

# Momentum

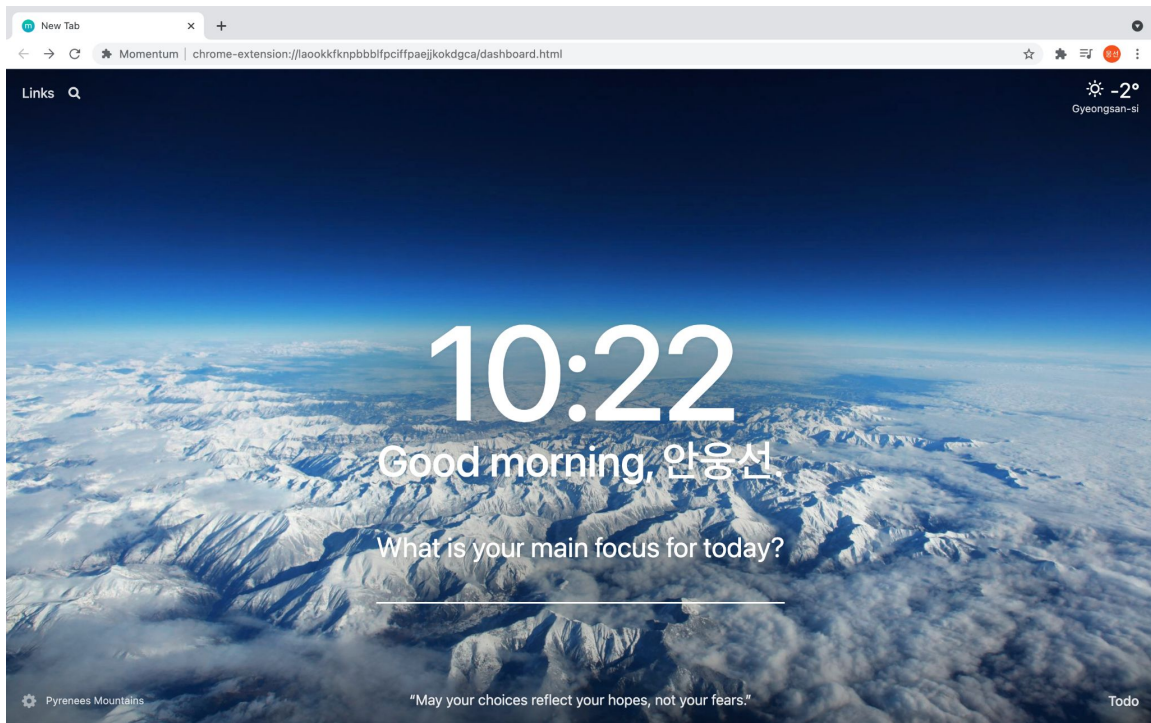
구글 크롬 앱 모멘텀 클론코딩

2021.12.13 ~ 12.17

# 모멘텀 소개

모멘텀은 현재 시간과 현위치의 날씨, todo 기능 등을 제공하는 구글 크롬 확장 프로그램입니다.

깔끔한 레이아웃과 브라우저 시작페이지로써 필요한 핵심적인 기능만을 제공하기 때문에 많은 유저들이 사용하고 있는 앱 입니다.



# 동기와 개발목표

---

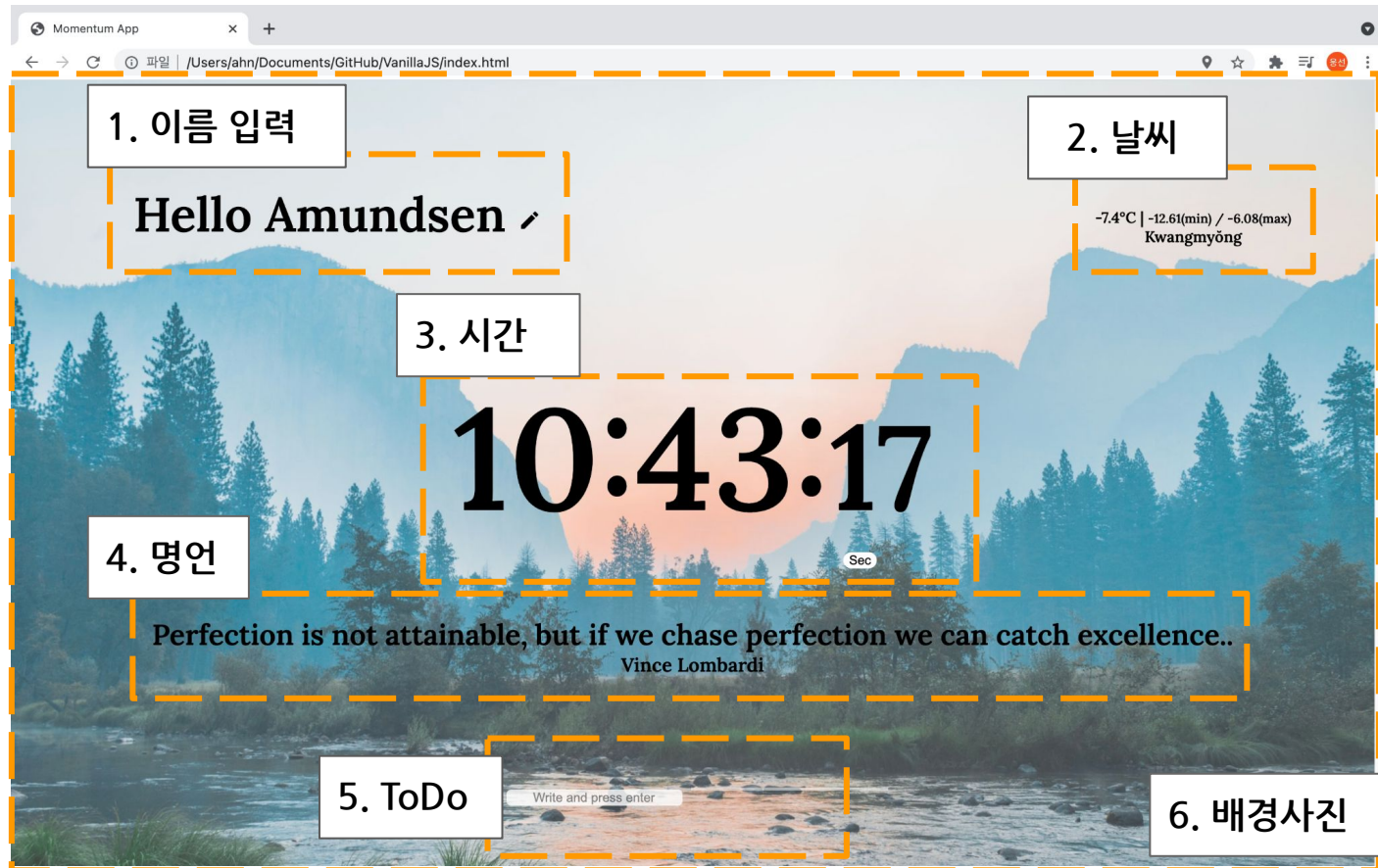
- 동기

저는 이 모멘텀 앱을 사용하며 문득 내가 원하는 배경사진들과 내가 좋아하는 명언들만을 출력해주는 나만의 모멘텀 앱이 있으면 좋겠다는 생각을 하게 되었습니다. 더불어 이 앱이 제공하는 기능들을 순수 자바스크립트만으로 구현해보면 재미도 있고 좋은 학습이 될 것 같아 저만의 작은 앱을 만들게 되었습니다.

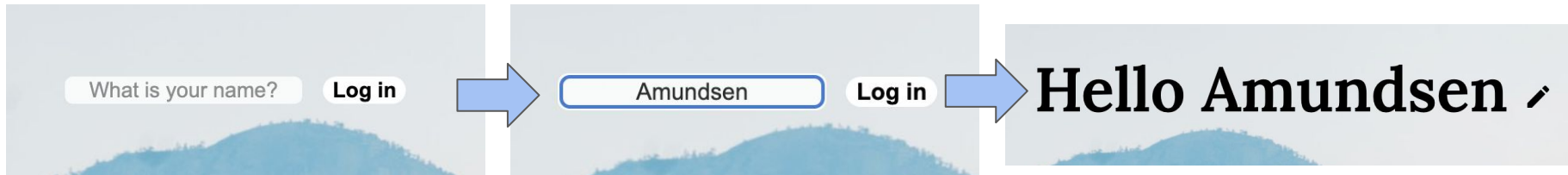
- 개발목표

모멘텀 앱이 제공하는 시간, 날씨, ToDo 등 대표적인 기능들을 모두 구현하고자 했습니다. 위 기능들을 구현하는 데에는 순수 자바스크립트만을 사용해서 해당 언어에 대해 보다 깊은 이해를 얻고자 하였습니다.

# 결과물



# 1. 이름 입력

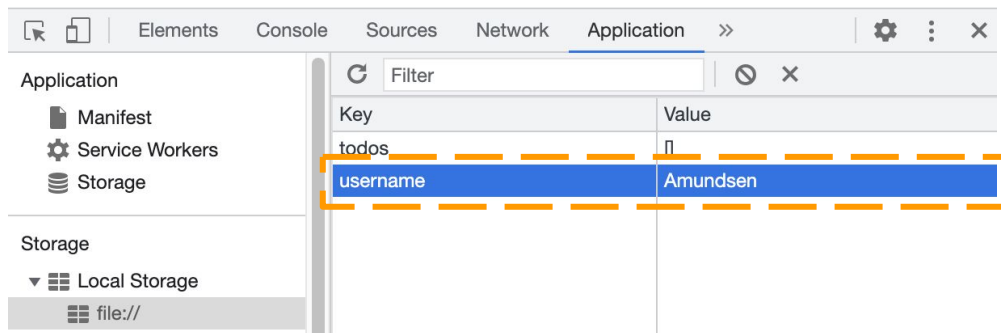


```
// 유저 정보 저장. => localStorage: 브라우저 기본 API로 브라우저에 무언가를 저장할 수 있게 해줌. 작은 DB라고 생각하면 됨.
// localStorage.setItem("username", "Ahn");      username=key / Ahn=value

function paintGreetings(username) {
  loginForm.classList.add(HIDDEN_CLASSNAME);
  salut.classList.remove(HIDDEN_CLASSNAME);
  const date = new Date();
  if((date.getMonth()+1) === 12 & (date.getDate()) === 25) { Amundsen, 2 weeks ago • 22.01.07
    salut.innerText = 'Merry Christmas ${username}';
  } else {
    salut.innerText = 'Hello ${username}';
  }

  edit.classList.add("material-icons");
  edit.classList.remove("hidden");
}

const savedUsername = localStorage.getItem(USERNAME_KEY);
```



form 태그를 만들어 이름을 입력하고 submit하게 되면 입력한 값이 username이라는 key와 로컬 스토리지에 저장되는 방식으로 구현하였습니다. 해당 값을 불러와 화면에 “Hello 이름명” 형식으로 출력해주고, 우측 펜 모양의 수정버튼을 누르면 로컬스토리지에서 username에 대한 value를 삭제하는 방식입니다.

## 2. 날씨

자바스크립트의 geolocation API와 openweathermap.org의 API를 사용하여 구현하였습니다.

geolocation의 `getCurrentPosition` 함수를 이용해 현재 위치의 위도와 경도값을 받아내고 외부 API를 이용해서 현재 좌표에 대한 지역명과 현재기온, 최저/최고기온을 가져와 출력해주었습니다.

```
navigator.geolocation.getCurrentPosition(onGeoOk, onGeoError);
```

```
function onGeoOk(position) {  
  const latitude = position.coords.latitude; // 위도  
  const longitude = position.coords.longitude; // 경도  
  
  const url = `https://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${API_KEY}`;  
  // fetch(): 자바스크립트가 url을 불러옴, 끝에 units=metric 붙여줘야 섭씨로 출력됨. You, seconds ago • Uncommitted  
  
  fetch(url).then(response => response.json()).then(data => {  
    const location = data.name;  
    const temp = data.main.temp;  
    const tempMin = data.main.temp_min;  
    const tempMax = data.main.temp_max;  
    const weather = data.weather[0].main;  
  
    const weatherSpan = document.querySelector("#weather span:first-child");  
    const locationSpan = document.querySelector("#weather span:last-child");  
  
    console.log(weatherSpan);  
    weatherSpan.innerHTML = temp+"°C | <small>"+tempMin+"(min) / "+tempMax+"(max)</small>";  
  
    locationSpan.innerText = location;  
  });  
}
```

-6.14°C | -6.14(min) / -6.14(max)  
Kwangmyŏng

### 3. 시간



Date 객체의 함수들을 이용해 현재 시간을 가져오고, 가져온 값들을 h2태그에 1초에 한 번씩 넣어주는 방식으로 시계를 구현하였습니다.

초 아래에 Sec버튼을 누르면 시간과 분만 출력되게끔 하였습니다.

```
const clock1 = document.querySelector("#clock1");
const clock2 = document.querySelector("#clock2");
const noSecBtn = document.querySelector("#noSecBtn");

function getTime() {
  const date = new Date();
  const hours = String(date.getHours()).padStart(2, "0"); // padStart: 문자열이 시작(s
  const minutes = String(date.getMinutes()).padStart(2, "0"); // 문자열의 길이는 2여야 하고,
  const seconds = String(date.getSeconds()).padStart(2, "0");
  clock1.innerHTML = `${hours}:${minutes}<small>${seconds}</small>`;
  clock2.innerText = `${hours}:${minutes}`;
}

// setInterval(sayHello, 5000); // 5초에 한 번씩 sayHello 함수 실행.
// setTimeout(sayHello, 5000); // 페이지 실행 후 5초 뒤 함수 실행

getTime() // 한번 실행해줘야 페이지 실행과 동시에 시간이 출력됨. 아니면 1초를 기다려야 함.
setInterval(getTime, 1000);

noSecBtn.addEventListener("click", function noSecFnc() {
  clock1.classList.toggle("hidden");
  clock2.classList.toggle("hidden");
});
```



## 4. 명언

```
const quotes = [
  {
    quote: "The best preparation for tomorrow is doing your best today.",
    author: "H. Jackson Brown, Jr.",
  },
  {
    quote: "The undertaking of a new action brings new strength.",
    author: "Richard L. Evans",
  },
  {
    quote: "Perfection is not attainable, but if we chase perfection we can catch excellence..",
    author: "Vince Lombardi",
  },
  {
    quote: "There are two ways of spreading light: to be the candle or the mirror that reflects it.",
    author: "Edith Wharton",
  },
  {
    quote: "Somewhere, something incredible is waiting to be known.",
  },
];
```

```
// Math.random(): 숫자 랜덤 생성
// Math.floor(): 내림, ceil(): 올림, round(): 반올림
const todayQuote = quotes[Math.floor(Math.random() * quotes.length)];

quote.innerText = todayQuote.quote; // quote 오브젝트 가져오기
author.innerText = `${todayQuote.author}`;
```

People may

There is nothing either good or bad but thinking makes it so

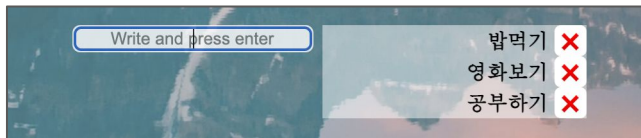
The most beautiful thing in the world is, of course, the world itself.

Wallace Stevens

제가 좋아하는 명언 10개와 위인들의 이름을 quotes라는 배열에 넣어두고 random 함수를 이용해 quotes 배열에서 랜덤 순서로 명언과 인물의 이름을 가져오도록 하였습니다.



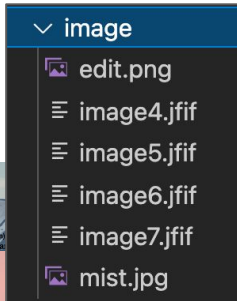
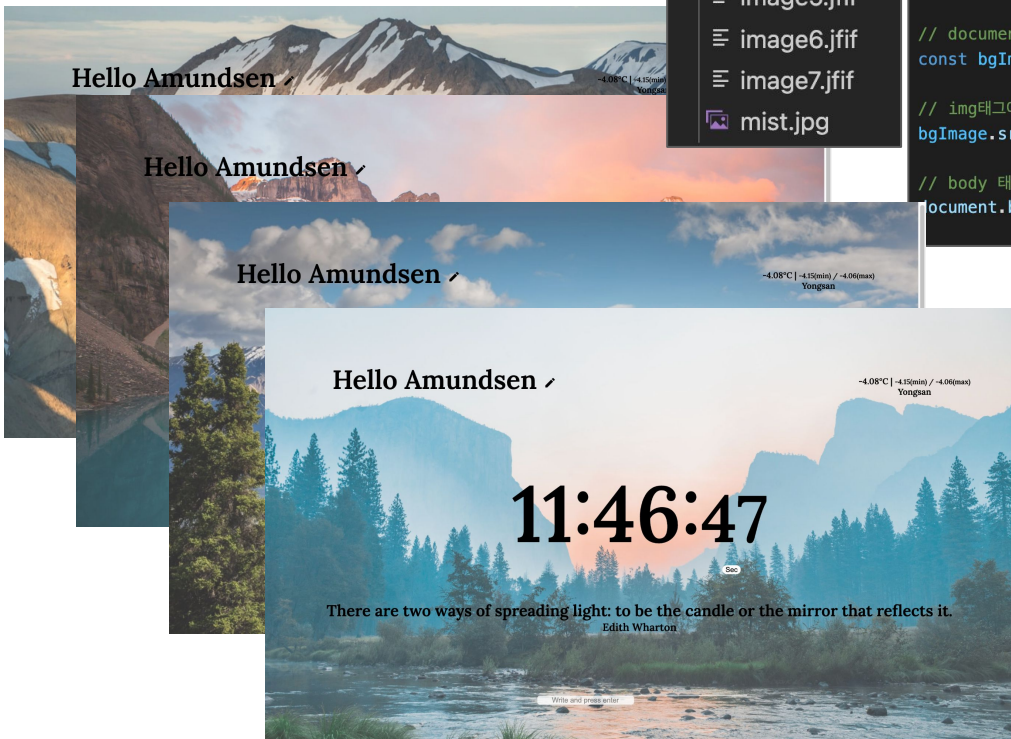
## 5. Todo



Key	Value
todos	[{"id": "1642646030119", "text": "밥먹기"}, {"id": "1642646050269", "text": "영화보기"}, {"id": "1642646050269", "text": "공부하기"}]
username	Amundsen

이름을 입력했을 때와 마찬가지로, 원하는 ToDo 리스트를 입력하고 엔터키를 누르게 되면 해당 ToDo 내용이 랜덤한 숫자를 갖는 id key와 함께 오브젝트 형태로 로컬스토리지에 저장됩니다. 그리고 ToDo 내용은 우측에서 리스트 형태로 출력되는데, 이때 각각의 li 태그는 ToDo 내용이 로컬스토리지에서 갖는 id와 같은 id 값을 가지게 됩니다. X버튼을 눌렀을 때 해당 li가 갖는 id값과 동일한 id 값을 로컬스토리지에서 삭제하는 방식으로 삭제기능을 구현하였습니다.

## 6. 배경사진



```
const images = ["image5.jfif", "image4.jfif", "image6.jfif", "image7.jfif"];  
  
const chosenImage = images[Math.floor(Math.random() * images.length)]; // floor로 내렸기때문에  
  
// 자바스크립트를 이용해서 HTML 안에 이미지 넣기  
  
// document 안에 img태그 요소를 만들고 bgImage라는 변수에 넣는다.  
const bgImage = document.createElement("img");  
  
// img태그에 src속성을 넣어주고 그 안에 경로를 넣어준다. 이미지파일명은 랜덤 생성.  
bgImage.src = `image/${chosenImage}`;  
  
// body 태그 안에 생성한 img태그 요소를 넣어준다.  
document.body.appendChild(bgImage); // append: 끝에 위치시킴. prepend: 가장 앞에 위치시킴.
```

미리 프로젝트 내에 원하는 배경사진들을 저장해두고 자바스크립트에서 images라는 배열을 만들어 저장해두었던 이미지 파일명들을 넣어주었습니다.

명언 부분과 동일하게 random 함수를 이용해 무작위로 이미지 파일명을 받아오고, 해당 이미지를 배경사진으로 적용시켰습니다.

## 느낀점 / 결론

자바스크립트를 사용해서 무언가 결과물을 낸 것이 처음이라 많이 어설프기도 했지만 기존에 사용하던 앱의 기능들을 직접 구현해볼 수 있었기에 뜻 깊은 시간이었습니다.

특히 로컬스토리지에 저장하는 방식을 사용해서 페이지 새로고침을 해도 이름과 ToDo 리스트가 유지되도록 했는데, 이런 방식은 처음 사용해보았기에 굉장히 새로웠습니다.

이번 클론 코딩은 프로젝트라고 하기엔 좀 소소했지만 결과적으로 자바스크립트에 대해 전보다 확실히 깊게 이해할 수 있었고 써보지 못했던 기능들을 많이 사용해보았기 때문에 개인적으로 만족스러운 프로젝트였습니다.

Thank you 😊