```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>usingPromises</title>
</head>

<body>

    <script>
        /*
,            ***** using Promises ****
,
,            Promise를 사용하려면 뭔가를 불러올건데
,
,            * Then
,
,            Then을 사용하는 방법을 볼것이다.
,            자바스크립트에 Promise가 끝난 이후의 명령어를 전달하려면
,            언제 끝나는것은 중요하지않고 끝난 '이후'
,            Promise가 끝이났을때 값을 돌려달라고해야한다.
,
,            const amISexy = new Promise((resolve, reject) => {
,                setTimeout(resolve, 3000, "yes you are");
,            });
,
,            amISexy.then(value => console.log(value));
,            새로고침 한 직후에는 뜨지 않지만 3초후에는 값을 돌려받는다.
,            지금 우리가 하는것은 Promise를 통해 값을 얻는것이다.
,            변수부분에 넣고싶은 이름을 넣어주면됨.
,
,            윗부분의 amISexy쪽에서 했던부분은
,            API에서 무언가를 불러오고 유저의 local storage에서 쿠키라던가 이런것을 불러오는것이다.
,            그리고 어떤값을 resolve로 입력해주면 입력한 값이 then이후에 들어오게되는것이다.
,
,            * 아마 대부분이 본인이 사용할 Promise를 만들지는 않고
,            다른사람들을 위해 Promise를 만드는 작업을 할것이다.
,
,            Promise를 return해줄때는 then을 사용하면된다.
,            이것이 then을 넣어서 Promise를 만드는 방법
,
,            ** function 넣기
,
,            원하면 function도 넣어줄 수 있다.
,            const amICool = new Promise((resolve, reject) => {
,                resolve("yes you are cool")
,            });
,            const thenFn = (value) => console.log(value);
,            amICool.then(thenFn);
,            이런식으로 작성
,
,            ***** Promise error *****
,
,            Promise에 에러가 있다는것을 어떻게 알까.
,            현재는 바로 적용이 되었는데 적용이 되지 않았다면?
,            그때가 reject를 써야할 타이밍이다.
,
,            Promise에 에러가 생기면 우린 그걸 catch하면 된다.
```

```
,
,             const amIShy = new Promise((resolve, reject) => {
,                setTimeout(reject, 3000, "you are ugly")
,            });
,            amIShy.then(value => console.log(value))
,
,            Uncaught (in promise) you are ugly 라고 내가 쓴 글도 함께 에러로 나오고있다.
,            에러를 잡지 못했기 때문에 콘솔에서 에러를 못잡았다고 뜨는것이고, 에러를 만들어볼것이다.
,
,            그럴때에 * catch를 써줌
,            amIShy
,                .then(result => console.log(result))
,                .catch(error => console.log(error))
,            catch문에서 error메세지로 나옴.
,
,            * Promise에서 resolve가 되면 then구문을 실행할것이고,
,            * Promise에서 reject가 되면 catch구문이 실행.
,
,            ****
,            setTimeout(reject, 1000, "you are ugly")
,            이부분을 진행할 때는 Promise들을 좀 확인해보아야한다.
,            어떻게 사용해야할지에 대해서. 어떻게 구성할지. 어떻게 쓸모있게 만들지.
,            * then이 먼저 실행되고 catch가 실행된다고 생각하는것은 옳지않고
,            둘중에 한가지가 조건에 맞게 진행된다고 보아야한다.
,            resolve 하거나 reject 할때 각기 따로따로로 생각해야함.
,         */

        // then
        const amISexy = new Promise((resolve, reject) => {
            setTimeout(resolve, 3000, "yes you are");
        });

        amISexy.then(item => console.log(item));

        // use function
        const amICool = new Promise((resolve, reject) => {
            resolve("yes you are cool")
        });
        const thenFn = (value => console.log(value));
        amICool.then(thenFn);

        // reject
        const amIShy = new Promise((resolve, rejectFunction) => {
            setTimeout(rejectFunction, 1000, "you are ugly")
        });
        amIShy
            .then(result => console.log(result))
            .catch(error => console.log(error));


    </script>
</body>

</html>
```