

검색기능

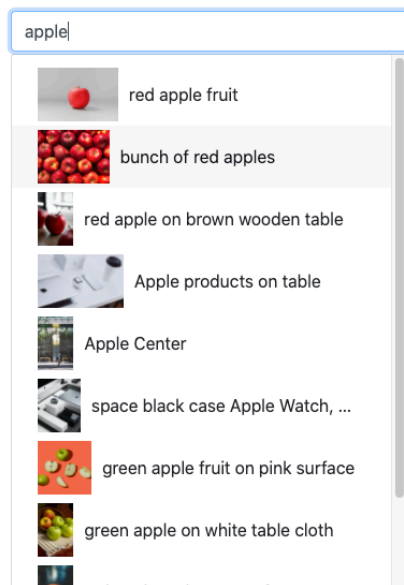
요구사항 : 검색 리스트를 드롭다운 메뉴로 표기하고, 메뉴를 선택하면 선택한 이미지를 화면에 표기합니다.

목표 : 검색어 입력에 따른 API 호출을 제어하는 방법을 학습하게 됩니다.

- 이미지 API : <https://unsplash.com/documentation#search-photos>
- debounce : <https://lodash.com/docs/4.17.15#debounce>

<적용 전>

<검색 리스트>



<검색 리스트 선택>

red apple fruit



1. 자바스크립트로 구현하는 방법

- Unsplash API를 사용하여 이미지 리스트 정보와 단일 이미지 정보를 가져오는 코드를 작성합니다.

```
const accessKey = "hPyF-tz9tHnxaoTwb7q0GTw10Wxwr85cD63lk7d7UE";
const input = document.querySelector('input');
const dropdownMenu = document.querySelector('.dropdown-menu');
const imageInfo = document.querySelector('.image-info');

// 이미지 검색 API
const fetchImages = async search => {
  const res = await axios.get('https://api.unsplash.com/search/photos/', {
    params: {
      client_id: accessKey,
      query: search
    }
  });

  return res.data.results;
};

// 이미지 단일 정보 API
const fetchImage = async id => {
  const res = await axios.get(`https://api.unsplash.com/photos/${id}`, {
    params: {
      client_id: accessKey,
    }
  });

  return res.data;
}
```

- 검색 결과에 따라 드롭다운 메뉴와 리스트를 화면에 출력하는 코드를 추가합니다.

```
// 검색 드롭다운 표기
const onSearch = async e => {
  const images = await fetchImages(e.target.value);
  if(!images.length) {
    dropdownMenu.classList.remove('show');
    return;
  }
  dropdownMenu.classList.add('show');
  dropdownMenu.innerHTML = '';
  listTemplate(images);
}

// 검색 드롭다운 리스트 표기
const listTemplate = (images) => {
  for(const image of images) {
    const {id, alt_description, urls} = image;
    const li = document.createElement('li');
    li.classList.add('dropdown-item');
    li.innerHTML = `
      
      <span>${alt_description}</span>
    `;
    dropdownMenu.appendChild(li);
  }
}

input.addEventListener('input', onSearch);
```

- 검색 리스트를 선택했을 때 단일 이미지를 가져오는 이벤트와 코드를 추가합니다.

```
const onSearch = async e => {.....}

const listTemplate = (images) => {
  for(const image of images) {
    const {id, alt_description, urls} = image;
    const li = document.createElement('li');
    li.classList.add('dropdown-item');
    li.innerHTML = `
      
      <span>${alt_description}</span>
    `;
    dropdownMenu.appendChild(li);

    li.addEventListener('click', async () => {
      dropdownMenu.classList.remove('show');
      input.value = alt_description;
      imageTemplate(await fetchImage(id));
    })
  }
}

// 선택한 이미지 정보 표기
const imageTemplate = (image) => {
  const {urls, alt_description} = image;
  imageInfo.src = urls.regular;
  imageInfo.alt = alt_description;
}

input.addEventListener('input', onSearch);
```

- 현재 코드에서는 검색어를 입력할 때마다 서버와 통신을 진행하게 됩니다.
- 검색어를 입력하고 일정 시간 후에 통신이 진행되도록 코드를 변경해야 불필요한 호출을 크게 줄일 수 있습니다.
- 이때 사용되는 것이 lodash 라이브러리에서 제공하는 **debounce()** 입니다.

```
const onSearch = async e => {.....}

const listTemplate = (images) => {.....}

const imageTemplate = (image) => {.....}

.addEventListener('input', _.debounce(onSearch, 600));
```

2. 제이쿼리로 구현하는 방법

```
const accessKey = "hPyF-tz9tHnxiaoTwb7q0GTwl0Wxwr85cD63lk7d7UE";
const $input = $('input');
const $dropdownMenu = $('.dropdown-menu');
const $imageInfo = $('.image-info');

const fetchImages = async search => {
  const res = await axios.get('https://api.unsplash.com/search/photos/', {
    params: {
      client_id: accessKey,
      query: search
    }
  });
  return res.data.results;
};

const fetchImage = async id => {
  const res = await axios.get(`https://api.unsplash.com/photos/${id}`, {
    params: {
      client_id: accessKey,
    }
  });
  return res.data;
}
```

```

const onSearch = async e => {
  const images = await fetchImages(e.target.value);
  if(!images.length) {
    $dropdownMenu.removeClass('show');
    return;
  }
  $dropdownMenu.addClass('show');
  $dropdownMenu.empty();
  listTemplate(images);
}

const listTemplate = (images) => {
  $.each(images, (index, image) => {

    const {id, alt_description, urls} = image;
    const $li = $(`<li class="dropdown-item"><span>${alt_description}</span></li>`);
    $dropdownMenu.append($li);

    $li.on('click', async () => {
      $dropdownMenu.removeClass('show');
      $input.val(alt_description);
      imageTemplate(await fetchImage(id));
    })
  })
}

const imageTemplate = (image) => {
  const {urls, alt_description} = image;
  $imageInfo.attr('src', urls.regular);
  $imageInfo.attr('alt', alt_description);
}

$input.on('input', _.debounce(onSearch, 600));

```


- data 속성 활용

```
// data 속성 추가
const listTemplate = (images) => {
  $.each(images, (index, image) => {
    const {id, alt_description, urls} = image;
    const $li = $('<li class="dropdown-item" data-image='$
{JSON.stringify({id, alt_description})}'><span>${alt_description}</span></li>`);
    $dropdownMenu.append($li);
  })
}

// 새로 추가된 코드
$(document).on('click', '.dropdown-item', async function() {
  const image = $(this).data('image');
  const {id, alt_description} = image;

  $dropdownMenu.removeClass('show');
  $input.val(alt_description);
  imageTemplate(await fetchImage(id));
})
```