

The background of the slide is a dark blue field filled with a complex, glowing pattern. This pattern consists of numerous overlapping squares and lines that create a sense of depth and movement, reminiscent of a digital or network environment. The glowing elements are in various shades of blue, from deep indigo to bright cyan.

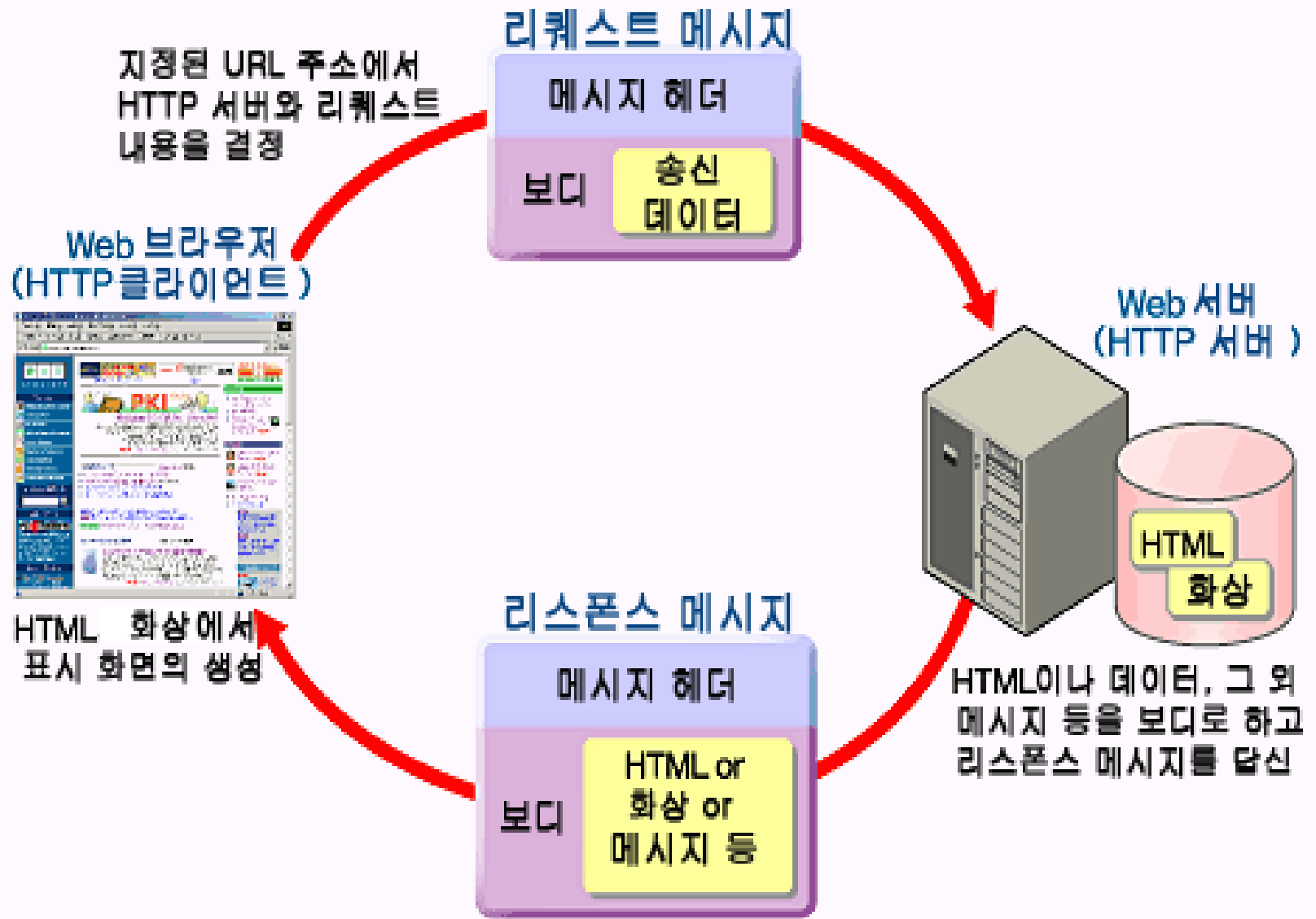
HTTP

(Hyper Text Transfer Protocol)

Agenda

- ◆ HTTP 개요
- ◆ HTTP 동작방식
 - ◇ 요청(request)
 - ◇ 응답(response)
- ◆ HTTP 요청방법
 - ◇ GET
 - ◇ POST
 - ◇ URL 구조

HTTP 개요



HTTP 동작 방식

HTTP(Hypertext Transfer Protocol)는 웹 서버와 웹 클라이언트(웹 브라우저)간에 통신하기 위한 프로토콜(약속)이다. CGI나 서버사이드 프로그래밍을 하기 위해서는 HTTP 프로토콜을 어느 정도 이해할 필요성이 있다. 이곳에서는 간단하게 HTTP 프로토콜에 대해 알아보자. 웹 브라우저는 HTTP 프로토콜에 맞게 요청(request)을 웹 서버에 전송하고, 웹 서버는 이에 따라 응답(reply)을 HTTP 프로토콜에 맞게 웹 브라우저에 전송한다. 요청과 응답은 모두 크게 세 부분으로 구성되어 있다. 첫 번째는 요청/응답 라인, 두 번째는 헤더 부분, 셋째는 엔티티 바디(entity body) 부분이다.

웹 브라우저와 웹 서버의 통신은 웹 브라우저의 요청으로부터 시작된다. 다음은 요청에 사용되는 HTTP 포맷이다.

형태: HTTP 요청

```
<method><resource identifier><HTTP version><crlf>
```

```
[<Header>: <value>] <crlf>
```

```
.....
```

```
[<Header>: <value>] <crlf>
```

```
<crlf>
```

```
[Entity body]
```

HTTP 동작 방식 (요청1)

웹 브라우저가 웹 서버에 요청을 전달하는 작업은 다음과 같은 절차를 거쳐서 진행된다.

1. 웹 클라이언트(웹 브라우저)는 미리 알려진 포트(보통 80번)로 연결을 시도한다. 연결이 이루어진 다음에 웹 클라이언트는 HTTP 명령어, 문서 주소, HTTP 버전 정보를 웹 서버에 전달한다. 다음 예는 index.html 문서를 요청하기 위해서 GET 메소드를 사용하는 것을 보여준다.

예:

```
GET /index.html HTTP/1.0
```

2. 요청 라인 다음 줄에 웹 클라이언트는 헤더 정보, 구성 정보(configuration information), 받아들일 문서 포맷 등의 정보를 옵션으로 웹 서버에 전달한다. 모든 헤더 정보들은 한 줄에 하나씩 기술되며, 헤더 이름과 값으로 구성된다. 헤더 정보의 끝에는 공백 라인을 추가해서 헤더의 끝임을 알려준다.

예:

```
User-Agent: Mozilla/2.03Gold(WinNT: I)
```

```
Accept: text/html, img/gif
```

HTTP 동작방식 (요청2)

3. 요청과 헤더 정보를 보낸 다음에 웹 클라이언트는 추가적인 정보를 웹 서버에 보낼 수 있다. 보통 이 데이터는 CGI에서 POST 방법으로 웹 서버에 전달하기 위해서 사용된다.

웹 클라이언트로부터 요청을 받은 웹 서버는 요청에 맞는 적당한 작업을 수행하고, 응답을 웹 클라이언트에 전달한다. 웹 브라우저의 요청에 응답하는 서버의 HTTP 포맷은 다음과 같은 형태를 취한다.

형태: HTTP 응답

```
<HTTP> <result code> [<explanation>]<crLf>
```

```
[<Header>: <value>] <crLf>
```

```
.....
```

```
[<Header>: <value>] <crLf>
```

```
<crLf>
```

```
[Entity body]
```


HTTP 동작방식 (응답1)

웹 서버는 클라이언트의 요청에 대해 다음과 같은 절차에 따라 응답한다.

1. 서버의 응답은 HTTP 버전, 상태 코드, 설명으로 구성된 상태 라인을 이용해서 응답한다. HTTP 버전은 웹 서버가 사용하는 HTTP 버전을 의미한다. 상태 코드는 클라이언트의 요청에 따른 서버의 결과를 표현하기 위한 것으로 세 자리 숫자로 표현된다. 다음은 각 상태 코드의 의미이다.

·1xx: 정보 - 요청이 받아들여졌고, 처리중임.

·2xx: 성공 - 클라이언트의 요청이 성공적으로 처리되었음.

·3xx: 경로변경 - 요청이 완전히 처리하기 위해 추가적인 액션

·4xx: 클라이언트 에러 - 클라이언트의 요청 문법이 틀렸거나, 요청이 처리될 수 없음.

A漢

·5xx: 서버 에러 - 정당한 요청.

HTTP 동작 방식 (응답2)

설명 부분은 상태 코드를 사람이 읽을 수 있는 간단한 단어로 표시한 것이다. 다음 예의 상태 코드 200은 클라이언트의 요청이 성공적으로 수행되었음을 의미한다.

예:

```
HTTP/1.0 200 OK
```

2. 상태 라인 이후에 웹 서버는 자신과 요청된 문서에 대한 정보를 헤더 정보를 통해서 클라이언트에 전달한다. 헤더의 끝은 빈 라인으로 표시한다.

예:

```
Date: Tue, 28 Mar 2000 06:18:26 GMT
```

```
Server: Apache/1.3.9 (Unix) tomcat/1.0 PHP/3.0.12
```

```
Last-Modified: Tue, 21 Sep 1999 02:35:00 GMT
```

3. 클라이언트의 요청이 성공적으로 수행되었다면, 요청된 자료가 전송된다. 전송되는 자료는 파일 내용이거나, CGI 프로그램의 응답일 수 있다.

HTTP 요청 방식

웹 클라이언트(예: 웹 브라우저)의 요청에서 HTTP 명령어(GET, POST 등)는 method라고 불리운다. method는 클래스의 함수와 혼동될 수 있다. 책에서는 분명하게 구별할 수 있는 경우에는 "메소드"라는 단어를 사용하고, 클래스 함수와 혼동될 수 있는 경우에는 "method"라는 단어를 사용하기로 한다. 앞에서 텔넷을 이용해서 테스트할 때 사용한 GET이 바로 HTTP의 메소드이다. 이 메소드의 종류로는 GET, POST, HEAD가 많이 사용된다. 이외에 LINK, UNLINK, PUT, DELETE, OPTIONS, TRACE 가 정의되어 있지만, 웹 서버에 따라 구현되어 있는 경우도 있고 아직 구현되지 않은 부분도 있다.

HTTP 요청 방식-GET 메소드

GET 메소드는 서버의 URI에 기술된 위치에 있는 정보를 요청하는 명령어이다. 우리가 웹 브라우저에 URL을 기술해서 어떤 사이트를 방문하는 것은 모두 GET 메소드를 사용하는 것이다. GET 메소드를 사용하면 클라이언트의 요청에서 엔티티 바디(entity body) 부분은 항상 비어있는 상태로 서버에 전달된다. 웹 클라이언트가 GET 메소드를 이용하면 서버는 상태 라인, 헤더, 요청된 데이터로 구성된 응답을 보내준다.

웹 브라우저에서 URL을 기술해서 사이트를 방문하는 것은 모두 GET 메소드 방식으로 작동한다. 따라서 다음과 같은 URL은 GET 메소드를 사용한다.

예:

`http://java.sun.com/`

HTTP 요청 방식-POST 메소드

GET 메소드는 웹 서버에 데이터를 요청하는데 주로 사용되지만, 어떤 경우에는 데이터를 웹 서버에 전송하기 위해서도 사용된다. 이것은 HTML에서 FORM의 METHOD에 GET을 기술하는 경우에 웹 서버에 데이터를 전달하기 위해서 사용될 수 있다. 그러나, GET을 이용해서 데이터를 전송하는 경우에는 전달되는 데이터의 길이에 한계가 있고, 전달되는 데이터가 정확하게 전달되어 있는지 체크하는 기능이 없다. 따라서, 웹 서버에 큰 데이터를 보내기 위해서는 POST 메소드를 사용한다. GET을 이용해서 데이터를 전송하기 위해서는 이름과 값의 쌍으로 전달된다. 다음 예에서 month=august&date=24는 웹 서버에 전달되는 이름과 값의 쌍이다. 이것을 다른 말로는 쿼리 스트링(query string)이라고도 한다. 쿼리 스트링은 ? 문자 다음부터 “이름=값”의 형태로 되어 있는데, 이것이 여러 번 반복될 때는 구별하기 위해서 & 문자를 사용한다.

예:

```
GET /cgi-bin/birthday.cgi?month=august&date=24 HTTP/1.0
```

HTTP 요청 방식-POST 메소드

POST 메소드는 클라이언트의 데이터를 웹 서버에 전달하기 위해서 사용된다. 전달될 데이터는 클라이언트 요청의 엔티티 바디 부분에 포함되어 웹 서버로 전달된다. POST 메소드는 HTML의 FORM에서 METHOD를 POST로 지정하는 경우에 사용된다. 다음은 POST를 사용하는 클라이언트 요청의 예이다.

예: 클라이언트 요청

```
POST /cgi-bin/birthday.pl HTTP/1.0
```

```
User-Agent: Mozilla/2.02Gold (WinNT; I)
```

```
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

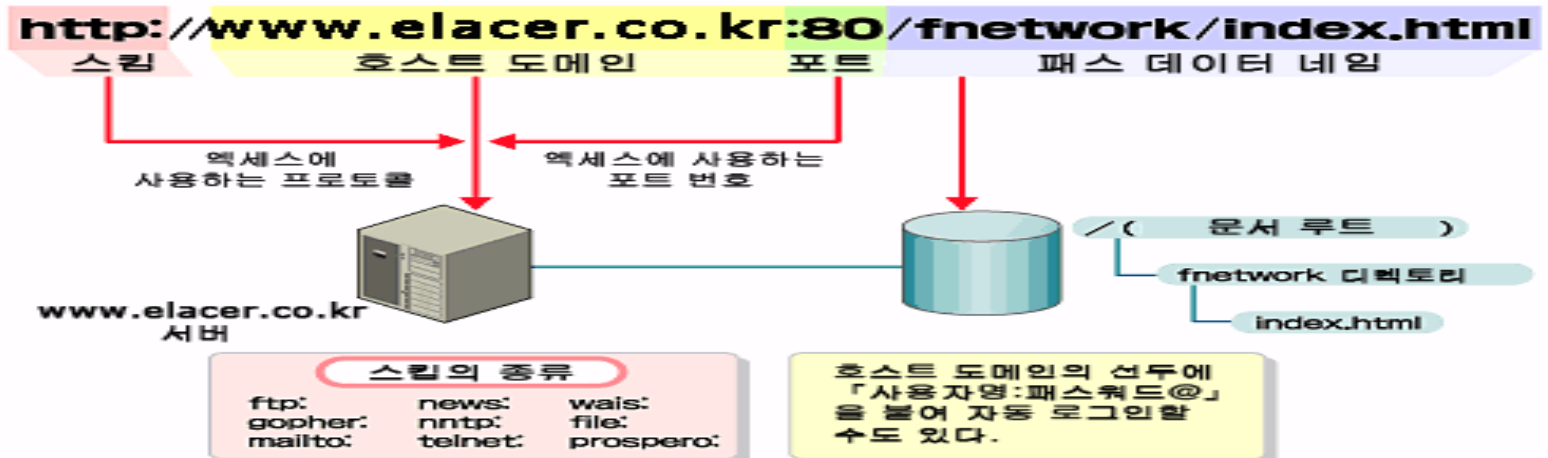
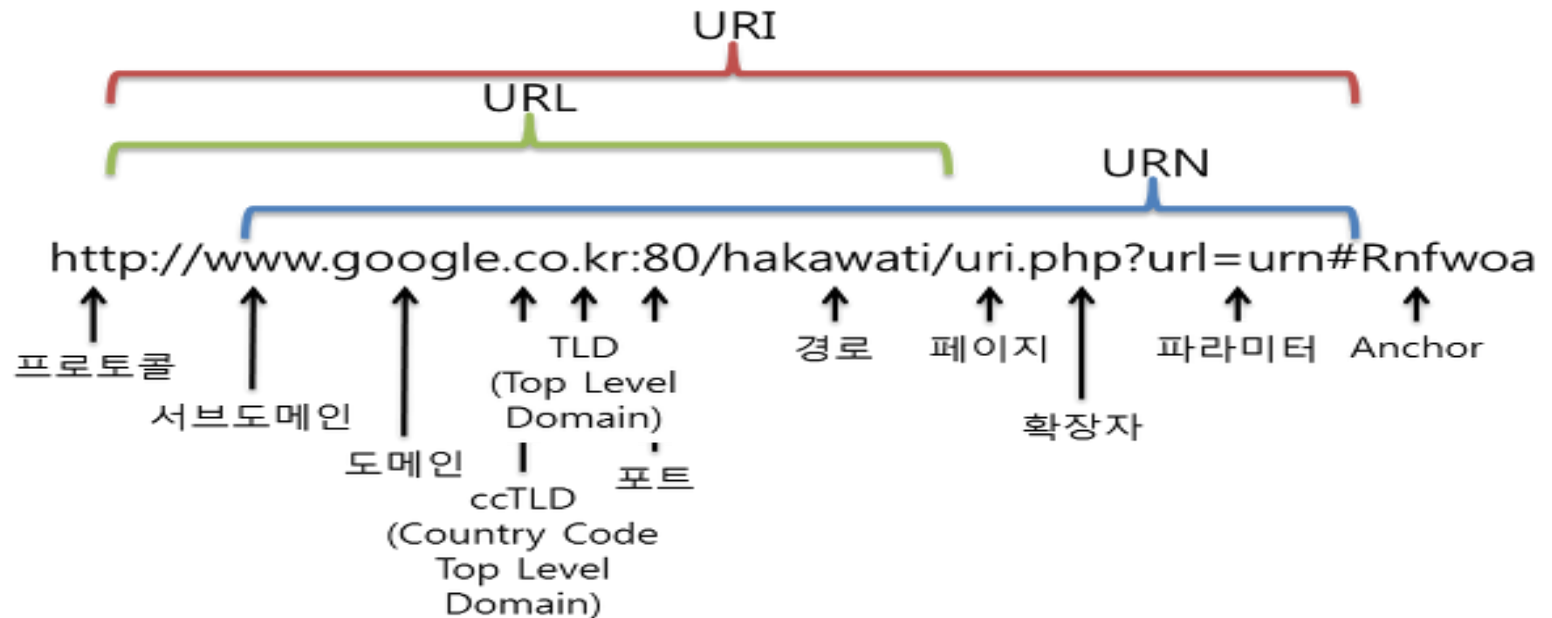
```
Host: www.ora.com
```

```
Content-type: application/x-www-form-urlencoded
```

```
Content-length: 20
```

```
month=august&date=24
```


URL 구조



The background of the slide is a deep blue with a complex, abstract pattern. It features a grid of glowing, slightly irregular squares that overlap with each other. Superimposed on this grid are several bright, wavy lines that curve across the frame, creating a sense of motion and depth. The overall effect is a high-tech, digital aesthetic.

THANK YOU