



바닐라JS 챌린지

신청자 **875명**

기간 **2주**

Day 11 of 15

Previous Solution

Today's Blueprint

[챌린지 사용방법](#)

[바닐라JS 챌린지 진도표](#)

[Code Sandbox 사용방법](#)

[슬랙 이용방법](#)

Today's Assignment



정답 해설





(1)

- `const xmasDay = new Date(`${new Date().getFullYear()}-12-25:00:00:00+0900`);`
- 크리스마스 날짜를 얻기 위한 코드입니다.
- 올해의 연도를 얻기 위해 `getFullYear()` 함수를 사용했습니다. `Date` 함수는 기본적으로 미리 초로 계산되지만, `Date`의 인수로 문자열을 전달하면 지정된 형태로 날짜와 시간을 반환합니다. 여기서 [템플릿 리터럴](#)을 사용했습니다.
- 위 코드는 `Sat Dec 25 2021 00:00:00 GMT+0900 (한국 표준시)`를 생성합니다.

(2)

- `const now = new Date();`
- `Date`의 인수에 아무것도 적지 않으면 현재 시간을 생성해 줍니다.

(3)

- `const difference = new Date(xmasDay - now);`
- `Date` 함수로 생성한 `Date` 객체는 연산이 가능합니다. 즉 `xmasDay`와 `now`를 연산할 수 있습니다.
- 크리스마스까지 얼마나 남았는지를 알기 위해 `xmasDay`에서 `now`를 빼주고 그 값을 `difference`로 선언해 줍니다.

(4)

- 1초는 1000밀리 초입니다. 미리 초 단위의 값을 가지고 있는 `difference` 변수를 1000으로 나눠주어 `secondsInMs`에 저장해 줍니다.
- 같은 방법으로 `secondsInMs`를 이용해 `minutesInMs`를 `minutesInMs`를 이용해 `hoursInMs`를 구해줍니다. 1시간은 60분, 1분은 60초, 하루는 24시간으로 나눠줍니다.
- `Math.floor()`은 나누기 후 생기는 소수점을 없애기 위해 사용합니다.



```
1  const secondsInMs =  
    Math.floor(difference / 1000);  
2  const minutesInMs =  
    Math.floor(secondsInMs / 60);  
3  const hoursInMs =  
    Math.floor(minutesInMs / 60);  
4  const days =  
    Math.floor(hoursInMs / 24);
```

(5)

- 화면에 출력될 값을 가지고 있는 변수들입니다.
- 출력되는 값들은 크리스마스까지 남은 일수, 남은 시간, 남은 분, 남은 초이므로 나머지 값을 얻을 수 있는 모듈러(%) 연산을 사용해야 합니다.
- 모듈러(%) 연산을 할 때 시간, 분, 초의 단위에 맞춰 60 또는 24로 나눠주면 됩니다.

```
1  const seconds = secondsInMs %  
    60;  
2  const minutes = minutesInMs %  
    60;  
3  const hours = hoursInMs % 24;
```

(6)

- 남은 시간이 10보다 작아질 경우 앞에 0을 붙여 두 자리로 나타내기 위한 코드입니다. ex) 9h -> 09h
- 위 코드에선 [삼항연산자](#)를 사용하고 있습니다. if 문처럼 생각하면 됩니다.



입니다.

- 즉, days 가 10보다 작을 때 0\${days} 가 클 때 days 가 출력됩니다.
- 같은 방법으로 hoursStr, minutesStr, secondsStr 도 작성해 주면 됩니다.

```
1 const daysStr =  
  `${days < 10 ? `0${days}` :  
    days}d`;  
2 const hoursStr =  
  `${hours < 10 ? `0${hours}` :  
    hours}h`;  
3 const minutesStr =  
  `${minutes < 10 ?  
    `0${minutes}` : minutes}m`;  
4 const secondsStr =  
  `${seconds < 10 ?  
    `0${seconds}` : seconds}s`;
```

(7)

- clockTitle.innerHTML = `\${daysStr} \${hoursStr} \${minutesStr} \${secondsStr}`;
- clockTitle 의 innerHTML 프로퍼티에 앞에서 구한 변수들을 넣어 화면에 출력해 주기 위한 코드입니다.

(8)

- setInterval(getTime, 1000);
- setInterval 함수를 사용해 getTime 함수를 1000밀리 초(1초)마다 실행되게 해줍니다.
- 하지만 이렇게 되면 브라우저가 시작하고 1초간은 시간이 화면에 나타나지 않습니다. 그러므로 setInterval() 전에 getTime() 을 먼저 한번 실행시켜 줍니다.

결론



는 쉼터시였습니다.

오늘의 과제



힌트 엿보기



NOMAD CODERS

유한회사 노마드컴퍼니

대표: 박인

개인정보책임관리자: 박인

사업자번호: 301-88-01666

주소: 서울시 마포구 백범로 8, 532호

-

원격평생교육원: 서울시 서부교육지원청(제2020-13호)

통신판매업 신고번호: 2020-서울마포-1987

이메일: help [at] nomadcoders.co

NAVIGATION

Courses

Challenges

Community

FAQ

Roadmap

LEGAL

이용약관

개인정보취급방침

취소 및 환불정책



Clone Startups. Learn to Code.



© 2017-2022 Nomad Coders. All rights reserved.

