

JavaScript

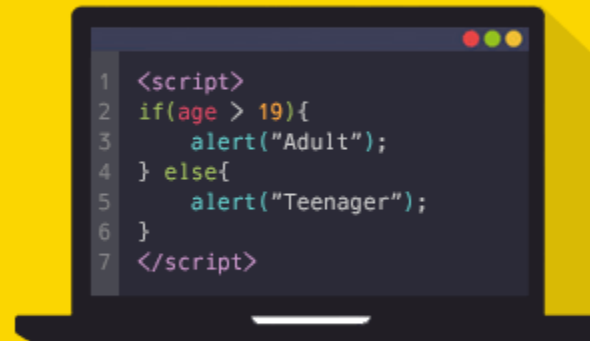
자바스크립트 개요 – 김근형 강사

JavaScript 개요

- JavaScript



JavaScript



JavaScript 개요

○ JavaScript

- 브랜든 아이크가 10일만에 설계한 것으로부터 시작
- 처음에는 Mocha라는 이름이었지만 4달 만에 LiveScript라는 이름으로 개명하고 다시 3달 후에는 JavaScript라는 이름이 됨.
- JS는 본래 넷스케이프 서버에서 애플리케이션을 제작하기 위한 고수준 추상화 언어로 설계되어 LiveScript라는 이름으로 네비게이터에 포함되었다.
- 그러나 JS는 당시 기준에서 무리한 추상화를 시도했기 때문에 성능 문제가 많았다.
- 여기에 더해서 그나마 클라이언트용 JS 엔진에 있던 시스템 자원 접근용 API들이 보안사고의 원인이 되면서 삭제되는 과정에서, 별다른 보완 방법을 제시하지 못하는 등 한계가 여신했다.
- 마이크로소프트에서는 JavaScript를 표준으로 채택함과 동시에 자사만의 문법을 끼워 넣어 JScript를 탄생.
- 제 1차 브라우저 전쟁에서 MS는 OS 점유율을 늘려감에 따라 사용자들을 IE를 사용하도록 지속적으로 유도하였고, 이에 버티지 못한 넷스케이프는 망함.

JavaScript 개요

○ JavaScript

- JavaScript를 단일화한 움직임은 이미 97년 ECMA TC39라는 문서를 통해 시작이 되었지만, 각 벤더가 비협조적으로 나옴.
- 두 번째 브라우저 전쟁이 발발한 후 구글한테 처참하게 진 MS가 꺾이고 나서야 ECMAScript(이하 ES) 5가 제정되고 표준 문제가 '다소' 해결됨.
- 한편 그렇게 표준화를 거친 JS는 AJAX, jQuery 등의 등장으로 거침없는 발전을 보였고, 기어이 Node.js의 등장으로 Server side 언어로서 탈바꿈하게 된다.
- JIT 컴파일 방식을 도입한 구글의 V8이라는 JavaScript 엔진을 개발하였고, ES6 구현율이 당시 80% ~ 현재 99%에 달하면서 CommonJS, AMD 진영 등의 등장도 나타났다.

JavaScript 개요

○ JavaScript

- 최근 들어 JavaScript는 진입 장벽이 타 언어에 비해 낮은 편으로 인식되고 있다.
- 해외는 front-end 전담 개발자가 있을 만큼 전문화되어있는 반면, 국내에서는 수준이 현격히 낮은 만큼 일반인도 쉽게 접근하고 배울 수 있다.
- 구글 등을 비롯한 여러 벤더 사에서는 이 언어를 활용한 다양한 플랫폼 환경을 지원하고 있으며, Chrome Extension이나 App들이 그 좋은 예이다.
- 단, 동적 타입 언어인 만큼 예상과 다른 동작을 하는 경우도 많고, 콜백과 비동기 처리는 초보자 입장에서 코드 리딩이 굉장히 어렵다.
- 원래 JavaScript 엔진들은 모두 실시간 인터프리팅을 하고 있었는데, 모질라에서 SpiderMonkey 엔진에 JIT 컴파일 방식을 도입했다.
- 이는 JavaScript 엔진으로는 최초로 도입한 것이고, 이 때 알려진 것으로는 순수 JavaScript 성능만 기존 버전의 20~40배에 달했다.
- 그리고 구글 역시 V8이라는 JavaScript 엔진을 개발하면서 JIT 컴파일 방식을 도입하였고, JavaScript 성능이 비약적으로 향상하여 지금은 JavaScript 3D게임 엔진도 개발되고 있다.

JavaScript 개요

○ JavaScript 특징

- JavaScript는 멀티-패러다임 언어로 명령형, 함수형, 객체지향형 언어다. 기본적으로는 함수형 프로그래밍 패러다임을 따른다.
- JS는 동적 타이핑, 약타입 언어고, 간단한 문법에 코딩 방법이 비교적 유연하기 때문에 초기 진입장벽이 거의 없어서 쉽다고 이야기 되지만, 깊이 들어가 보면 매우 특이한 언어다.
- 프로그래밍 언어로서 상당히 우수한 설계를 자랑한다. 간편하면서도 강력한 텍스트 표기법을 가졌으며, 구조적으로 비동기 프로그래밍에 유리하다.
- 긴 코드를 짜보면 의외로 골치 아프다. (호이스팅, 코루틴, 스레드 처리 등등...)
- 대표적인 라이브러리로는 jQuery가 있다. 2019년 기준으로 jQuery 보다는 순수 자바스크립트를 운용하는 것을 더욱 선호하는 편이다. 또한 오늘날 JavaScript는 구글의 V8 JS Engine 버프를 받아 하루가 다르게 발전하고 있다.

JavaScript 개요

- JavaScript 선언 방법
 - html 소스 안에 <script> 태그 안에 선언하는 방법
 - 외부의 .js 파일을 html에서 읽어 들이는 방법
 - 이벤트를 통해 불러오는 방식

JavaScript 개요

- html 소스 안에 선언하는 방식
 - 보통 head쪽에는 자바스크립트에 들어가는 외부 라이브러리들을 선언한다.
 - body 쪽에는 이 페이지에 돌아가는 javascript 소스를 선언한다.
 - 외부 라이브러리 선언 자체를 body쪽에도 선언할 수도 있다.

```
<script>  
    // 자바스크립트 코드  
</script>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
    <script type="text/javascript">  
        // 이곳에 자바 스크립트를 쓸 수 있습니다.  
        document.write("<p>head 영역에 선언한 자바 스크립트</p>");  
    </script>  
</head>  
<body>  
  
    <script type="text/javascript">  
        // 보통 자바스크립트는 이곳에서 작성합니다.  
        document.write("<p>body 영역에 선언한 자바 스크립트</p>");  
    </script>  
</body>  
</html>
```


JavaScript 개요

- html 소스 외부에 js 파일 선언
 - .js 파일을 외부에서 선언하고 src를 통해 해당 파일을 html에서 읽어서 선언하는 방식
 - body쪽에 선언할 수도 있다.

```
<script src="js파일 있는 위치"></script>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script type="text/javascript" src="js/example.js"></script>
</head>
<body>

</body>
</html>
```

JavaScript 개요

- html 이벤트를 통해 선언하는 방법
 - 이벤트 안쪽에 자바 스크립트 문장을 넣어 실행할 수 있다.

```
<p onclick = 'javascript 문장'>이곳을 클릭하세요</p>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <p onclick="alert('이벤트를 통해 넣은 자바 스크립트 소스');">
    여기를 클릭하세요
  </p>
</body>
</html>
```

JavaScript 개요

○ 주석문

- 작성한 코드에 영향을 미치지 않는 즉 실행되지 않는 소스
- 주로 소스에 대한 설명을 나타내고자 할 경우 사용한다

```
// 한 줄 주석문  
/* 다중 줄 주석문 */
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Document</title>  
</head>  
<body>  
  <script type="text/javascript">  
    // 이걸 한 줄 주석문입니다.  
    /*  
      이걸 다중 줄 주석문입니다.  
    */  
  </script>  
</body>  
</html>
```

JavaScript 개요

○ 출력 방식

- alert() : 알림창으로 출력한다.
- document.write() : 웹 페이지에 출력한다.(태그 사용 가능)
- console.log() : 로그로 출력.

```
<script type="text/javascript">  
  // 알림 창으로 출력  
  alert('알림 창으로 출력');  
  // 웹 페이지에 출력  
  document.write('<p>웹 페이지에 출력</p>');  
  // 로그로 출력  
  console.log('로그로 출력');  
</script>
```

JavaScript 개요

○ 작성 방법

- 코드는 위에서 아래로, 좌에서 우로 쓰고 읽는다.
- 자바 스크립트는 대,소문자를 구분하여 작성한다.
- 코드 한 줄을 작성한 후에는 세미콜론(;)을 쓴다.
- 코드의 가독성을 위해 한 줄에 한 문장만 작성한다.
- 문자형 데이터를 작성할 때는 큰 따옴표와(" ") 작은 따옴표(' ')의 겹침 오류를 주의한다
- 코드를 작성할 때 중괄호{} 또는 소괄호()의 짝이 맞아야 한다.