



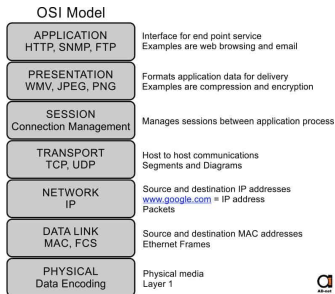
HTTP의 기본 동작과 PHP기초

허강준

충남대학교 정보보호동아리 ARGOS

2021. 04. 03

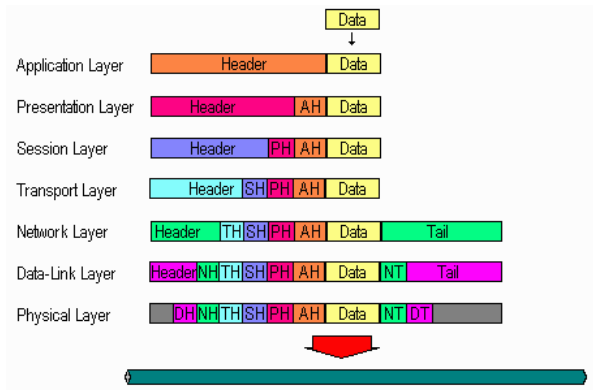
HTTP와 OSI 7계층



- Hyper Text Transfer Protocol
- OSI 7계층중 최상위 계층에 속하는 프로토콜
- 주로 웹 브라우저와 웹 서버 사이에서의 통신에서 사용

<https://www.ad-net.com.tw/osi-model-tcp-ip-network-models-must-concept-understand-move-deeper-networking-adventures/>

HTTP와 OSI 7계층



http://www.interfacebus.com/Design_OSI_Stack.html

HTTP 패킷 구조

- HTTP 패킷 = 헤더 + 데이터 (택배 상자)
- 헤더 = 보내고자 하는 데이터에 대한 설명 (택배 송장)
- 데이터 = 실제로 전송하고자 하는 데이터 (물건)
- 요청과 응답시 헤더가 조금씩 다름

HTTP 패킷 구조

```
GET https://www.argos.or.kr/ HTTP/1.1
Host: www.argos.or.kr
Connection: keep-alive
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) App
Accept: text/html,application/xhtml+xml,application/xml;q
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ko,en;q=0.9,en-US;q=0.8
Cookie: PHPSESSID= ; mobile=fals
```

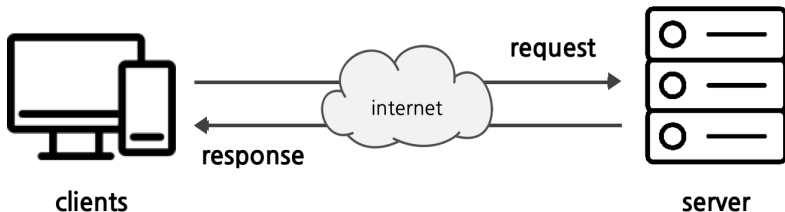
```
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Tue, 23 Mar 2021 05:05:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Last-Modified: Tue, 23 Mar 2021 05:05:38 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 43873

<!DOCTYPE html>
<html lang="ko">
<head>
<!-- META -->
<meta charset="utf-8">
<meta name="Generator" content="XpressEngine">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

■ 요청 패킷 (GET)

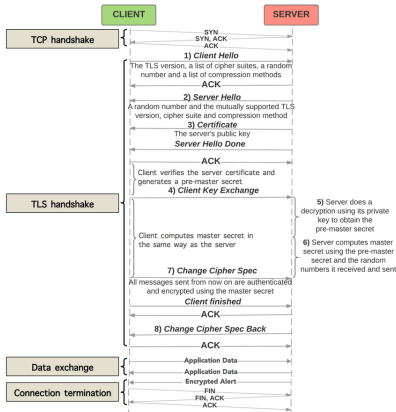
■ 응답 패킷

클라이언트와 서버



- 메일서버: 메일 (SMTP, IMAP 등) 관련 처리에 특화된 서버
- 게임서버: 게임 프로토콜 처리에 특화된 서버
- 웹서버: 웹 (특히 HTTP 요청) 관련 처리에 특화된 서버

HTTPS? TLS?



https://www.researchgate.net/figure/HTTPS-message-sequence-diagram-with-detailed-TLS-handshaking-steps_fig1_306187575

HTML : 웹페이지의 구조를 설명하는 언어

- Hyper Text Markup Language
- 프로그래밍 언어가 아님!
- 확장자는 .html

HTML 문서의 기본 구조

```
<!-- 이걸 주석! -->
<!doctype html> <!-- DOCTYPE 선언 (html5) -->
<html> <!-- Html 문서 태그 -->
  <head> <!-- Head 태그 -->
    <meta charset="utf-8"> <!-- 문서 인코딩 -->
    <title>Hello, HTML!</title> <!-- 창 제목 -->
  </head>
  <body> <!-- 본문 태그 -->
    <!-- h1 태그 (주로 제목에 사용) -->
    <h1>Hello, HTML!</h1>
  </body>
</html>
```

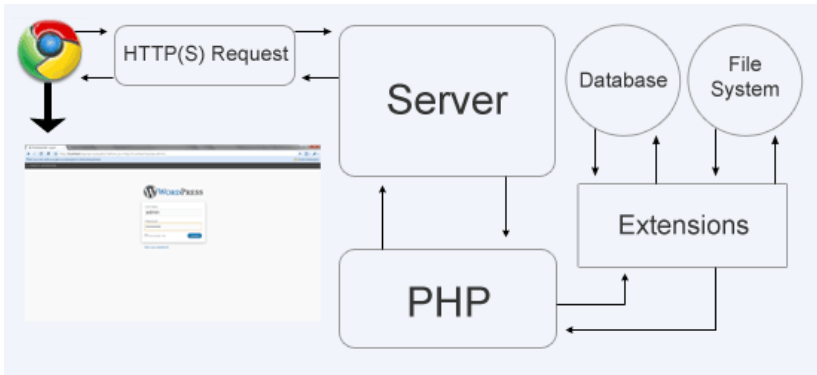
주로 사용할 태그들

- `html, head, body` : HTML 문서를 정의할 때 필요
- `title` : 문서의 제목 설정
- `link` : 주로 CSS를 불러올 때 사용
- `script` : 자바스크립트를 불러오거나 작성할 때 사용
- `h1, h2, ..., h6` : 제목을 표현하는 태그
- `p` : 문단을 표현하는 태그
- `a` : 링크를 표현하는 태그
- `img` : 이미지를 불러올 때 사용
- `table, tr, td` : 표를 표현하는 태그
- `form, input` : 폼을 표현하는 태그 (제일 중요)

PHP

- HTML은 기본적으로 정적(static)한 페이지를 작성
- 때로는 동적인 페이지가 필요 (e.g. 로그인 여부 확인, 게시글 표시 등...)
- 상황에 따라 페이지가 생성되어야 함!

PHP의 동작 방식

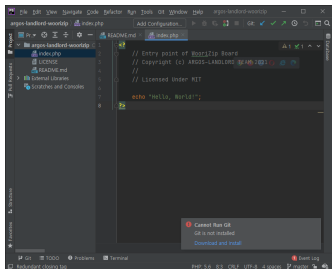


<http://nepsdotin.github.io/php-tutorial/site/unit-3/how-php-works/>

여태까지 내용 정리하기

- HTTP를 처리하는 서버는 웹 서버
- 웹 문서를 작성하는 언어 HTML
- 동적 페이지를 작성하는 언어 PHP
- 웹서버가 PHP 요청을 받으면 PHP 인터프리터로
- PHP 인터프리터의 실행 결과는 웹서버를 거쳐 사용자로

JetBrains PhpStorm



- PHP 개발에 필요한 편의기능 제공
- 라이브 서버, 인텔리센스, 패키지관리 등...
- 학교 이메일로 무료 사용가능

PHP 디버그 환경 설정

- Windows: windows.php.net
- Linux: 각 패키지 매니저별 설치방법 참조
- OSX: ???

인텔리센스나 이런저런 도움이 필요 없다면...

- Visual Studio Code를 써도 됩니다!
- PHP Server 확장기능: 라이브 서버가 필요할 경우
- 단 직접 경로 잡고 세팅해야함!
- 제대로 세팅하지 않으면 인텔리센스가 많이 불편

echo "Hello, World!"

■ 문자열 출력은 echo를 사용



<?

```
echo "Hello, World!";  
// 숫자도 OK  
echo 12345;
```


특별한 변수들: Superglobals

- \$_GET : GET 파라미터를 담는 전역변수
- \$_POST : POST 파라미터를 담는 전역변수
- \$_FILES : POST로 날아온 파일 정보를 담는 전역변수
- \$_SERVER : 서버 환경설정, 클라이언트 연결 정보를 담는 전역변수

GET? POST?

- HTTP 메서드의 일종
- 메서드에 따라 백엔드로 데이터를 전달하는 방법이 다름
- 데이터가 주소에 실려있다? \$_GET 에 할당
e.g.) `http://example.com/index.php?page=100`
- 데이터가 패킷에 실려있다? \$_POST 에 할당
- 자세한 내용은 2주차에!

연산자들

- C나 자바에서 사용하는것과 거의 동일!
- 대입: =
- 산술연산자: +, -, *, /, <<, >>, &, |
- 논리연산자: ==, ===, !=, !==, <, > &&, ||
- 증감연산자: ++, --
- 문자열연결: .

제어문, 반복문

```
switch($var) {  
    case 1:    // var가 1일 때  
        break; // switch 빠져나오기  
  
    case 2:    // var가 2일 때  
        break;  
  
    case 3:    // $var가 3, 4일 때  
    case 4:  
        break;  
  
    default:   // 그 외의 경우?  
        break;  
}
```

제어문, 반복문

```
while ($cond) { // $cond가 true일때 계속 반복
    continue; // 밑에 있는거 다 무시하고 루프를 처음부터
    break; // $cond나 밑에 있는거 다 무시하고 루프 빠져나오기
}
```

제어문, 반복문

```
$a = 0;  
  
for($i = 0; $i < 10; $i++) {  
    $a = $a + $i; // $a += $i; 도 가능  
}  
  
echo $a; // 45
```

제어문, 반복문

```
$arr = ['a', 'r', 'g', 'o', 's'];  
  
// argos 출력  
foreach ($arr as $char) {  
    echo $char;  
}
```


실습: Hello, landlord!

- echo 문을 이용해서 Hello, (이름) 을 출력해보세요.
- 변수를 이용해볼 것!
- String Interpolation을 이용할 것!
- 파일명은 hello.php 로 (브라우저로 접속 가능해야함)
- 기한: 다음주 활동시간까지

Project: WooriZip Board : 게시판 만들기

- 웹 서비스에서 가장 많이 사용되는 유형의 프로그램
- CRUD (Create, Read, Update, Delete)를 학습하기 좋음
- 프로그램 크기도 적당히 크기 때문
- 진도 나가면서 매주 기능 하나하나 구현해보기!
- 깃허브 레포에 푸쉬하면 자동반영 (조금 지연 있음)

질문?