

## 13 자바스크립트와의 첫 만남

## 13-1 자바스크립트로 무엇을 할까

## ■ 웹 요소 제어

- 웹 요소를 가져와서 필요에 따라 스타일을 변경하거나 움직이게 할 수 있음
  - 웹 사이트 UI 부분에 많이 활용
  - 사용 예
    - : 마우스 포인터를 올렸을 때 펼쳐지는 메뉴
    - : 한 화면에서 탭을 눌러 내용만 바뀌도록 하는 콘텐츠
    - : <form> 요소에 입력된 정보 형식 체크
    - : 마우스의 클릭으로 동작하는 루틴

## ■ 웹 애플리케이션 작성

- 사용자와 실시간으로 정보를 주고 받으며 애플리케이션처럼 동작
  - 사용 예
    - : 온라인 지도의 길 찾기 서비스,
    - : 데이터 시각화 서비스 (공개된 API를 활용한 다양한 서비스\_

## ■ 다양한 라이브러리 사용

- 웹 중심 서비스가 늘어나면서 브라우저에서 처리해야 할 일이 증가
  - 라이브러리와 프레임워크가 계속 등장하고 있음
  - 프레임 워크
    - : 웹 애플리케이션 개발을 위한 React, Angular, Vue 등
  - 라이브러리
    - : 시각화를 위한 d3.js 등
    - : DOM 조작을 위한 jQuery 등

## ■ 백엔드 개발

- 자바스크립트를 이용한 백엔드 프로그램 개발
  - node.js 프레임 워크

## ■ 웹 어플리케이션 개발에서 자바스크립트의 특징

- 클라이언트 스크립트 언어 (Client Script language) 로 브라우저에 의해 해석되고 실행
  - 서버가 아닌 사용자의 컴퓨터에서 실행되는 언어  
HTML 문서 내에서 CSS와 함께 사용
- 인터프리터(interpreter) 언어
  - 코드가 작성된 순서대로 순차적으로 해석되어 실행됨
  - 코드에 에러가 발생할 경우, 에러 발생 다음 라인부터는 구문분석을 하지 않음  
디버깅에 어려움이 있음
- 객체를 기반으로 하는 언어
  - 다양한 종류의 객체들과 그 객체들에 대한 다양한 메서드와 속성을 제공
  - 완전한 형태의 객체지향 언어는 아님  
제한된 캡슐화와 상속 기능 중에서도 일부만 지원

- C언어 기반 프로그래밍 언어
  - 객체지향의 일부 기능을 지원하나 C언어를 기반으로 프로그래밍 수행  
C언어의 기본적인 개념만 있으면 배우기 쉬움
- 공개된 소프트웨어(Open Software)
  - HTML이나 CSS와 함께 사용되는 공개 소프트웨어  
HTML과 CSS를 사용해 개발된 정적 웹페이지에 동적인 기능을 제공
  - 다양한 라이브러리나 프레임워크 사용 가능

## 13-2 웹브라우저가 자바스크립트를 만났을 때

## ■ 자바스크립트 선언

- 자바스크립트는 웹 스크립트 언어이기 때문에 HTML 문서 내에서 html 코드와 함께 표현됨
- 자바스크립트 선언은 자바스크립트 코드가 작성되는 영역을 선언하는 것을 말함

## ■ 선언 방법

- `<script>` 태그를 이용한 선언
  - `<script>` 태그와 `</script>` 사이를 사용해 HTML 문서 내에 선언하는 방법
- 외부 스크립트 파일로 선언
  - 외부 파일로 자바스크립트를 선언하고 HTML 문서에 연결하는 방법
- 인라인 선언
  - HTML의 특정 요소에만 적용되는 자바스크립트를 선언하기 위해 요소 내부에 선언하는 방법

## ■ <script> 태그를 사용하는 선언

- <script> 태그와 </script> 태그 사이에 자바스크립트를 선언
  - HTML 문서의 모든 위치에서 선언될 수 있음  
선언된 위치에서 실행됨
  - 주로 <head> 태그 내에 선언
  - 현재 문서 전체에 적용되는 자바스크립트 코드를 작성하는데 사용

```
<script>
```

자바스크립트 코드;

```
</script>
```

<> script-embed.html X

chapter13 > <> script-embed.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <script>
6     alert("안녕하세요");
7   </script>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

이 페이지 내용:

안녕하세요

확인



## ■ 외부 스크립트 파일로 선언

- 자바스크립트 선언을 외부에서 별도의 파일로 선언하고, HTML 문서에 연결해서 사용하는 방식

- 선언문을 포함하는 외부의 파일은 js라는 확장자를 가짐  
외부로 선언할 경우, 외부 선언 파일 내에는 <script> 태그를 사용하지 않음
- HTML 문서에서 <script> 태그의 src 속성을 사용해 연결

```
<script src="자바스크립트 파일"> </script>
```

- 외부 선언을 사용하는 이유

- 보안상의 이유  
HTML 코드와 자바스크립트 코드는 브라우저에서 [소스보기]를 실행하면 모두 출력됨  
자바스크립트 코드를 외부에서 선언하면 소스 코드가 출력되지 않음
- 파일의 공유  
HTML 문서 내에 선언된 자바스크립트 코드는 그 문서에만 영향을 미침  
외부에 별도로 선언하고 문서에 연결해 사용하면 여러 문서에서 자바스크립트 코드를 공유할 수 있음

```
chapter13 > JS script-external-sub.js
```

```
1  alert("안녕하세요");
```

```
chapter13 > <> script-external.html > ...
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <script src="script-external-sub.js"></script>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

이 페이지 내용:

안녕하세요

확인

## ■ 인라인 선언

- HTML의 특정 요소에만 적용되는 자바스크립트를 선언하기 위해 요소 내부에 선언하는 방법

- 한 라인의 간단한 자바스크립트 코드의 실행을 위해 사용
- 주로 이벤트와 함께 사용

지정한 이벤트가 발생할 때 적용한 자바스크립트가 실행됨

```
chapter13 > <> script-inline.html > ...
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  </head>
6  <body onload="alert('안녕하세요')">
7
8  </body>
9  </html>
```

이 페이지 내용:

안녕하세요

확인

## 13-3 자바스크립트의 용어와 기본 입출력 방법

## ■ 식과 문

- 자바스크립트의 구문은 식(expression)과 문(statement)으로 구성됨
- 식(expression)
  - 표현식 이라고도 하며 연산자가 포함된 연산식, 숫자나 문자로 구성된 값을 의미
  - 식은 변수에 저장됨
- 문(statement)
  - 브라우저에 의해 실행되는 명령어
  - 문들은 세미콜론(;)을 사용해 구분함
  - 자바스크립트의 함수와 메서드, 명령어 등이 해당함

## ■ 자바스크립트의 입출력

- 입력
  - 문서의 실행 도중 키보드로부터 데이터를 받아 들이는 것
- 출력
  - 수행의 중간 결과, 최종 결과, 알림이나 경고 등을 출력하는 방법
  - 윈도우를 이용해 출력하는 방법
  - 브라우저에 출력하는 방법
  - 개발 도구의 콘솔에 출력하는 방법

## ■ alert()

- 문자열과 함께 [확인] 버튼을 가진 대화상자를 출력

```
alert("대화상자에 출력할 문자열")
```

- [확인] 버튼을 클릭할 때까지 이하의 구문을 실행되지 않고 대기함
- 특정 메시지를 전달하거나 오류의 내용을 통지할 때 사용

```
chapter13 > <> alert.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  </head>
6  <body>
7  |   <script>
8  |   |   alert("안녕하세요?");
9  |   </script>
10 </body>
11 </html>
```

이 페이지 내용:

안녕하세요?

확인

## ■ confirm()

- 문자열과 함께 [확인] 버튼과 [취소] 버튼을 가진 대화상자 출력

변수 = `confirm`("대화상자에 출력할 문자열")

- [확인]이나 [취소] 버튼을 클릭할 때까지 이하의 구문은 실행되지 않고 대기
- `confirm()`은 항상 특정 변수와 함께 사용
  - [확인] 버튼을 클릭하면 해당 변수에 TRUE가 전달
  - [취소] 버튼을 클릭하면 해당 변수에 FALSE가 전달
- 주로 IF 문과 함께 사용

```
chapter13 > <> confirm.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4  |   <meta charset="UTF-8">
5  </head>
6  <body>
7  |   <script>
8  |       var reply = confirm("배경 이미지를 바꾸겠습니까?");
9  |   </script>
10 </body>
11 </html>
```

이 페이지 내용:

배경 이미지를 바꾸겠습니까?

확인

취소



## ■ prompt()

- 사용자로부터 숫자나 문자(열)를 받아 들이기 위해 사용

변수 = `prompt`("출력할 문자열" [,기본값])

- 한 줄 입력 상자, 문자열, [확인] 버튼, [취소] 버튼을 가진 대화상자를 출력
- [기본값]은 입력 상자에 출력될 기본 값을 의미
- [확인]이나 [취소] 버튼을 클릭할 때까지 이하의 구문은 실행되지 않고 대기
  - [확인] 버튼을 클릭하면 입력된 값이 변수에 저장됨
  - [취소] 버튼을 클릭하면 `prompt` 함수의 실행이 종료됨

```
chapter13 > <> prompt.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4  |   <meta charset="UTF-8">
5  </head>
6  <body>
7  |   <script>
8  |       var name=prompt("이름을 입력하세요.", "홍길동");
9  |   </script>
10 </body>
11 </html>
```

이 페이지 내용:

이름을 입력하세요.

홍길동

확인

취소

## ■ write()

- document 객체의 메서드이며, 인자의 내용을 브라우저에 출력

document.write(출력할 내용)

- 인자로 변수, 문자열, 수식 입력 가능  
문자나 문자열을 출력할 경우 인용부호 내에 표현해야 함
- HTML 태그도 사용할 수 있음  
HTML 태그도 인용부호 내에 표현
- 숫자나 변수를 문자열과 함께 출력하면 + 연산자를 사용

korea

어서오세요

60

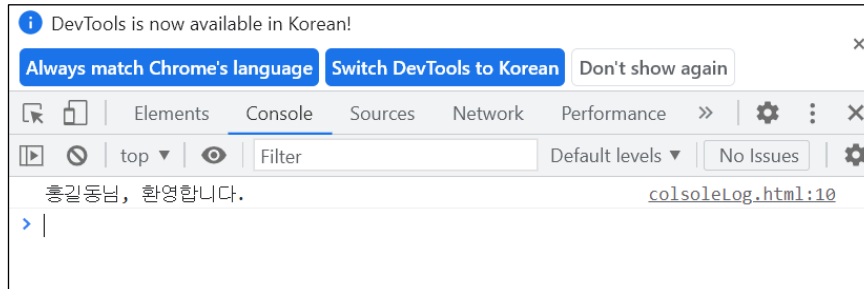
```
chapter13 > <> document-write.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5  </head>
6  <body>
7      <script>
8
9          let nation = "korea";
10
11          document.write(nation + "<br>");
12          document.write("<h2>어서 오세요</h2>");
13          document.write(10+20+30);
14
15      </script>
16  </body>
17  </html>
```

## ■ console.log()

- 인자로 지정한 내용을 콘솔 창에 출력

```
console.log("출력할 문자열")
```

- 콘솔은 브라우저의 개발자 도구에 포함되어 있음



chapter13 > <> colsoleLog.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5  </head>
6  <body>
7    <script>
8
9      var name = prompt("이름을 입력하세요.");
10     console.log(name + "님, 환영합니다.");
11
12   </script>
13 </body>
14 </html>
```

## 13-4 자바스크립트 스타일 가이드

- 자바스크립트 코드는 대소문자를 구분함

```
날짜 객체 생성: New date( ); (X)  
날짜 객체 생성: new Date( ); (O)
```

- 한 라인에 여러 실행 문장을 작성할 경우 각 문장은 세미콜론(;)으로 구분

```
document.write("hi") document.write("bye") (X)  
document.write("hi"); document.write("bye"); (O)
```

- 가독성을 위해 한 라인에 하나의 문장을 입력해야 함

- 문자열 데이터 작성 시 인용 부호의 겹침 오류 주의

- 이중인용부호 내에 단일인용부호 사용 가능/ 단일인용부호 내에 이중인용부호 사용 가능

```
document.write("대한민국의 수도는 '서울' 입니다");  
document.write('대한민국의 수도는 "서울" 입니다');
```

- 단일인용부호내에 단일이용부호를 사용하거나 이중인용부호내에 이중인용부호를 사용할 경우 역슬래시 문자를 사용해야 함

```
document.write("대한민국의 수도는 ₩"서울₩" 입니다");
```

## ■ 자바스크립트의 주석문

### • 한 라인 주석 처리

- // 기호를 사용

```
document.write("TEST")           // 한라인 주문
```

// 기호 다음의 문자열을 주석으로 간주함

### • 다중 라인 주석 처리

- /\* 기호와 \*/ 기호를 사용

```
/*  
첫 번째 주석 문장입니다.  
두 번째 주석 문장입니다.  
*/
```

/\*기호와 \*/ 기호 사이의 모든 문자열을 주석으로 간주함

```
chapter13 > <> comment.html > ...  
1  <!DOCTYPE html>  
2  <html lang="ko">  
3  <head>  
4  |   <meta charset="UTF-8">  
5  </head>  
6  <body>  
7  |   <script>  
8  |   |  
9  |   |   /*  
10 |   |   |   첫 번째 주석문입니다.  
11 |   |   |   두 번째 주석문입니다.  
12 |   |   */  
13 |   |  
14 |   |   document.write("<h2>주석문 연습</h2>"); // 단인라인 주석  
15 |   |  
16 |   </script>  
17 |  
18 |   <!-- 여기는 HTML 주석입니다.-->  
19 |  
20 </body>  
21 </html>
```

수고하셨습니다