웹 프로그래밍

6. 웹과 JavaScript 이해하기

playjs.html

```
<html>
<head>
</head>
<body>
        <header>
                <div id="title">
                        Play with JS!
                </div>
                <div id="ctime">
                </div>
                <nav>
                </nav>
        </header>
        <div id="content">
                Put contents here.
        </div>
        <footer>
                <p>Web Programming, Spring 2018</p>
                <p>Created by Soyoung Park</p>
        </footer>
</body>
</html>
```

playjs.html

- <nav></nav>영역에 메뉴 추가

```
<ul>
		<li><a class="main-menu" href="#">Basic JS</a></li>
		<li><a class="main-menu" href="#">Hangman</a></li>
		<li><a class="main-menu" href="#">jQuery</a></li>
		<li><a class="main-menu" href="#">Advanced JS</a></li>
</ul>
```

playjs.css

```
* {margin: 0; padding: 0;}
body {width:960px;
        margin: 0 auto;}
li {list-style: none;}
a {text-decoration: none;}
img {border: 0;}
header {  width: 900px; height: 80px;
        background-color: #f6f9d4;
        position:fixed;
        top: 0;
        padding: 40px 30px 0 30px;
}
#title {    height:40px;
        text-align: center;
        font-size: 30px;
        font-weight: bold; }
```

```
header > nav {height:40px;
        overflow:hidden;}
#ctime {  width: 250px;
        height:36px;
        float:right;
        border: 1px dashed orange;
        border-radius: 15px;}
.main-menu {
        display:block;
        float:left;
        width:100px;
        height:40px;
        text-align: center;
        line-height: 40px;
        color: black;
        text-decoration:underline;}
```

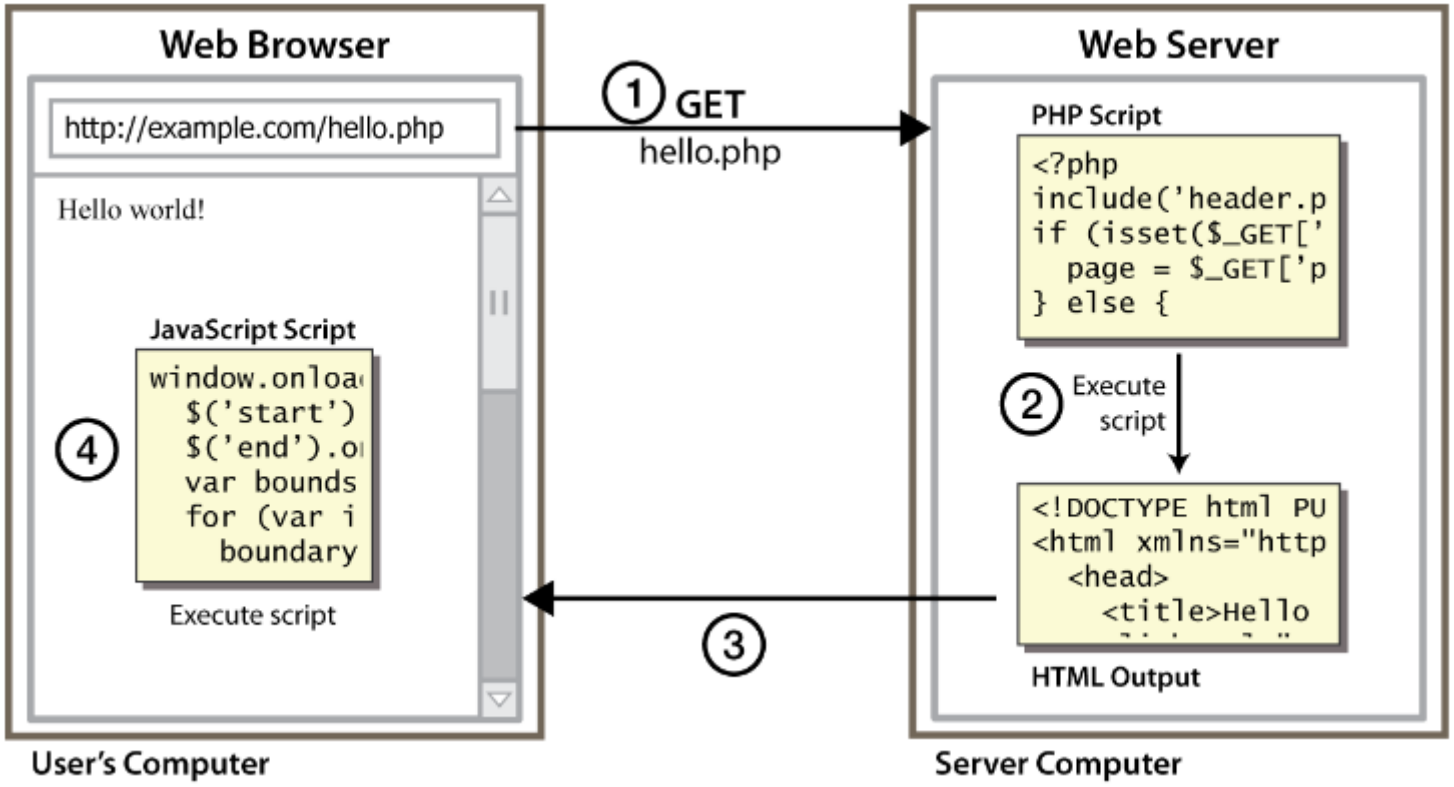
playjs.css

```
#content {
        margin-top:120px;
        padding: 10px; }
footer{
        padding:20px;
        width:920px;
        background-color: #d2f299; }

footer > p {
        font-weight: bold;
        text-align: center;}

.lab {  padding: 10px;
        border: 1px solid black;
        margin-top:10px; }
```

HTML5 기술의 핵심


웹페이지의
내용은
나한테 맡겨!
웹페이지의
표현은
나한테 맡겨!
웹페이지의
동작은
나한테 맡겨!
HTML
CSS
JAVASCRIPT

JS code running in Web browser



SSS vs CSS

- Client-side scripting benefits
  - **Usability**
    - Can modify a page without having to post back to the server (faster UI)
  - **Efficiency**
    - Can make small, quick changes to page without waiting for server
  - **Event-driven**
    - Can respond to user actions like clicks and key press
- Server-side scripting benefits
  - **Security**
    - Has access to server's private data; client can't see source code
  - **Compatibility**
    - Not subject to browser compatibility issues
  - **Power**
    - Can write files, open connections to servers, connect to databases

자바 vs 자바 스크립트

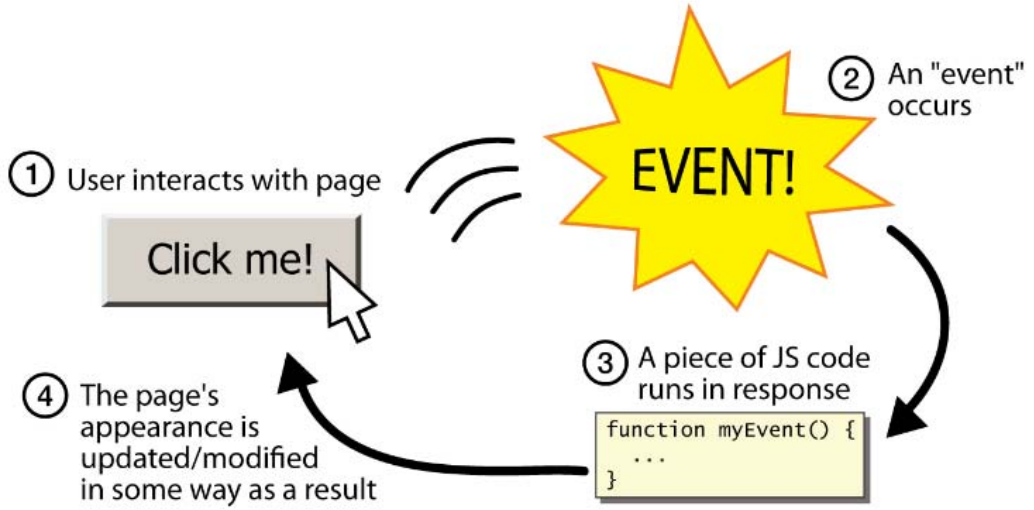
특징	자바 언어	자바스크립트
언어 종류	소스 파일을 컴파일하여 실행하는 컴파일 언어이다.	브라우저가 소스 코드를 직접 해석하여 실행하는 인터프리트 언어이다.
실행 방식	자바 가상 기계 위에서 실행한다.	브라우저 위에서 실행된다.
작성 위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수 선언	변수의 타입을 반드시 선언해야 함	변수의 타입을 선언하지 않아도 사용 가능

Class 기반 객체 지향 언어

객체 지향이긴 하나,
class 개념보다는 function 사용

자바 스크립트의 용도

- 이벤트에 반응하는 동작 구현(Event-driven Programming)
- AJAX(Asynchronous JavaScript and XML)
- HTML 요소들의 크기나 색상을 동적으로 변경
- 게임 및 애니메이션
- 사용자 입력 값 검증

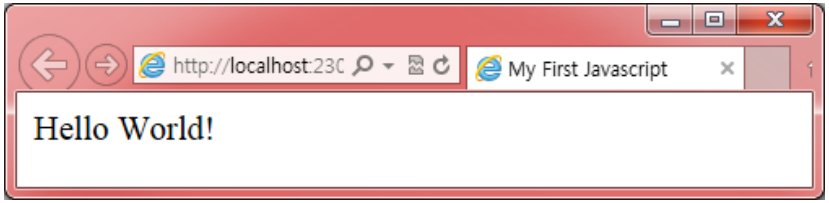


자바 스크립트의 위치

- 내부 자바스크립트
- 외부 자바스크립트
- 인라인 자바스크립트

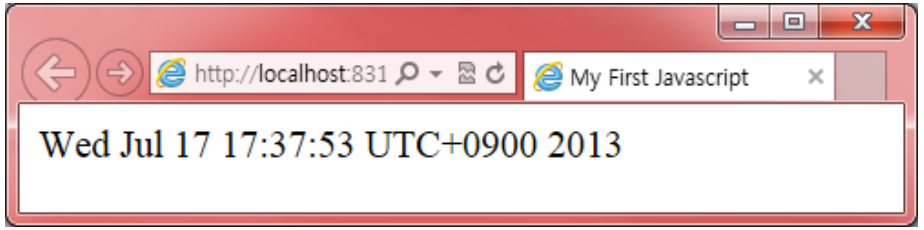
내부 JavaScript

```
<!DOCTYPE HTML>
<html>
<head>

    <title>My First Javascript </title>

    <script>
        document.write("Hello World!");
    </script>

</head>
<body></body>
</html>
```



내부 JavaScript

```
<!DOCTYPE HTML>
<html>
<head>
   <title> My First Javascript </title>
</head>
<body>

   <script>
      var now = new Date();
      document.write(now);
   </script>

</body>

</html>
```

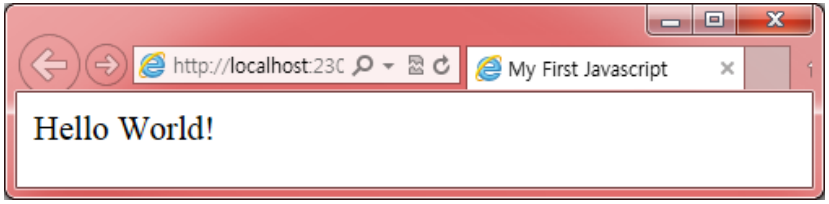


외부 자바스크립트

```
<!DOCTYPE html>
<html>
<head>
    <script src="myscript.js"></script>
</head>
<body>
</body>
</html>
```

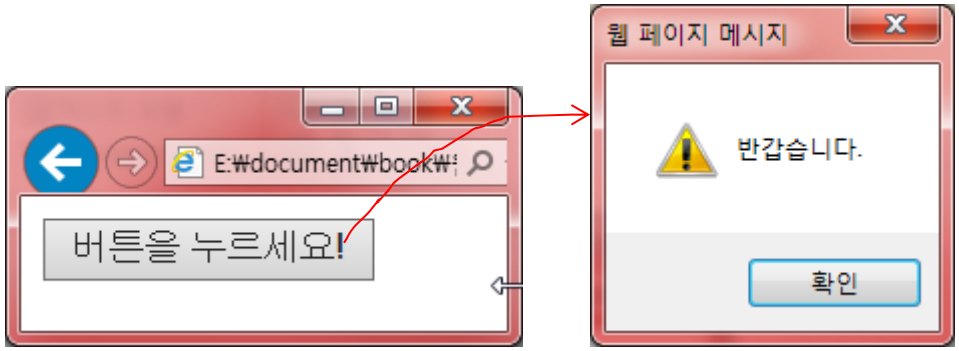
myscript.js

```
document.write("Hello World!");
```



인라인 자바 스크립트

```
<!DOCTYPE html>
<html>
<body>
    <button type="button" onclick="alert('반갑습니다.')">버튼을 누르세요!</button>
</body>
</html>
```



document.write()

- 현재 브라우저에 오픈된 문서에 쓰기

```
<body>
  <hr>
    <H1>JS TEST</H1>
    <script>
        document.write("Hollo~ JS <br>");
        document.write("begin at new line");
    </script>
<body>
```

=

```
<body>
        <hr>
        <H1>JS TEST</H1>
        Hello~ JS<br>
        begin at new line
<body>
```

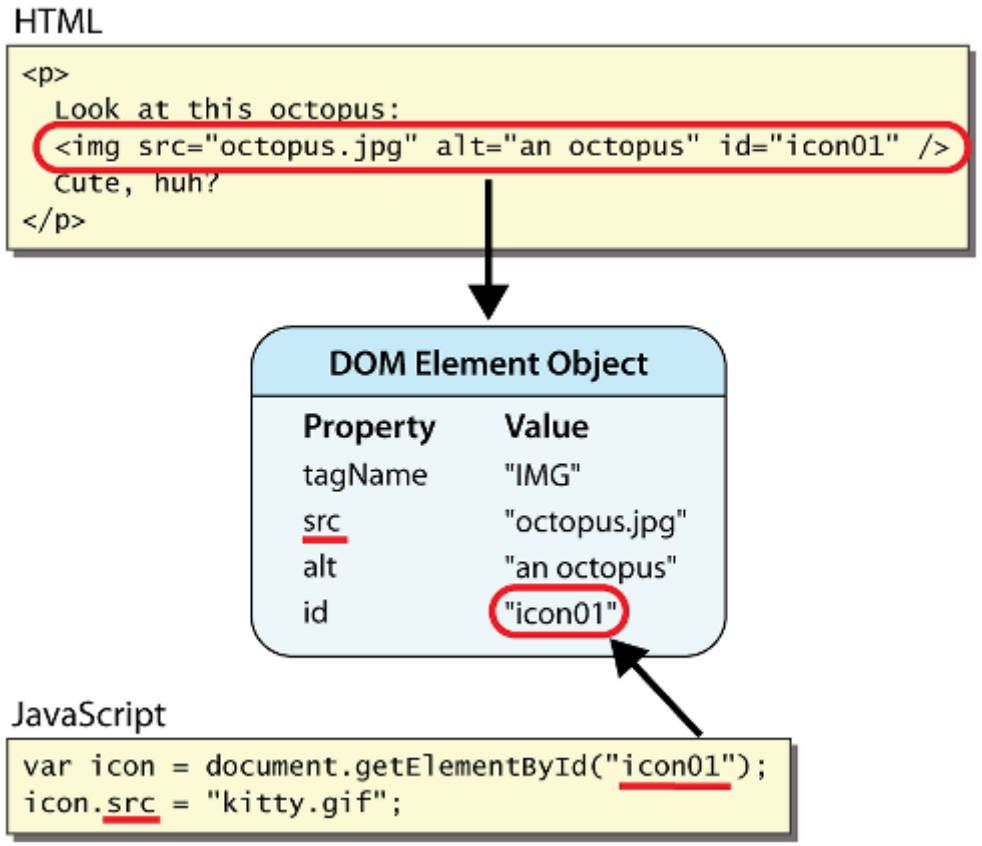
document.write()

- 주의사항
  - 문서가 완전히 로드 된 후,
  - 다시 document.write()를 호출하면 문서의 내용이 새로운 내용으로 완전히 재작성 됨

```
<body>
        <hr>
        <H1>JS TEST</H1>
        <script>
                document.write("Hollo~ JS <br>");
                document.write("begin at new line");
                function func()
                {
                        document.write("rewrite pape");
                }
        </script>
        <button type="button" onclick="func()">click</button>
</body>
```

함수 func()가 실행되면,
이전에 쓰여졌던 내용은
사라지고,
“rewrite page” 문자열만
화면에 보여짐

DOM: Document Object Model



DOM: Document Object Model

```
<button onclick="changeText();">Click me!</button>
<input id="output" type="text" value="replace me" />
function changeText() {
  var textbox = document.getElementById("output");
  textbox.value = "Hello, world!";
}
```

예제

```
<!DOCTYPE html>
<html>
<body>
   <h1 id="test">This is a heading.</h1>
   <script>
      function func() {
         e = document.getElementById("test");
         e.style.color = "red";
      }
  </script>
   <button type="button" onclick="func()">클릭하세요!</button>
</body>
</html>
```