

03

JavaScript Basic

JavaScript

2장. Javascript Basic

I. 자바스크립트 시작하기

II. 변수와 데이터 타입

III. Operator

IV. Control Flow

V. Function

VI. Scope와 Hoisting

VII. Event Programming

02

02-1. 자바스크립트 시작하기

JavaScript Basic

자바스크립트 코드 위치 – HTML 문서 내에 작성

- 브라우저에서 자바스크립트를 실행시키려면 HTML 파일이 있어야 합니다.
- 자바스크립트를 HTML 에 추가하려면 <script> 태그를 이용해야 합니다.
- <script> 태그 사이에 추가된 자바스크립트는 브라우저의 자바스크립트 엔진에 의해 실행되게 됩니다.

```
<script>
  const myButtonClick = () => {
    let title = document.getElementById('title')
    if(title.style.backgroundColor === 'yellow'){
      title.style.backgroundColor = 'pink'
    }else {
      title.style.backgroundColor = 'yellow'
    }
  }
</script>
```

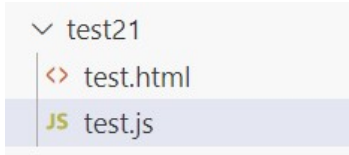
자바스크립트 코드 추가

자바스크립트 코드 위치 – HTML 문서 내에 작성

- `<script>` 태그는 HTML 문서 어디에다 작성해도 상관없지만 일반적으로 `<head>` 태그 사이에 추가하거나 `<body>` 태그 사이에 추가됩니다.
- `<script>` 태그의 역할은 `<script>` 태그로 감싸 있는 코드는 HTML 코드가 아니라 자바스크립트 코드라고 선언하는 역할을 합니다.
- 그래서 `<script>` 태그의 코드를 자바스크립트 엔진에 의해 처리되게 하게 됩니다.

자바스크립트 코드 위치 - HTML 문서 외부에 작성

- 자바스크립트 코드를 HTML 에서 분리시켜서 외부 파일로 작성할 수 있습니다.
- 외부 파일로 자바스크립트를 작성하려면 확장자가 JS 인 파일을 이용합니다.
- 외부 파일로 자바스크립트를 HTML 파일과 별도로 작성하게 되면 HTML 태그는 HTML 파일내에, 자바스크립트 코드는 JS 파일내에 위치하게 됨으로 관리적인 측면에서 효율적일 수 있습니다.
- 또한 자바스크립트가 JS 파일로 작성하게 되면 동일한 자바스크립트 코드를 이용해야 하는 여러 HTML 파일내에서 코드 중복 없이 하나의 JS 파일을 재사용할 수 있는 이점이 있습니다.



```
test21
├── test.html
└── JS test.js
```

자바스크립트 코드 위치 – HTML 문서 외부에 작성

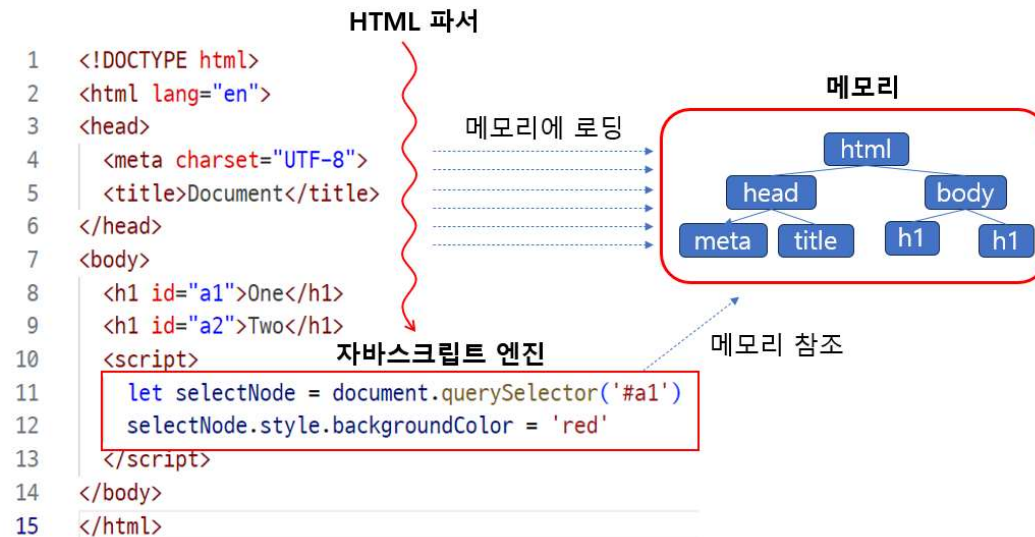
- `<script src="test.js"> </script>` 처럼 HTML 파일에서 자바스크립트 파일을 이용하겠다고 선언하면 `src`에 명시한 파일의 자바스크립트 코드가 `<script> </script>` 태그 사이에 있는 것과 동일한 효과가 납니다.

```
const myButtonClick = () => {  
  let title = document.getElementById('title')  
  if(title.style.backgroundColor === 'yellow'){  
    title.style.backgroundColor = 'pink'  
  }else {  
    title.style.backgroundColor = 'yellow'  
  }  
}
```

```
<script src="test.js"> </script>
```

자바스크립트 코드 위치 - HTML 실행 순서

- 자바스크립트 코드가 작성되는 HTML 문서내에서의 위치가 문제가 될 수 있습니다.
- 자바스크립트 코드의 내용이 실행되는 HTML 문서의 태그와 관련이 없다면, 즉 자바스크립트 코드에서 HTML 문서의 DOM 노드를 이용하지 않는다면 어느 위치에 작성되어 있든 상관은 없습니다.
- 자바스크립트 코드에서 DOM 노드를 이용한다면 이용하는 DOM 노드가 메모리에 로딩된 이후에 해당 자바스크립트 코드가 실행되게 해주어야 합니다.



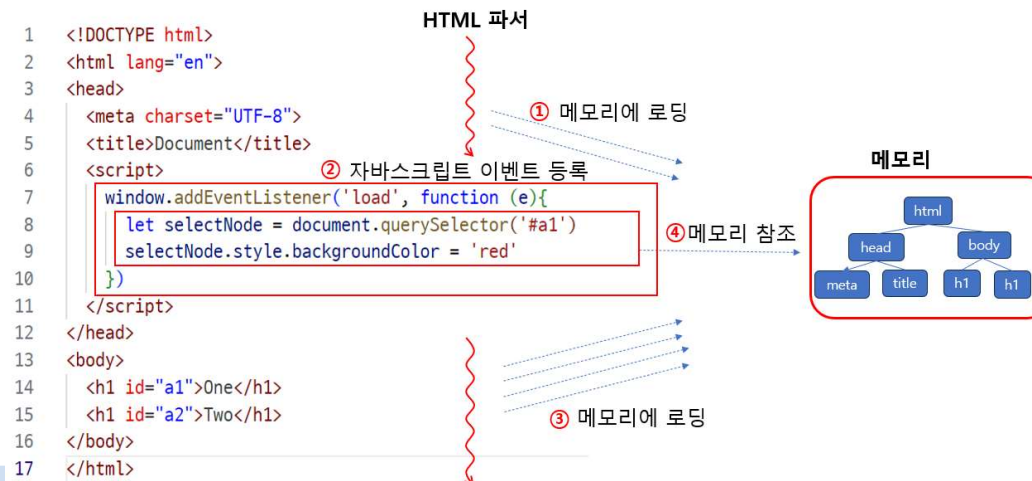
자바스크립트 코드 위치 - HTML 실행 순서

- Uncaught TypeError: Cannot read properties of null (reading 'style') 라는 자바스크립트 오류가 발생 가능성



자바스크립트 코드 위치 - HTML 실행 순서

- DOM 노드를 이용하는 자바스크립트 코드는 HTML 파서에 의해 이용하고자 하는 DOM 노드가 메모리에 로딩된 후에 사용해야 합니다.
- 이를 위해 자바스크립트의 코드 위치를 이용하고자 하는 DOM 노드의 태그 아랫부분에 작성하거나 문서 로딩이 완료된 후 자바스크립트 실행되게 해주어야 합니다.
- 브라우저에서는 어떤 HTML 문서를 분석하여 그 문서의 모든 태그를 메모리에 로딩하게 되면 로딩 완료 이벤트를 발생시키게 됩니다.



자바스크립트 주석

- 주석은 소프트웨어 언어로 개발되는 코드내에 추가되는 구문으로 애플리케이션을 실행시키는 플랫폼에 의해 처리되는 구문이 아닌 코드를 작성한 사람을 위한 구문을 의미합니다.
- 애플리케이션 개발 생산성 및 유지보수성을 위해 추가하는 코드
- 자바스크립트에서는 주석을 추가하는 방법이 2가지 입니다.
- // 로 추가되는 주석은 한 줄 주석이며 /* */ 로 추가되는 주석은 여러 줄 주석입니다.

```
//여기도 한줄 주석입니다.  
let sum = 0  
for(i=1; i<11; i++){  
    sum += i  
}  
/*  
    여기는 여러줄 주석입니다.  
    여기는 여러줄 주석입니다.  
*/
```

자바스크립트 주석

- HTML 문서에는 HTML 파서가 해석해서 처리하는 태그가 있고 CSS 엔진이 처리하는 CSS 구문이 작성되어 있을 수 있고, 자바스크립트 엔진에 의해 처리되는 자바스크립트 코드가 작성되어 있을 수 있습니다.
- HTML 태그 CSS 자바스크립트 세 가지의 구성 요소가 같이 작성 될 수 있는데 구성 요소마다 주석이 다 다르죠.
- 자바스크립트 주석은 위에서 살펴보았듯이 한 줄 주석은 `//` 이며 여러 줄 주석은 `/* */` 입니다. CSS 주석은 `/* */` 입니다. 그리고 HTML 주석은 `<!-- -->` 입니다.

자바스크립트 기초

자바스크립트는 대소문자를 구분한다.

- 자바스크립트는 영문자 대소문자를 구분합니다.
- 그럼으로 hello 와 Hello, HELLO 는 모두 다른 데이터입니다.

Cannot redeclare block-scoped variable 'data'. javascript

let data: string

문제 보기 (<Alt>+F8) 빠른 수정을 사용할 수 없음

```
let data = 'hello'
let data = 'world'
```

```
let data = 'hello'
```

```
let Data = 'world'
```

자바스크립트 기초

라인을 구분하기 위한 세미콜론을 강제하지 않는다.

- 자바 혹은 C 같은 언어에서는 라인을 구분하기 위한 세미콜론이 강제사항이어서 사용하지 않으면 에러가 발생합니다.
- 하지만 자바스크립트는 코드의 라인을 구분하기 위해서 세미콜론을 강제하지 않습니다.
- 강제하지 않는다는 의미는 써도 되고 쓰지 않아도 됩니다.

세미콜론으로 라인을 구분한 코드

```
let data = 10;  
let result = data * 10;  
console.log(result);
```

세미콜론을 사용하지 않은 코드

```
let data = 10  
let result = data * 10  
console.log(result)
```

자바스크립트 기초

window 함수 – alert prompt confirm

- 브라우저에서 실행되는 자바스크립트 코드에서 기본이 되는 객체는 window 입니다.
- window 는 브라우저 자체를 지칭하는 객체이며 이 window 객체에서 제공하는 변수, 함수를 이용해 프로그램을 작성할 수 있습니다.
- window 객체에서 다이얼로그를 띄우기 위한 alert(), confirm(), prompt() 함수를 제공합니다.

window 의 다이얼로그 함수			
함수명	용도	버튼	결과 값
alert()	간단한 메시지 출력	확인	undefined
confirm()	확인/취소 선택	확인/취소	true/false
prompt()	사용자 글 입력	확인/취소	입력값/ null

자바스크립트 기초

document.write()

- document 객체는 브라우저에서 실행되는 HTML 문서 자체를 지칭하는 객체입니다.
- document.write() 함수는 HTML 문서에 어떤 문자열을 출력하기 위해서 사용되는 함수입니다.

HTML 태그가 포함된 문자열 출력

```
document.write('<div>안녕하세요 <b style="color:blue;">' + name + '</b>님, 환영합니다.</div>')
```


자바스크립트 기초

console.log()

- 자바스크립트 코드가 실행되면서 발생한 어떤 결과 혹은 문자열을 출력하기 위한 구문인데 출력하는 위치가 자바스크립트 코드를 실행시켜주는 플랫폼의 콘솔입니다.
- 크롬 브라우저라면 개발자도구에 있는 콘솔창에 출력되게 됩니다. 또는 Node.js 에서 실행되는 자바스크립트라면 Node.js 의 콘솔창에 출력되게 됩니다.

```
console.log() 테스트  
console.log('첫번째 출력입니다. - 안녕하세요')  
console.log('두번째 출력입니다.', "홍길동", "임꺽정")  
console.log('세번째 출력입니다.', 10, true)
```

자바스크립트 코드 컨벤션

들여쓰기를 해야 한다.

- 코드 컨벤션(Code Convention)이란 프로그래밍 코드를 작성하는 일종의 가이드입니다.
- 코드를 작성할 때 읽고 관리하기 쉽게 작성하기 위한 일종의 관습이라고 보시면 됩니다.
- 코드 컨벤션에서 가장 기본적인 것은 들여쓰기입니다.
- 어떤 실행영역을 묶기 위해서 중괄호({ })을 사용하는데 이 중괄호 부분을 중괄호 밖의 코드보다 들여쓰기를 해서 한눈에 중괄호로 묶인 부분이 어디까지인지를 쉽게 식별할 수 있게 해주어야 합니다.

The diagram illustrates the application of the Shift + Ctrl + F shortcut to format JavaScript code. On the left, the code is shown in a flat style. A blue arrow points to the text 'Shift + Ctrl + F', and another blue arrow points to the right, where the same code is shown with proper indentation.

```
1  const calSum = () => {
2  let sum = 0
3  for(i=1;i<=10;i++){
4  sum += i
5  console.log(sum)
6  }
7  }
8  calSum()
```

→ Shift + Ctrl + F →

```
1  const calSum = () => {
2      let sum = 0
3      for(i=1;i<=10;i++){
4          sum += i
5          console.log(sum)
6      }
7  }
8  calSum()
```

자바스크립트 코드 컨벤션

식별자에 대소문자를 구분해서 사용한다.

- 코드를 작성하면서 변수명, 함수명, 클래스명 등 개발자가 직접 식별을 위해 이름을 지정하는 경우가 있습니다.
- 이때 대소문자를 구분해서 지정해 주어야 합니다.
- 변수명은 소문자로 시작하는 명사형을 이용한다.
- 함수명은 소문자로 시작하는 동사형을 이용한다.
- 클래스명은 대문자로 시작하는 명사형을 이용한다.
- 두 단어를 결합시켜서 이름을 주는 경우에는 뒷 단어의 첫 글자를 대문자로 지정한다.

자바스크립트 코드 컨벤션

식별자에 대소문자를 구분해서 사용한다.

- 뒤 단어를 대문자로 해서 식별자를 지정하는 것을 camel case 라고 합니다.
- 두 단어를 언더스코어(_) 로 연결해서 지정할 수도 있는데 이를 snake case 라고 합니다.
- 자바스크립트 포함 대부분의 소프트웨어 언어에서는 대소문자를 구분하며 이렇게 대소문자를 구분하는 언어에서는 관습상 camel case 를 주로 사용합니다.
- 만약 어떤 소프트웨어 언어가 대소문자를 구분하지 않는 경우에는 snake case 를 관습상 사용합니다.

자바스크립트 코드 컨벤션

스페이스를 적절히 사용하여 보기 좋게 작성

- 스페이스를 추가하지 않았다고 에러가 발생하지는 않지만 코드 해석의 용이성을 위해 적절하게 스페이스를 추가하는 것이 좋습니다.

스페이스를 추가하지 않은 코드

```
1  const calSum={()=>{
2    let sum=0
3    for(i=1;i<=10;i++){
4      sum+=i
5    }
6    console.log(sum)
7  }
```

스페이스를 추가한 코드

```
1  const calSum = () => {
2    let sum = 0
3    for(i = 1; i <= 10; i++){
4      sum += i
5    }
6    console.log(sum)
7  }
```

자바스크립트 코드 컨벤션

주석을 열심히 추가하자.

- 코드에 주석을 추가하지 않아도 애플리케이션은 작성되고 잘 실행됩니다.
- 하지만 적절한 주석을 추가하지 않은 코드는 나중에 그 코드가 왜 무엇을 위해 작성되었는지를 간파하기 어려워 전체 코드를 해석하기 어려워 집니다.

자바스크립트 코드 컨벤션

의미있는 식별자를 사용하자.

- 식별자는 변수명, 함수명, 클래스명등을 지칭합니다.
- 이름만 보고도 이 변수가 어떤 데이터가 저장되는 변수인지, 이 함수가 어떤 행위를 하는 함수인지를 간파할 수 있게 지정하는 것이 좋습니다.

의미 없는 단어로 식별자를 지정한 경우
<code>b = 10</code> <code>a()</code>

의미 있는 단어로 식별자를 지정한 경우
<code>age = 20</code> <code>calSum()</code>

자바스크립트 코드 컨벤션

구글 가이드

- 구글에서 자바스크립트 프로그래밍의 작성 스타일을 가이드한 문서(Google Javascript Style Guide - <https://google.github.io/styleguide/jsguide.html#naming-camel-case-defined>) 을 참조

디버깅

디버깅도 실력이다.

- 우리가 작성한 애플리케이션은 에러가 발생할 수 있습니다.
- 에러는 크게 컴파일 에러와 런타임 에러로 구분할 수 있습니다.
- 컴파일 에러는 소프트웨어 언어의 문법에 맞게 작성하지 않은 에러이며 런타임 에러는 문법에는 맞게 작성했지만 실행하다가 발생한 에러입니다.

디버깅

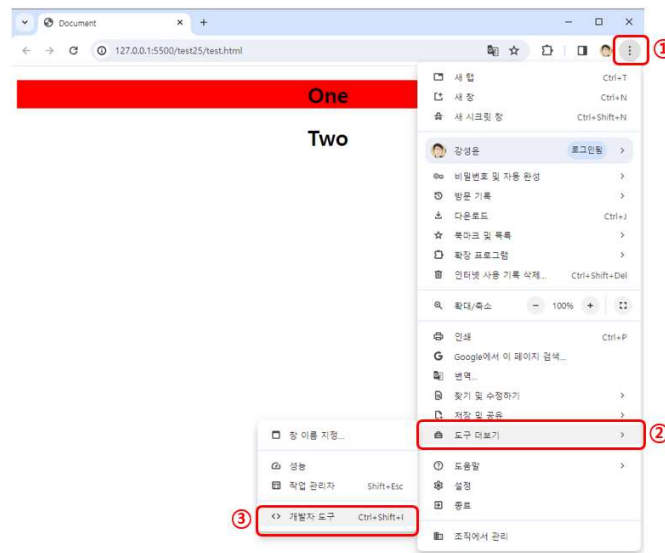
디버깅도 실력이다.

- 컴파일 에러는 문법에 맞지 않게 작성한 에러임으로 대부분의 IDE 에서 무엇이 잘못 된 것인지를 알려줍니다.
- 런타임 에러는 코드가 문법에 맞게 작성되었지만 실행할 수 없는 상태입니다.
- 아무리 코드를 잘 작성하더라도 런타임 에러는 발생할 수 있습니다.
- 런타임 에러도 애플리케이션을 실행시키는 플랫폼에 에러 메시지가 나옵니다.
- 프론트 웹 애플리케이션이라면 브라우저 콘솔창에 에러가 출력됩니다. 이 에러 메시지를 확인해서 디버깅을 하면 됩니다.

디버깅

크롬 브라우저 개발자 도구 이용하기

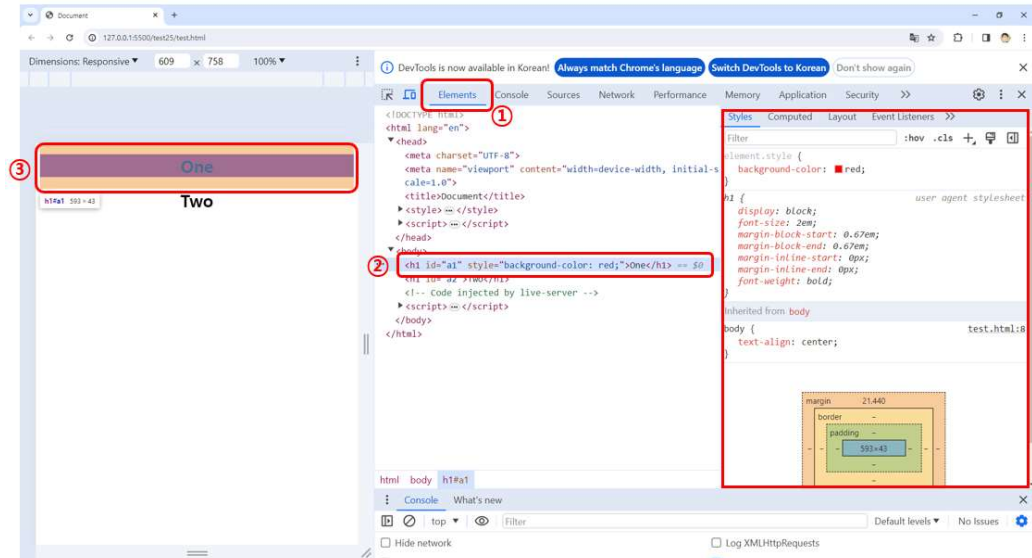
- '메뉴 -> 도구 더보기 -> 개발자 도구'



디버깅

크롬 브라우저 개발자 도구 사용하기 - Element

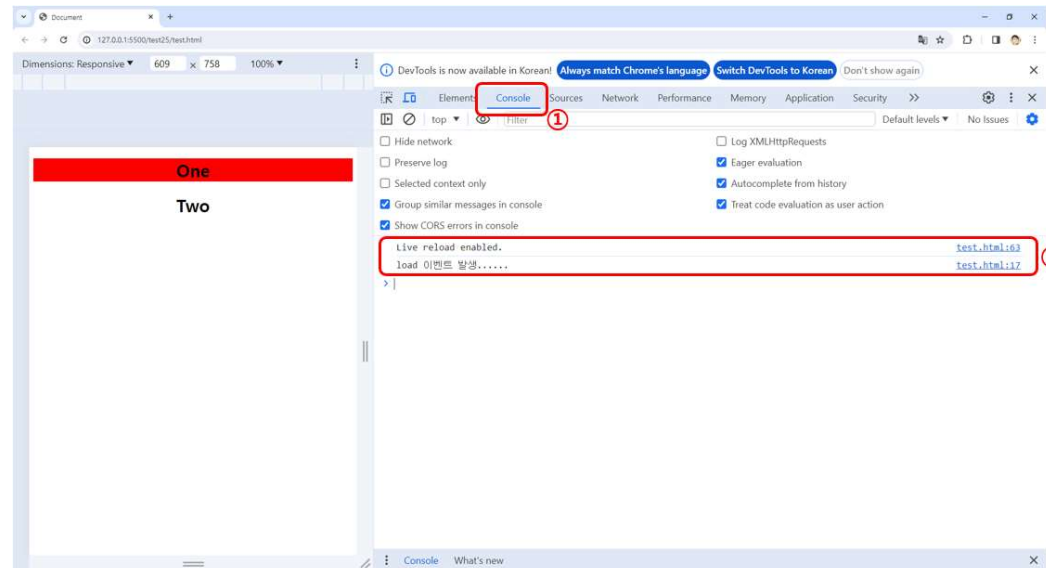
- 'Element' 탭에서 브라우저에 출력되는 HTML 문서의 태그들과 태그에 적용된 CSS 스타일을 살펴 볼 수 있습니다.



디버깅

크롬 브라우저 개발자 도구 이용하기 - Console

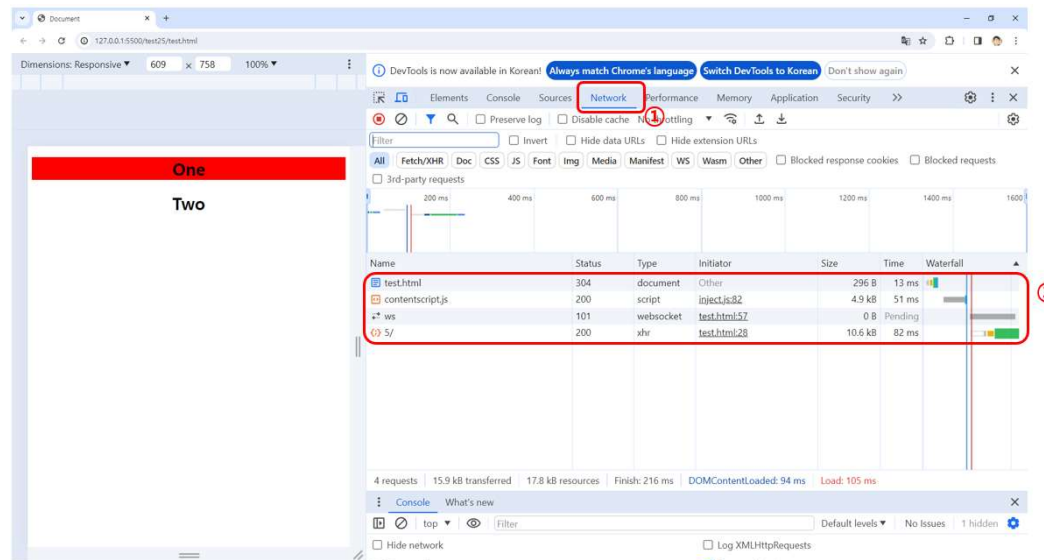
- 자바스크립트 코드에서 `console.log()` 로 출력시킨 내용
- 자바스크립트 코드가 실행되다가 발생한 에러 로그



디버깅

크롬 브라우저 개발자 도구 이용하기 - Network

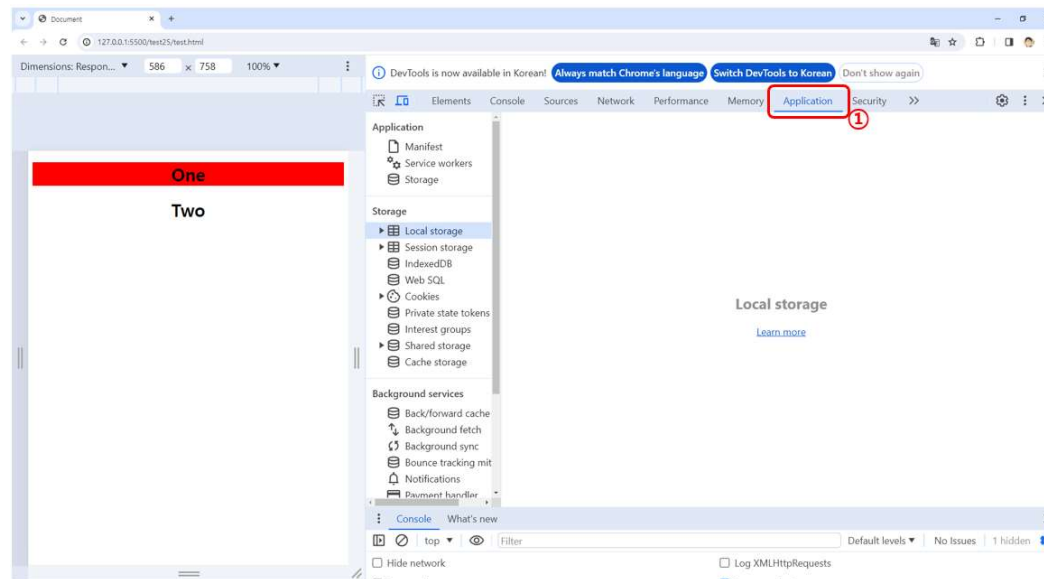
- Network 탭을 클릭하면 HTML 문서를 브라우저에 출력하기 위해서 발생한 네트워킹 이력을 볼 수 있습니다.



디버깅

크롬 브라우저 개발자 도구 사용하기 - Application

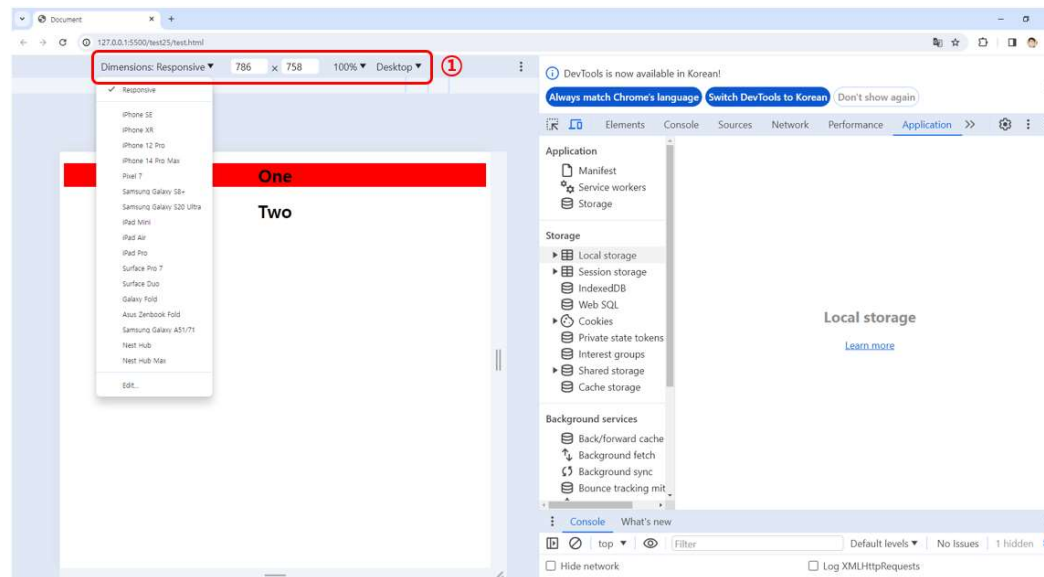
- Application 탭은 HTML 문서를 실행시키면서 애플리케이션 적인 다양한 정보를 제공



디버깅

크롬 브라우저 개발자 도구 이용하기 - Responsive

- 다양한 사이즈의 브라우저에서 우리가 만든 HTML 문서가 어떻게 출력될지 테스트를 해주어야 하는데 이부분을 개발자 도구에서 조정하면서 테스트 할 수 있습니다.





감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare