

웹프로그래밍

HTTP

웹프로그래밍

인터넷 : 전세계를 연결한 네트워크

www : 인터넷위에서 동작하는 하나의 서비스 (웹문서를 주고 받을 수 있는 서비스)

HTML : 웹문서를 만들 때 사용하는 언어

HTTP : HyperText 를 전송하기 위한 통신규약(약속)

웹서버 : 웹문서를 제공하는 서비스를 할 수 있는 프로그램 또는 컴퓨터

웹어플리케이션서버 : 동적인페이지를 생성하여 제공하는 웹서버

클라이언트, 서버구조 : 클라이언트와 서버로 구분됨
페이지요청, 페이지응답으로 구분 함

최초의 브라우저 : 월드와이드웹www

최초의 웹서버 : 1990년 12월 24일 웹 서버 라는 프로그램을 만들고 그 프로그램이 설치돼 있는 컴퓨터에 info.cern.ch 라는 주소를 부여

"home of the first website"

팀버너스리



영국의 물리학자. www, [URL](http://url), [HTTP](http://http) 등을 고안하여 웹을 탄생시킨 아버지로 유명하다. 그가 없었더라면 나무위키는 물론 구글, 네이버, 카카오, 페이스북의 등장도 늦었을 것이다.

위 공로를 인정받아 기사작위를 받았고, 왕립학회의 회원(FRS)으로 활동 중이다. 현재는 월드와이드웹 컨소시엄(W3C)¹⁵¹ 소장으로 재직.

1991년 8월 6일 최초의 웹사이트

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) ,[Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

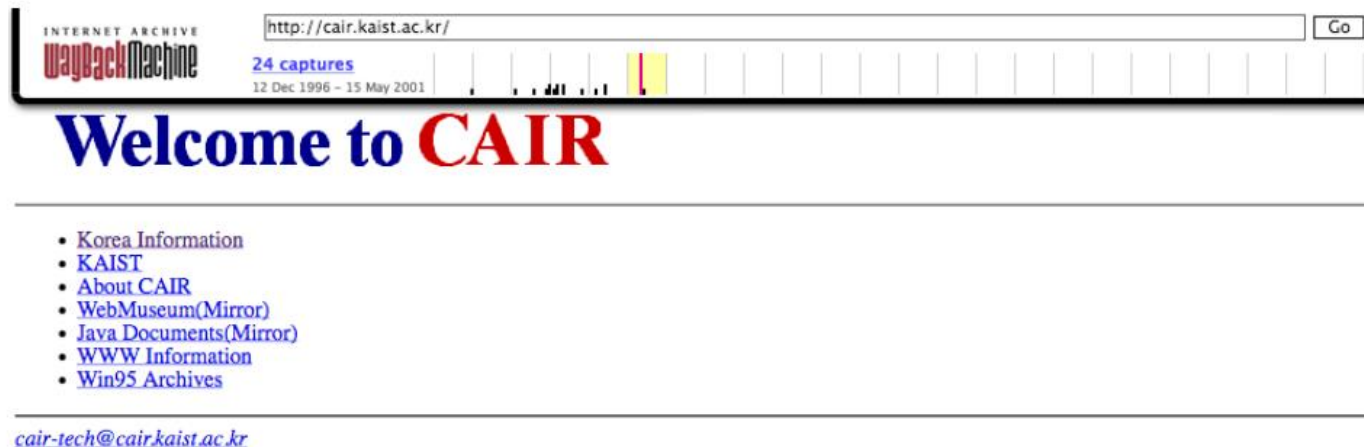
Getting the code by [anonymous FTP](#) , etc.

복원된 세계 최초 웹사이트 모습. 출처: <http://info.cern.ch/hypertext/WWW/TheProject.html>

어디에서나 컴퓨터에 저장된 모든 정보가 링크 될 수 있다고 상상해보세요. 모든 것을 다른 모든 것에 연결할 수있는 공간을 만들기 위해 컴퓨터로 프로그래밍 할 수 있다고 상상해 보세요.

– Tim Berners-Lee, World Wide Web 발명가

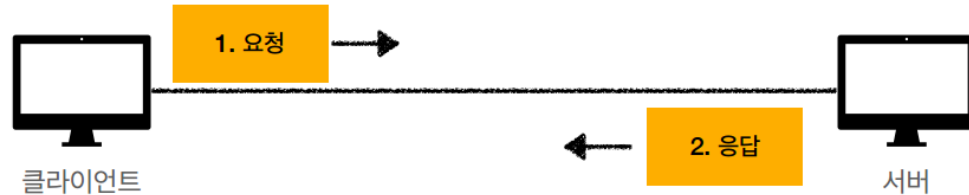
1993년 KAIST:우리나라 최초의 홈페이지



HTTP

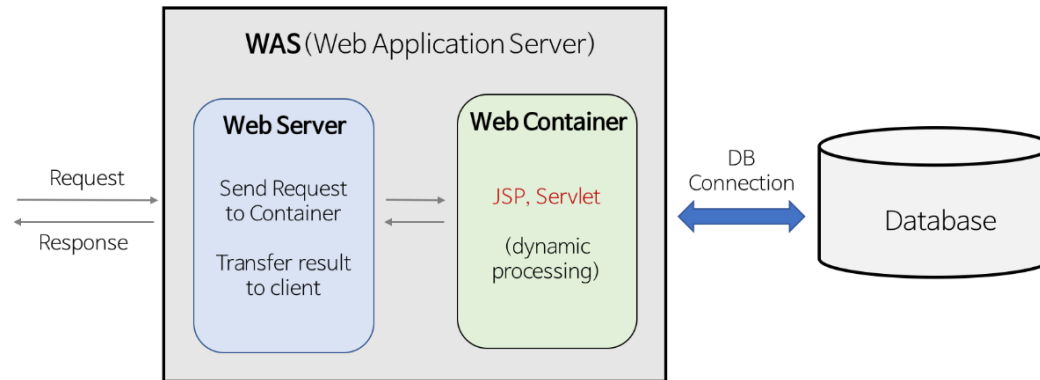
클라이언트 서버 구조

- Request Response 구조
- 클라이언트는 서버에 요청을 보내고, 응답을 대기
- 서버가 요청에 대한 결과를 만들어서 응답



HTTP 메시지에 모든 것을 전송

- HTML, TEXT
- IMAGE, 음성, 영상, 파일
- JSON, XML
- 거의 모든 형태의 데이터 전송 가능
- 서버간에 데이터를 주고 받을 때도 대부분 HTTP 사용



```
GET /search?q=hello&hl=ko HTTP/1.1
Host: www.google.com
```

예) HTTP 요청 메시지

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=UTF-8
Content-Length: 3423

<html>
<body>...</body>
</html>
```

예) HTTP 응답 메시지

```
GET /search?q=hello&hl=ko HTTP/1.1
Host: www.google.com
```

예) HTTP 요청 메시지

요청 메시지도 body 본문을 가질 수 있음

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=UTF-8
Content-Length: 3423

<html>
<body>...</body>
</html>
```

예) HTTP 응답 메시지

start-line 시작 라인

header 헤더

empty line 공백 라인 (CRLF)

message body

HTTP 메시지 구조