 vite기반으로 생성하시오.
- 프로젝트 명 : person-app



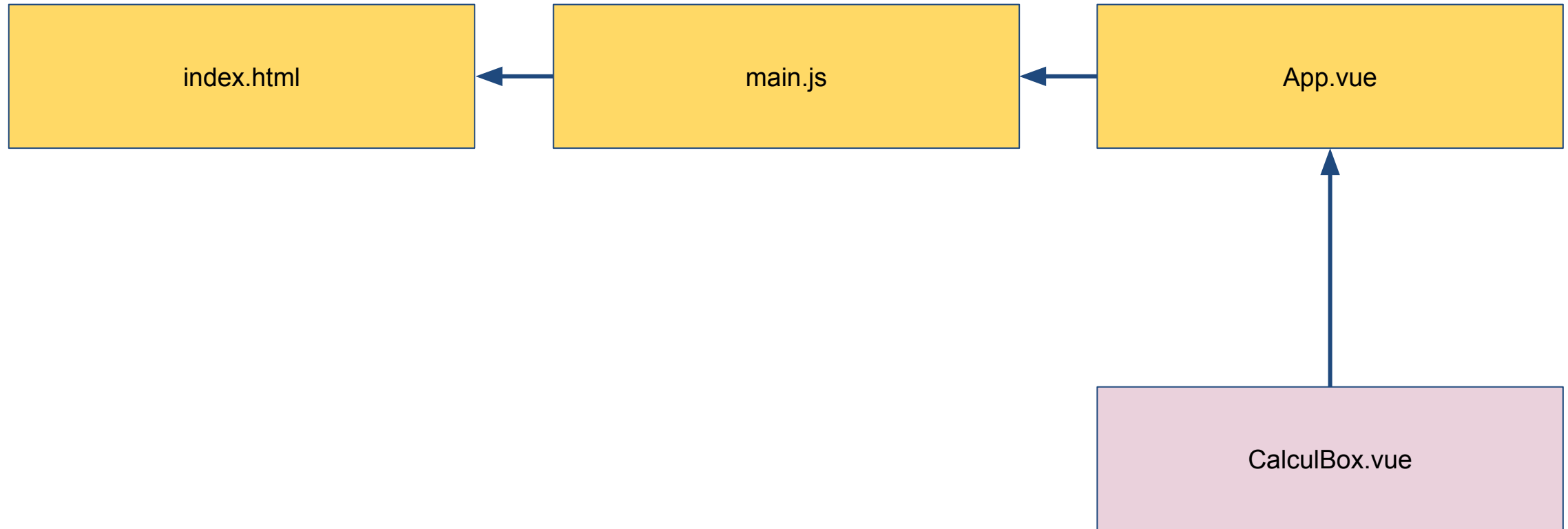
name :	hong
x(키) :	170
y(몸무게) :	88

정보표시

hong의 키는 170cm, 몸무게는 88kg입니다.

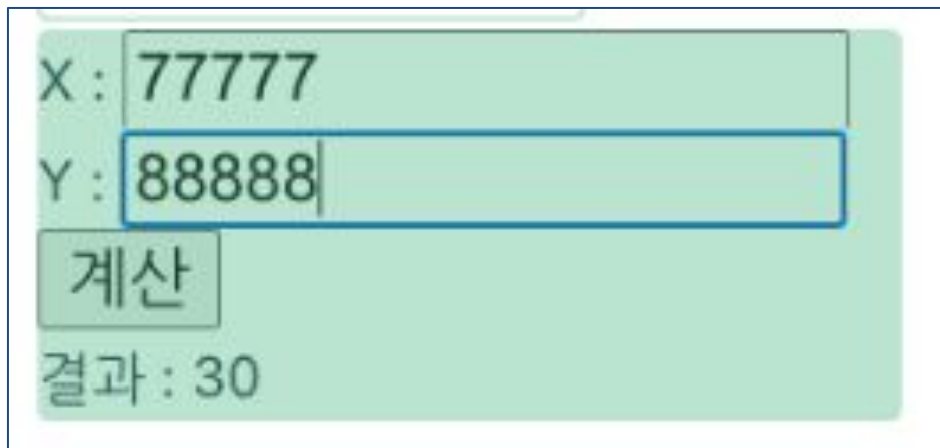
Q2 - Composition API

- 다음 프로젝트를 vite기반으로 생성하시오.
- 프로젝트 명 : **calcu-app**

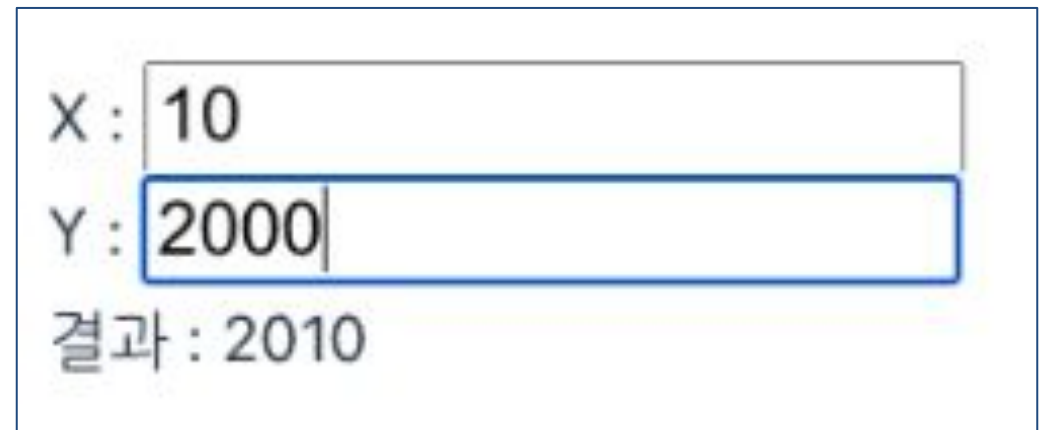


Q2 - Composition API

- 다음 화면을 **Composition API**를 이용하여 완성하시오.
- x, y 데이터에 대해 각각 setup() 메서드를 사용하여 반응성을 설정하시오.
- 계산을 클릭했을 때 두 수의 합이 결과 **result**에 나타나도록 함수를 정의하시오.
 - `computed()`를 사용
- **style**을 이용하여 화면을 스타일링 하시오.
 - 외부파일 또는 내부 파일
 - `bs5`설치 후 사용



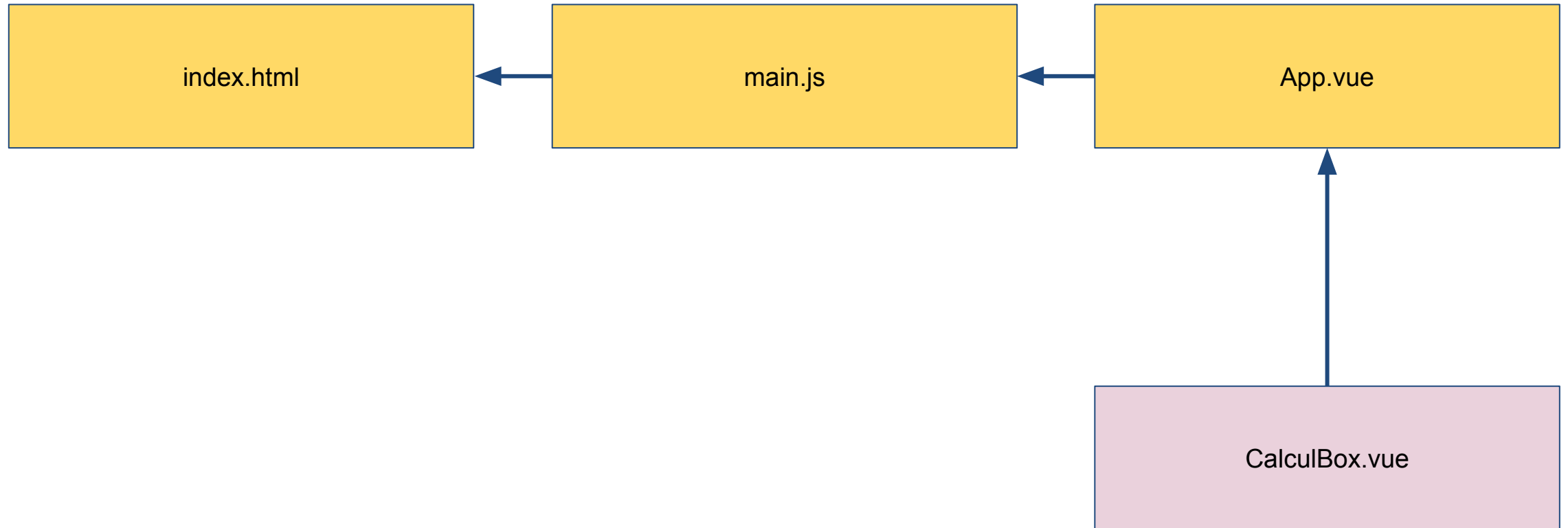
A screenshot of a web application interface with a light green background. It features two input fields: the first is labeled 'X:' and contains the value '77777'; the second is labeled 'Y:' and contains the value '88888'. Below these fields is a button labeled '계산' (Calculate). At the bottom, the text '결과 : 30' (Result : 30) is displayed.



A screenshot of the same web application interface, but with updated values. The 'X:' input field now contains '10' and the 'Y:' input field contains '2000'. The '계산' button is still present. The result at the bottom now reads '결과 : 2010'.

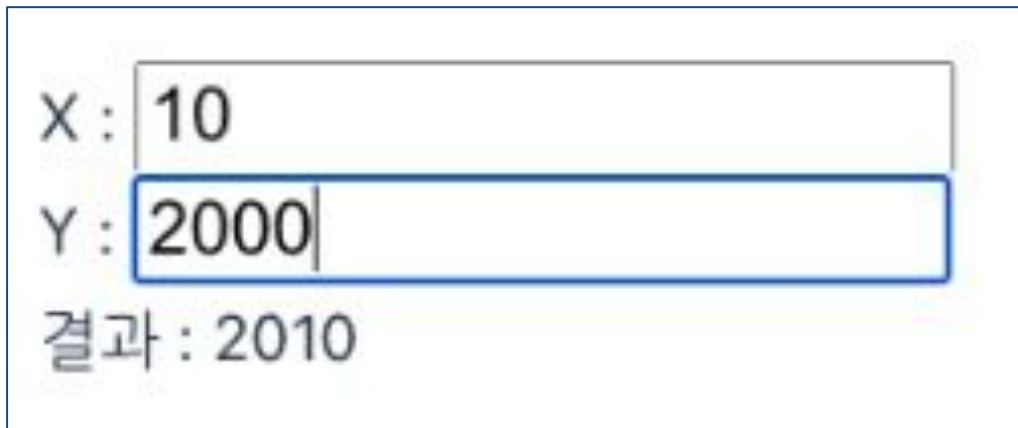
Q2 - Composition API

- 앞에서 생성한 프로젝트를 사용하여 x, y 값을 객체로 반응성을 설정하시오.

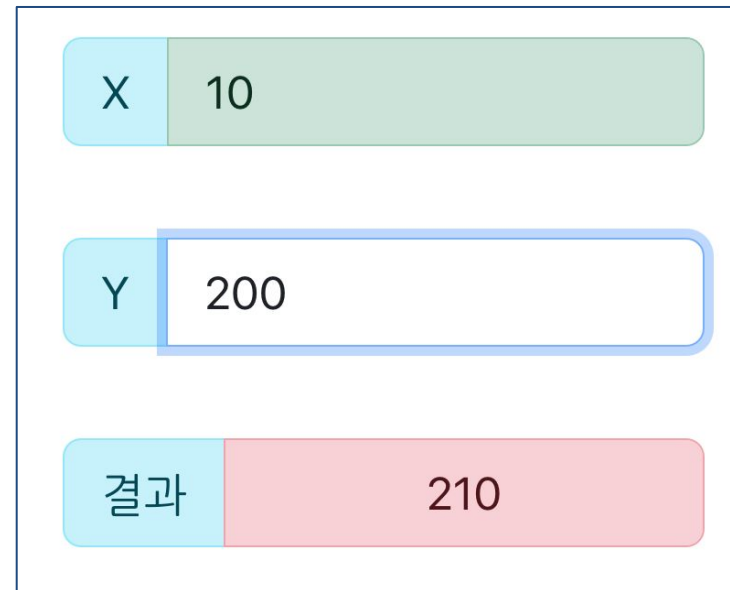


Q2 - Composition API

- 다음 화면을 **Composition API**를 이용하여 완성하시오.
- x, y를 객체로 생성해 **setup()** 메서드를 사용하여 반응성을 설정하시오.
- 계산을 클릭했을 때 두 수의 합이 결과에 나타나도록 함수를 정의하시오.
 - `computed()`를 사용
- **style**을 이용하여 화면을 스타일링 하시오.
 - 외부파일 또는 내부 파일
 - `bs5`설치 후 사용



X : 10
Y : 2000
결과 : 2010



X 10
Y 200
결과 210

꽃이름 입력 :

sun

꽃 가격 :

5000

꽃 수량 :

2



주문

SUN의 가격은 10000원입니다.

Q3 - watch

FLOWER ORDER

꽃 이름

sun

꽃 가격

1000

꽃 수량

2

주문

sun의 가격은 2000원입니다.



FLOWER ORDER

꽃 이름

rose

꽃 가격

1000

꽃 수량

2

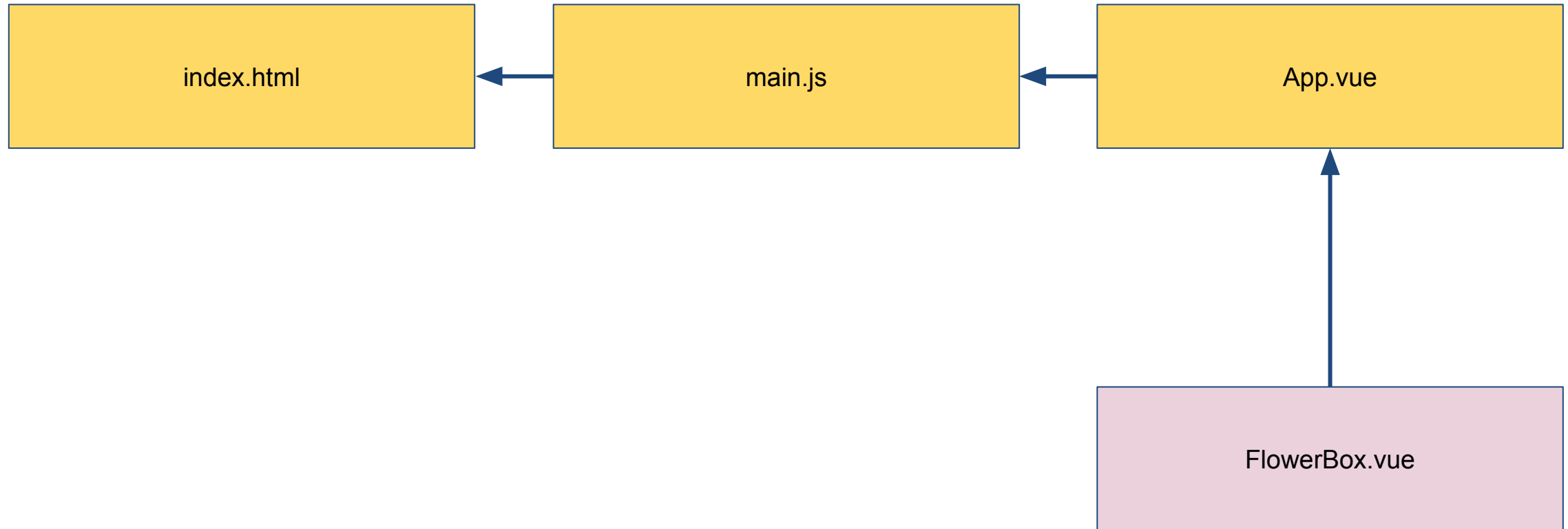
주문

rose의 가격은 2000원입니다.



Q3 - watch

- 다음 프로젝트를 vite기반으로 생성하시오.
- 프로젝트 명 : **flower-app**



Q3 - watch

- **Composition API**를 이용하여 완성하시오.
- **setup()** 메서드를 사용하여 반응성을 설정하시오.
- 주문을 클릭했을 때 두 수의 곱이 결과에 나타나도록 함수를 정의하시오.
 - `watch()`를 사용
 - 주문을 클릭했을 때 해당 꽃의 사진을 화면에 출력
- **style**을 이용하여 화면을 스타일링 하시오.

요트 주문 수량 (대)	10
요트 주문 가격 (만원)	10000

요트 주문 가격 비교

요트 이전 계산 주문 가격: 0만원
 요트 현재 계산 주문 가격:
 100000만원

가격차이 : 100000만원

요트 주문 수량 (대)	10
요트 주문 가격 (만원)	30000

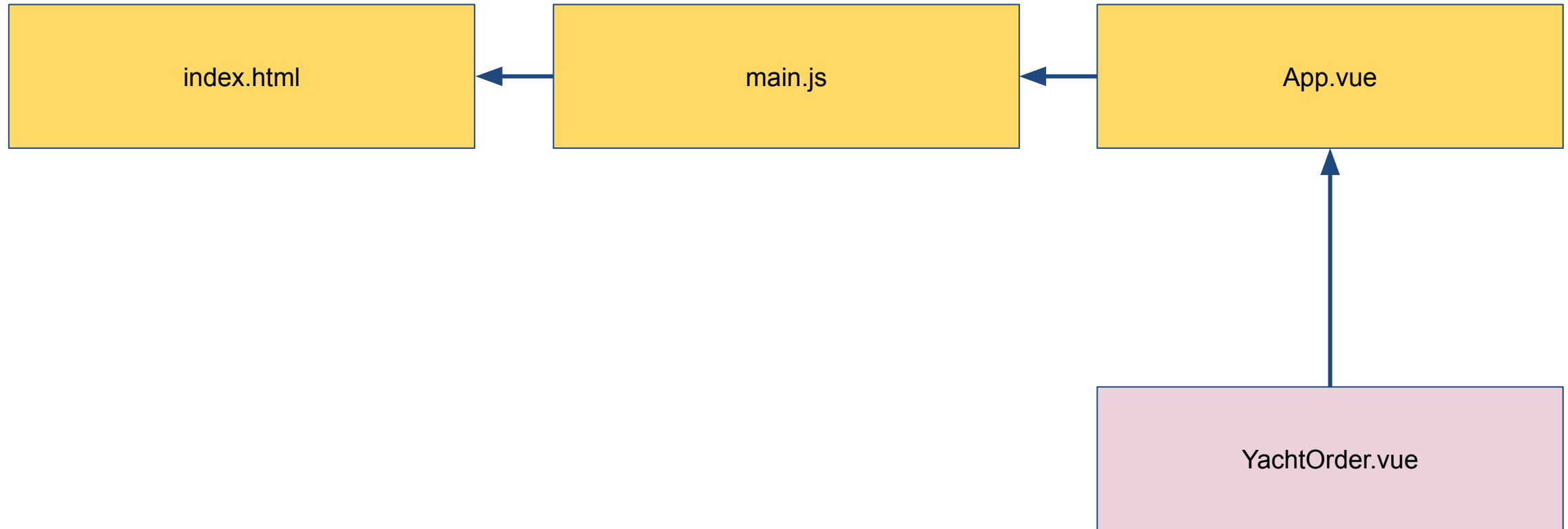
요트 주문 가격 비교

요트 이전 계산 주문 가격:
 100000만원
 요트 현재 계산 주문 가격:
 300000만원

가격차이 : 200000만원

Q4 - watch

- 다음 프로젝트를 vite기반으로 생성하시오.
- 프로젝트 명 : **order-project**



- **Composition API**를 이용하여 완성하시오.
- **setup()** 메서드를 사용하여 반응성을 설정하시오.
- **요트 주문 가격 비교를 클릭 때**
 - 두 수의 현재 곱한 금액이 아래 결과에 출력
 - 두 수의 이전 곱한 금액이 아래 결과에 출력
 - 현재/ 이전 곱한 금액의 차이를 아래 결과에 출력
 - 첫 계산에는 이전 금액은 0으로 출력
- **style**을 이용하여 화면을 스타일링 하시오.
 - 외부파일 또는 내부 파일
 - bs5설치 후 사용

요트 수량

10

요트 가격

20000

요트 주문 가격 비교

요트 이전 계산 주문 가격: 0원

요트 현재 계산 주문 가격: 200000원

가격 차이: 200000원

요트 수량

개

요트 가격

20000

요트 주문 가격 비교

요트 이전 계산 주문 가격: 120000원

요트 현재 계산 주문 가격: 140000원

가격 차이: 20000원

