

데이터베이스 구현

[2001020405_19v4]

1. DBMS 설치하기
2. 데이터베이스 생성하기
3. 데이터베이스 오브젝트 생성하기

송윤정

과정명: [디지털컨버전스] 뷰(Vue) 활용 프론트엔드 개발자 양성과정 22-1

교과목명: 데이터베이스 & SQL

주제

객체데이터로 선언된 외부 데이터파일을 이용하여 쇼핑몰 서브페이지의 상품 썸네일 이미지의 데이터베이스로 구현하려고 한다. 동적 페이지로 제작하기 위한 방법을 시나리오 형식으로 서술하시오.

평가기준 기본사항

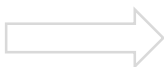
1. 객체 형식의 외부 데이터 파일을 제작 할 수 있다.
2. 엘리먼트 생성 코드를 이용한 동적 DOM을 생성하여 서브페이지를 제작 할 수 있다.
3. 서브페이지 썸네일 항목을 외부 라이브러리를 이용하여 정렬기능을 부여 할 수 있다.
4. 완성된 소스 코드를 포함한 사이트는 웹계정으로 개시 할 수 있다.

1. 객체데이터로 선언된 외부 데이터파일을 제작

*fruit_data.js

applePear라는 객체데이터를 생성한다.
마지막에는 export하여 데이터를 내보내준다.

```
const applePear = [  
  {  
    id: 'product01',  
    link: '../detail_food_fruit/apple.html',  
    img: './images/fruit_01.jpg',  
    imgOn: './images/fruit_01_on.jpg',  
    logo: '이마트몰',  
    delivery: '',  
    fresh: '신선보장',  
    emartFresh: 'E-MART FRESH',  
    name: '유명산지 사과 4~11입/봉 (1.8kg)',  
    price: '10,800',  
    won: '원',  
    sale: '12,800',  
    saleWon: '원'  
  },  
  {  
    id: 'product02',  
    link: '#!',  
    img: './images/fruit_02.jpg',  
  }  
]  
  
export default applePear
```



*fruit.js

import로 applePear 객체데이터를 가져온다.

```
import applePear from "../fruit_data.js";  
  
const productBox = document.querySelector('.product_box')  
  
for(let i=0; i<applePear.length; i++) {  
  let productLi = document.createElement('li')  
  productLi.setAttribute('class', 'product')  
  
  let productLink = document.createElement('a')  
  productLink.setAttribute('href', applePear[i].link)  
  productLi.appendChild(productLink)  
  
  let img = document.createElement('div')  
  let imgImg = document.createElement('img')  
  img.setAttribute('class', 'img')  
  img.appendChild(imgImg)  
  imgImg.setAttribute('src', applePear[i].img)  
  productLink.appendChild(img)  
  
  let text = document.createElement('ul')  
  text.setAttribute('class', 'text')  
  productLi.appendChild(text)  
}
```

2. 엘리먼트 생성 코드를 이용한 동적 DOM을 생성(1)

```
const productBox = document.querySelector('.product_box')

for(let i=0; i<applePear.length; i++) {
  let productLi = document.createElement('li')
  productLi.setAttribute('class', 'product')

  let productLink = document.createElement('a')
  productLink.setAttribute('href', applePear[i].link)
  productLi.appendChild(productLink)

  let img = document.createElement('div')
  let imgImg = document.createElement('img')
  img.setAttribute('class', 'img')
  img.appendChild(imgImg)
  imgImg.setAttribute('src', applePear[i].img)
  productLink.appendChild(img)

  let text = document.createElement('ul')
  text.setAttribute('class', 'text')
  productLi.appendChild(text)

  let emartLi = document.createElement('li')
  let emartLiText = document.createTextNode(applePear[i].logo)
  emartLi.setAttribute('class', 'emart_mall')
  emartLi.appendChild(emartLiText)
  text.appendChild(emartLi)

  if(applePear[i].delivery != 0) {
    let deliveryLi = document.createElement('li')
    let deliveryLiText = document.createTextNode(applePear[i].delivery)
    deliveryLi.setAttribute('class', 'delivery')
    deliveryLi.appendChild(deliveryLiText)
    text.appendChild(deliveryLi)
  }

  let freshLi = document.createElement('li')
  let freshLiText = document.createTextNode(applePear[i].fresh)
  freshLi.setAttribute('class', 'fresh')
  freshLi.appendChild(freshLiText)
  text.appendChild(freshLi)

  let emartFreshLi = document.createElement('li')
  let emartFreshLiText = document.createTextNode(applePear[i].emartFresh)
  emartFreshLi.setAttribute('class', 'emart_fresh')
  emartFreshLi.appendChild(emartFreshLiText)
  text.appendChild(emartFreshLi)
}
```

→ 생성된 엘리먼트를 .product_box 안에 넣어줄 예정

→ for를 이용하여 applePear 객체 데이터 수 만큼 엘리먼트를 생성해 입력해준다.

→ [i]로 객체 데이터 인덱스에 맞는 정보를 가져온다.

setAttribute를 통해서 href 속성에 applePear 객체의 링크를 넣어주었다.

→ .delivery를 필요로 하지 않는 객체도 있기 때문에 if를 사용했다.

(.delivery에는 css로 background-color를 적용 해놓아서, if가 없으면 background-color가 적용되기 때문에 이를 방지하기 위함이다.)

2. 엘리먼트 생성 코드를 이용한 동적 DOM을 생성(2)

```
let nameLi = document.createElement('li')
let nameLiText = document.createTextNode(applePear[i].name)
nameLi.appendChild(nameLiText)
text.appendChild(nameLi)

let priceLi = document.createElement('li')
let priceLiText = document.createTextNode(applePear[i].price)
priceLi.setAttribute('class', 'price')
priceLi.appendChild(priceLiText)
text.appendChild(priceLi)

let wonLi = document.createElement('li')
let wonLiText = document.createTextNode(applePear[i].won)
wonLi.setAttribute('class', 'won')
wonLi.appendChild(wonLiText)
text.appendChild(wonLi)

let saleLi = document.createElement('li')
let saleLiText = document.createTextNode(applePear[i].sale)
saleLi.setAttribute('class', 'sale')
saleLi.appendChild(saleLiText)
text.appendChild(saleLi)

let saleWonLi = document.createElement('li')
let saleWonLiText = document.createTextNode(applePear[i].saleWon)
saleWonLi.setAttribute('class', 'sale_won')
saleWonLi.appendChild(saleWonLiText)
text.appendChild(saleWonLi)

productBox.appendChild(productLi)

imgImg.addEventListener('mouseover', () => {
  imgImg.setAttribute('src', applePear[i].imgOn)
})
imgImg.addEventListener('mouseout', () => {
  imgImg.setAttribute('src', applePear[i].img)
})
}
```

→ 생성된 엘리먼트를 .product_box 안에 넣어준다.

→ mouseover 했을 때 다른 이미지 경로를 주어서 이미지가 바뀌도록 적용했다.

→ mouseout 했을 때 원래의 이미지 경로를 주어서 처음 이미지로 돌아오도록 적용했다.

2. 엘리먼트 생성 코드를 이용한 동적 DOM을 생성(3)

*생성된 엘리먼트 적용결과



이마트몰 신선보장

E-MART FRESH

유명산지 사과 4~11입/봉 (1.8kg)

10,800 원 ±2,800 원



이마트몰 신선보장

저탄소인증 사과 1.4kg/봉

7,980 원



이마트몰 새벽배송 신선보장

[당도선별] 유명산지 청송 햇 홍로사과 1.5kg (봉)

9,800 원 ±2,800 원



이마트몰 신선보장

GAP 죽장사과 4~6입/봉 (1.3kg내외)

5,980 원