



7_20_2022__js_promise,closure

Step09_promise.html

~

closure

```
function test(num){
    setTimeout(function(){
        document.write("num: "+num);
    }, 5000)
}

test(10);
test(20);

//10이 출력되고 20이 출력됨
```

독립적인 함수

```
let myName="두부"

function test(num){
    const delay = Math.random() * 5000 + 5000;
    setTimeout(function(){
        document.write("num: "+num);
        document.write("nyName"+myName)
    }, delay)
}

test(10);
test(20);
test(30);
```

`document.write("num: "+num);`

`document.write("nyName"+myName)`

이 둘은 함께 처리됨(각각 독립되지 않음)

```

let myName="두부"

function test(num){
  const delay = Math.random() * 5000 + 5000;
  setTimeout(function(){
    document.write("num: "+num);
    document.write("myName"+myName)
  }, delay);
  num=999;
}
test(10);
test(20);
test(30);

```

999가 결과로 출력된다.

```

function test(num){
  const delay = Math.random() * 5000 + 5000;
  setTimeout(function(){

와

    num=999;

```

바로 실행되지만

```

document.write("num: "+num);
    document.write("myName"+myName)
  }, delay);

```

는 딜레이가 걸려서

```

let myName="두부"

function test(num){
  const delay = Math.random() * 5000 + 5000;
  setTimeout(function(){
    document.write("num: "+num);
    document.write("myName"+myName)
  }, delay);
  num=delay;
}
test(10);
test(20);
test(30);

```

딜레이가 걸리는 시간을 확인 할 수 있다 이해하기 쉬움

```
let myName="두부"

function test(num){
  const delay = Math.random() * 5000 + 5000;
  setTimeout(function(){
    document.write("num: "+num);
    document.write("myName"+myName)
  }, delay);
  num=delay;
}
test(10);
test(20);
test(30);
```

가 실행중에 console창으로 값을 출력되기 전에 바꾸면 바뀐다!

javascript closure