

쿠키 만들기



클라이언트 PC에 쿠키를 만드는 기능이다.

쿠키 생성하기 (예시)

```
// 기본 쿠키 설정
document.cookie = "username=John Doe";

// 쿠키에 만료일을 추가
document.cookie = "username=John Doe; expires=Fri, 31 Dec 2024 23:59:59 GMT";

// 도메인, 경로, 보안 설정
document.cookie = "username=John Doe; expires=Fri, 31 Dec 2024 23:59:59 GMT; domain=.example.com; path=/; secure;";
```



쿠키의 구성

```
//key = value
// 기본 쿠키 설정
document.cookie = "userName=값";
```

- name과 value 속성은 데이터를 저장하고 읽는데 사용
- 쿠키를 사용할 때, user와 value는 반드시 저장



Option



Expire

```
document.cookie = "username=John Doe; expires=Fri, 31 Dec 2024 23:59:59 GMT";
```

- 쿠키의 파기날짜를 정의
- GMT 혹은 UTC 형식으로 날짜 입력
- 파기 날짜 입력하지 않을 경우, 브라우저 종료 시 쿠키 삭제 (해당 브라우저를 모두 종료)

```
var date = new Date();
//현재 날짜에서 더하기
date.setTime(date.getTime() + (day * 24 * 60 * 60 * 1000));
console.log(date.toUTCString());
expires = "; expires=" + date.toUTCString();

document.cookie = "user=value; "+expires;
```

응용하기

```
function gmtCookie() {
    let day = document.getElementById('cookieDay').value;
    let name = document.getElementById('cookieName');
    let value = document.getElementById('cookieValue');
    if(day == 0) {
        console.log('cookie 날짜는 0');
        document.cookie = name.value+'='+value.value;
    } else {
        let date = new Date();
        //현재 날짜에서 더하기
        date.setTime(date.getTime() + (day * 24 * 60 * 60 * 1000));
        console.log(date.toUTCString());
        expires = "; expires=" + date.toUTCString();
        document.cookie = name.value+'='+value.value+';'+expires;
    }
}
```

```
}  
}
```

Max-age

```
document.cookie = "username=va; max-age=3600";
```

- expire 대안이다.
- 쿠키 만료 기간을 설정할 수 있다.
- 현재부터 시작하여, 만료 시간까지 초로 계산한다. (새로 고침 하면, 쿠키가 삭제 된다.)
- Max-Age가 `-1` 은 삭제를 의미한다.

Secure

```
document.cookie="user=va; Secure";
```

- https에서만 적용되는 cookie 이다.

Domain

```
document.cookie="user=va; Domain=localhost";
```

- Domain속성을 입력하지 않는다면, 현재 도메인이 자동으로 등록된다.
- 페이지 요청과 비교하여, 현재의 도메인과 저장된 Cookie의 도메인이 일치하지 않는다면
쿠키의 접속을 할 수 없다.

Path

```
document.cookie="user=va; path=/;";
```

- Path속성은 특정경로를 지정할 때 많이 사용 된다. (대표적으로 폴더의 경로 이다.)
- Path속성을 지정한 곳으로부터 현재폴더 혹은 하위 디렉터리도 접근이 가능하다.
- Path속성을 '/'으로 정의할 경우, 도메인 내의 모든 곳에서 접근이 가능하다.

Cookie 불러오기

```
function getCookies(name) {  
    var nameEQ = name + "=";  
    var ca = document.cookie.split(';');  
    for (var i = 0; i < ca.length; i++) {  
        var c = ca[i];  
        while (c.charAt(0) == ' ') c = c.substring(1);  
        if (c.indexOf(nameEQ) == 0) {  
            return c.substring(nameEQ.length, c.length);  
        }  
    }  
    return null;  
}  
  
//cookie 저장 형태  
cook=100; cook1=150
```

- cookie를 불러올 때, js코드 로직을 만들어서 불러온다.

Cookie 삭제하기

```
function delCookie() {  
    let name= document.getElementById('inpName').value;
```

```
console.log('name: ', name);  
document.cookie = name + '=; Max-Age=-1;';  
}
```

//Path=/으로 저장했을 경우

```
document.cookie="user=va; Max-Age=-1; Path="/;
```

- cookie를 삭제할 때, 설정 된 **Domain** 혹은 **Path** 와 일치 되어야 삭제가 가능하다.
- Max-Age가 **-1** 은 삭제를 의미한다.

예시 파일

```
//요일  
Sun: 일  
Mon: 월  
Tue: 화  
Wed: 수  
Thu: 목  
Fri: 금  
Sat: 토
```