

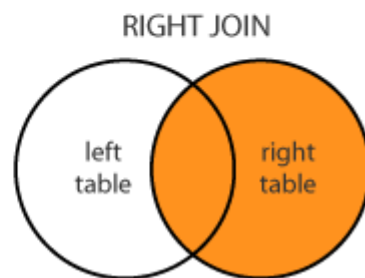
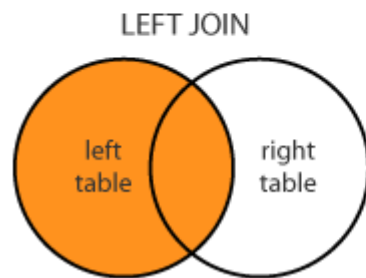
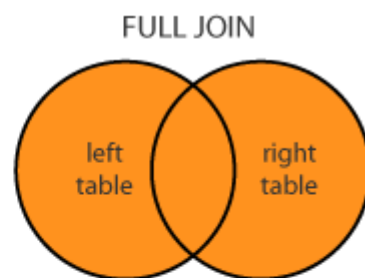
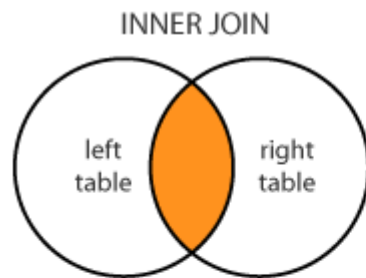
# **Fastcampus Data Science Extension SCHOOL**

**join, cohort with GA, All about Web**

**join**

# JOIN

- INNER JOIN: 양쪽의 값 비교 후 조건에 맞는 데이터 병합
- LEFT JOIN: JOIN 왼쪽을 기준으로 오른쪽의 일치하는 데이터 병합
- SELF JOIN: 자기 자신을 병합
- RIGHT JOIN: JOIN 오른쪽을 기준으로 왼쪽의 일치하는 데이터 병합
- FULL OUTER JOIN : 일치하지 않는 값까지 모두 병합



# **Cohort Analysis**

## **with google analytics**

# User retention

- Full retention
  - The number of returning visitors every single day
- Classic retention
  - returning user in specific day after day 1
- Rolling retention
  - Full + Classic

# Web Structure



## Network

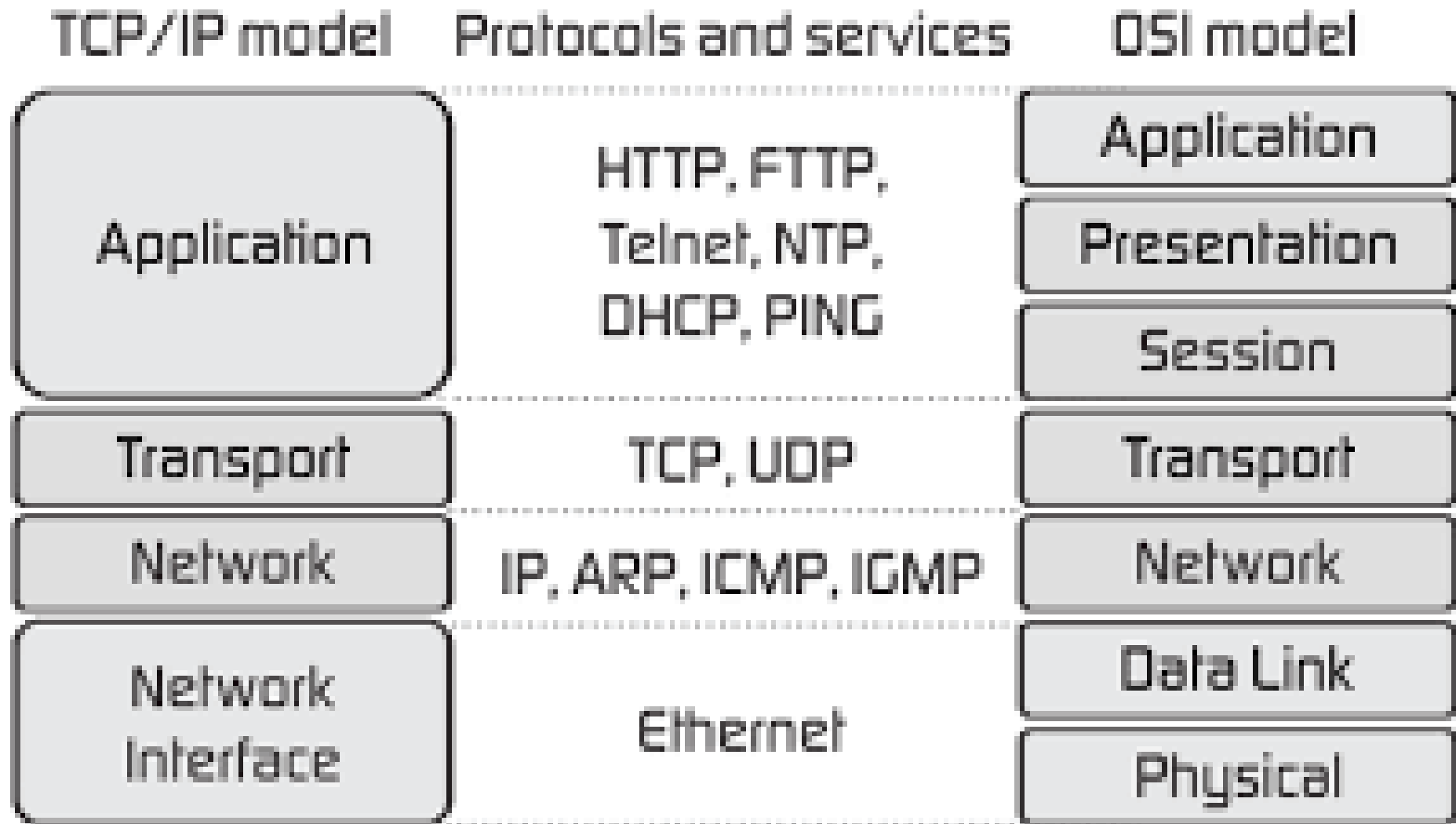
A computer network or data network is a telecommunications network which allows nodes to share resources.

--> 컴퓨터간 리소스를 공유 가능하게 만드는 통신망

# Ethernet

- 전세계의 사무실이나 가정에서 일반적으로 사용되는 유선 LAN에서 가장 많이 활용되는 기술 규격
- ether == 에테르 == 빛의 매질
- IEEE 802.3 규약 기반
- OSI 7 Layer에서 Data-link Layer 에 위치

# OSI 7 layer

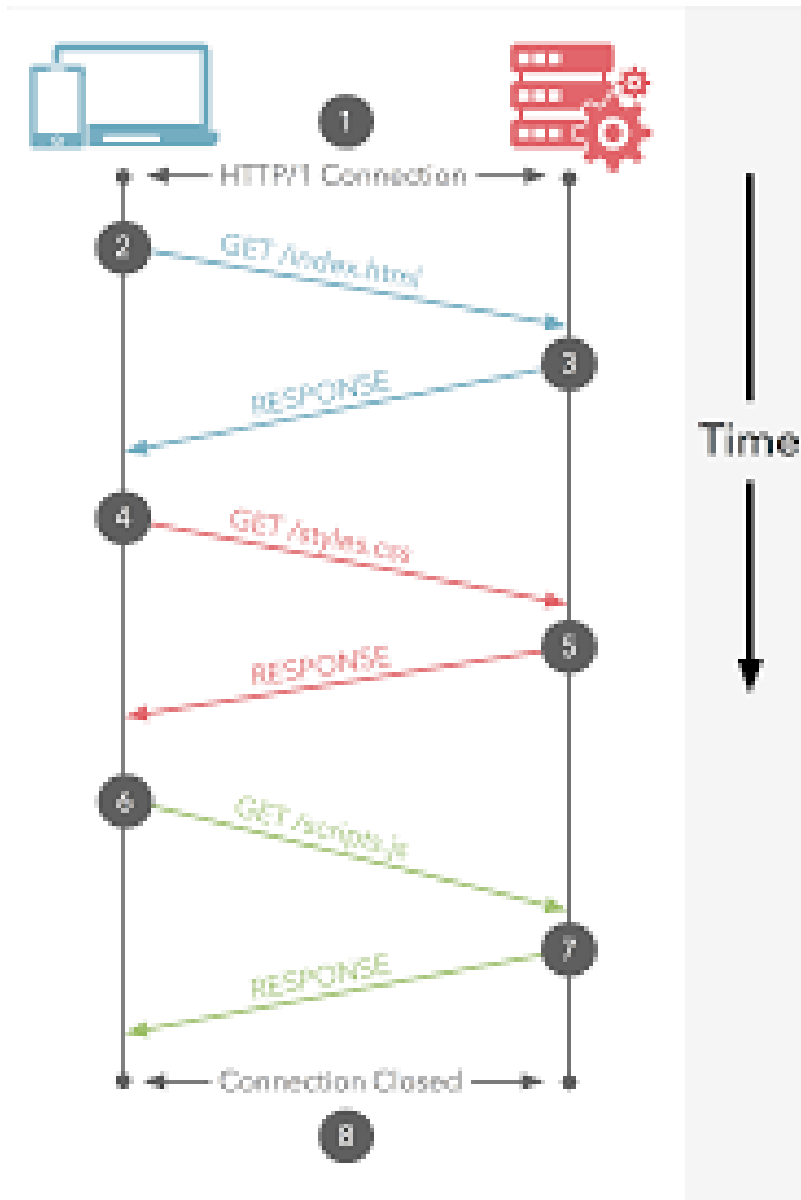


## Internet

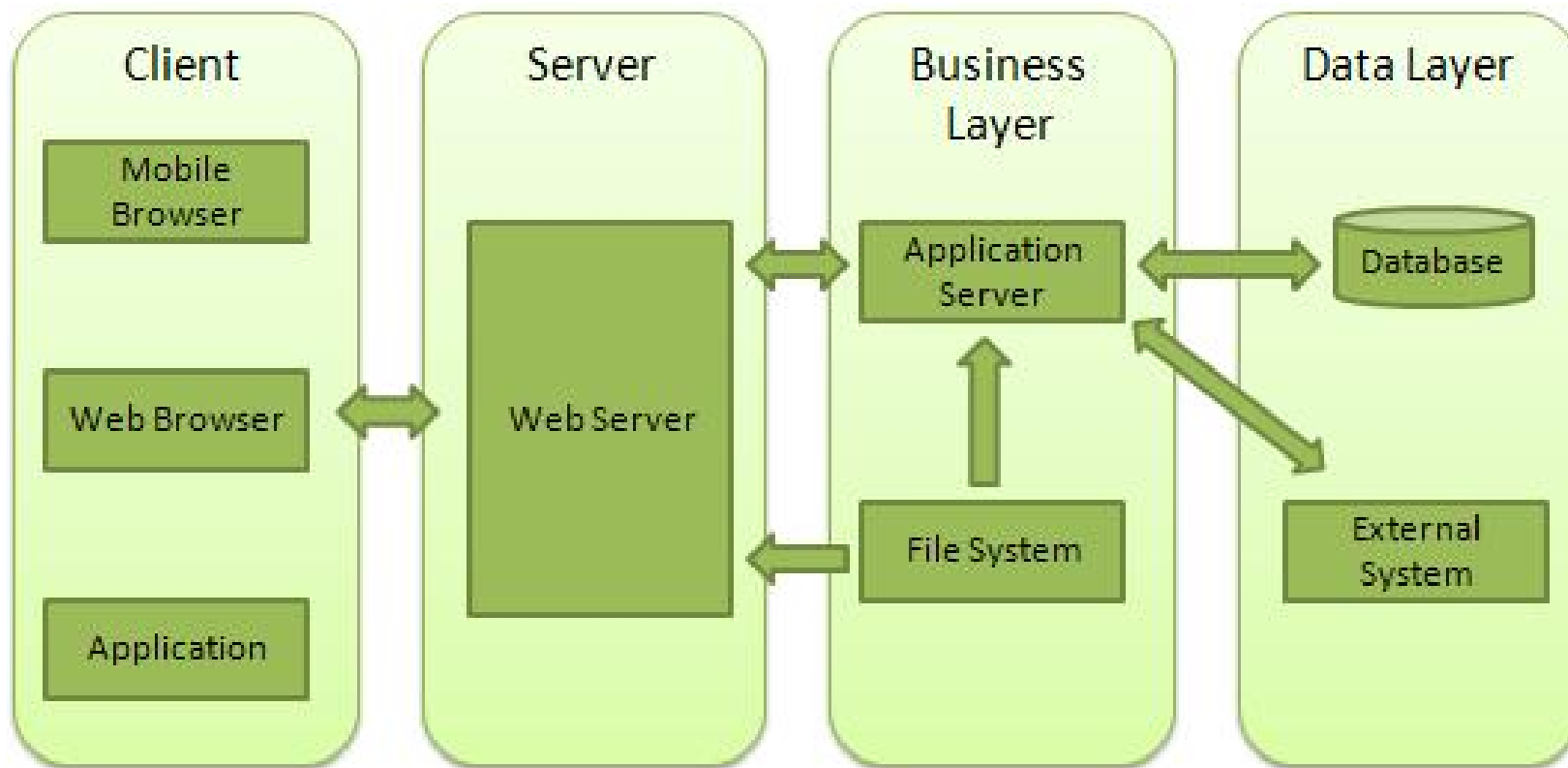
TCP/IP를 활용하여 정보를 주고 받는 통신 네트워크(www)

**WWW(World Wide Web) == hypertext transfer through TCP/IP**

# Request & Response



# Web Architecture



## 웹 개발 패턴의 변화

```
<html>
<head></head>
<body>
<h1>Static Header</h1>
<div>Static Contents</div>
</body>
</html>
```

- 1991 ~ 1999: Sir Timothy John "Tim" Berners-Lee가 하이퍼텍스트 기반의 프로젝트를 제안한 이후 정적인 콘텐츠를 중심으로 한 웹 기술이 발달



## 웹 개발 패턴의 변화

```
<html>
<head></head>
<body>
<h1>{% Dynamic Header %}</h1>
<div>{% Dynamic Contents %}</div>
</body>
</html>
```

- 1999 ~ 2009: Linux, Apache, Mysql, Php 중심의 동적인 서버, 정적인 클라이언트 모델이 지속됨

# 웹 개발 패턴의 변화

```
<html>
<head>
<script src="https://unpkg.com/vue"></script>
</head>
<body>
<h1>{{ header }}</h1>
<div id="app">
  {{ message }}
</div>
<script>
var app = new Vue({
  el: '#app',
  data: {
    message: '안녕하세요 Vue!'
  }
})
</script>
</body>
</html>
```

- 2010 ~ 현재: JavaScript!! (Dynamic Web Client)

# HTML

- HyperText Markup Language

# HTML

```
<!doctype html>
```

# HTML

```
<!doctype html>  
<html>  
  <head></head>  
  <body></body>  
</html>
```

# HTML

```
<head>  
  <meta charset="utf-8">  
  <meta name="viewport"  
    content="width=device-width, initial-scale=1.0">  
  <title></title>  
</head>
```

# HTML

```
<body>
  <div id="main-wrapper">
    <h1 class="article-title"></h1>
    <p>This is <span>Home</span>.</p>
    <a href="#" target="_blank">hypertext</a>
    
  </div>
</body>
```

## HTML - Semantic Element

```
<header>  
  <nav></nav>  
</header>  
  
<section>  
  <article></article>  
</section>  
  
<aside></aside>  
<footer></footer>
```



# CSS

- Cascading Style Sheet
- 웹 문서의 스타일링을 위한 스타일시트

## CSS basic style

```
body {  
    background-color: gray;  
}
```

# CSS Selector

## id, class, just tags

```
#some-id {color:#ff0000;}  
  
.some-class {color:#00ff00;}  
  
body {background-color:#dddddd;}
```

## group selector

```
h1, h2, h3, h4, h5, h6 {font-family:Helvetica;}
```

## child selector

```
body > h1 {align:center;}
```

## attribute selector

```
p[title='introduce'] {font-family:Helvetica;}
```

## JavaScript

- 객체 기반의 스크립트 프로그래밍 언어
- 웹페이지의 동적인 제어 목적
- Netscape의 Brendan Eich가 모카(Mocha)를 개발
- LiveScript -> JavaScript로 개명

## JS DOM API

- DOM: Document Object Model
- HTML 문서를 분석하여 구조화

## XPath

- XML Path Language
- XML 문서의 요소와 속성을 통해 특정한 요소로 접근할 수 있도록 도와줌



## Basic XPath

```
<body>
  <div id="site-wrapper">
    <h1 class="main-title">Page Title</h1>
    <div>
      <p class="paragraph">
        I am
        <span>a</span>
        boy.
      </p>
      <a href="#">Hypertext</a>
    </div>
  </div>
</body>
```

## Basic XPath

`h1` : nodename

`/html` : root node

`//div` : select from current node

`.` : current node

`..` : parent node

`@` : attribute

## Basic XPath

```
//body/div/p
```

```
//*[@id="site-wrapper"]/div/a
```

```
//*[@class="paragraph"]/text()
```

