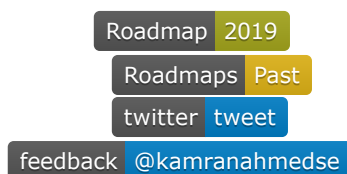


## 개발자 로드맵

2019년 웹 개발자가 되기 위한 로드맵



아래에는 프론트엔드 및 백엔드 또는 데브옵스가 되기 위한 경로와 사용하고자 하는 기술을 보여 주는 차트가 있습니다.

저는 제자들의 시야를 넓히고자 하는 옛 교수님을 위해 이 차트를 만들었습니다.  
누구나 커뮤니티를 돕기 위해 이곳에 공유해주세요.

저의 [블로그](#)를 확인하여 [트위터](#)에 "안녕"이라고 말해주세요.

## 경고사항

이 로드맵의 목적은 전체적인 방향에 대한 아이디어를 제공하고  
다음에 무엇을 배워야 할지 혼란스러운 경우와 트렌드에 뒤떨어진 것을 배우지 않도록 안내하는  
것입니다.  
왜 어떤 도구가 다른 도구보다 적합하다고 생각되는지에 대한 이해를 넓혀 가야 하며  
유행에 따른 도구를 기억하는 것이 결코 직업에 가장 적합한 것은 아니라는 것을 기억해야 합니다.

## 소개

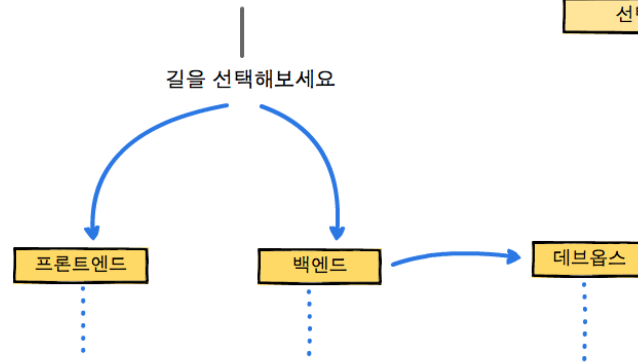
필수 사항 (어떠한 길이든)

Git - 버전 관리
기본적인 터미널 사용법
자료구조 & 알고리즘
SOLID, KISS, YAGNI
GitHub
라이선스
Semantic Versioning
SSH
HTTP/HTTPS와 APIs
디자인 패턴
문자 인코딩

설명

개인적인 추천
선택 가능한 옵션

## 웹 개발자 (2019년)



## 프론트엔드 로드맵

---

## 프론트엔드

기초를 배우요

기초 배우기  
의미론적(Semantic)인 HTML 작성  
기본적인 SEO  
접근성

HTML  
CSS  
JavaScript의 기초

기초 배우기  
레이아웃 만들기  
미디어 쿼리  
CSS 3 배우기

Floats  
Positioning  
Display  
Box Model  
CSS Grid  
Flex Box

기본 구문과 기본 문법  
DOM 조작 방법 배우기  
Fetch API / Ajax (XHR) 배우기  
모듈러 JavaScript와 ES6+ 배우기

개념 이해하기  
호이스팅, 이벤트 버블링, 스코프  
프로토타입, Shadow DOM, strict,  
브라우저 동작원리, DNS, HTTP

npm  
yarn

npm은 많이 향상되었고 v5+ 버전까지 출시되었지만  
여전히 yarn에 뒤처져있습니다.  
하지만 심각한 것은 아닙니다. 무엇이든 고르세요!

패키지 매니저

SASS  
PostCSS  
Less

CSS 전처리기

PostCSS가 전처리기는 아니지만  
하나의 전처리기로 사용할 수 있습니다.  
SASS를 사용하고 이후에 PostCSS에 대해 논의해보세요.  
아직 시장에 Less가 있지만  
제가 2019년에 시작한다면 선택하지 않겠습니다.

CSS 프레임워크

Bootstrap  
Materialize CSS  
Bulma  
Semantic UI

BEM  
OOCSS  
SMACSS

CSS 아키텍처

현대 프론트엔드 프레임워크에서는  
CSS in JS 방법론에 대한 요구가 더욱 커지고 있습니다.  
그렇지만 적어도 BEM 정도는 배워놓는다면  
장기적으로 도움이 될 것입니다.

빌드 도구

태스크 러너

린트와 포맷터

Prettier  
ESLint  
JSHint  
JSLint  
JSCS

모듈 번들러

npm 스크립트  
gulp

Webpack  
Parcel  
Rollup

Redux  
MobX  
이것들은 React에만  
관련이 있는 것이 아니며  
어떠한 프레임워크나 앱  
에서도 사용 가능합니다.

프레임워크 선택

React.js  
Angular  
Vue.js

RxJS  
ngrx

Vuex

시작하기 전에 싱글 페이지 어플리케이션의  
작동 방식 그리고 긍정적인 측면과  
부정적인 측면을 잘 이해해야 합니다.

CSS in JS

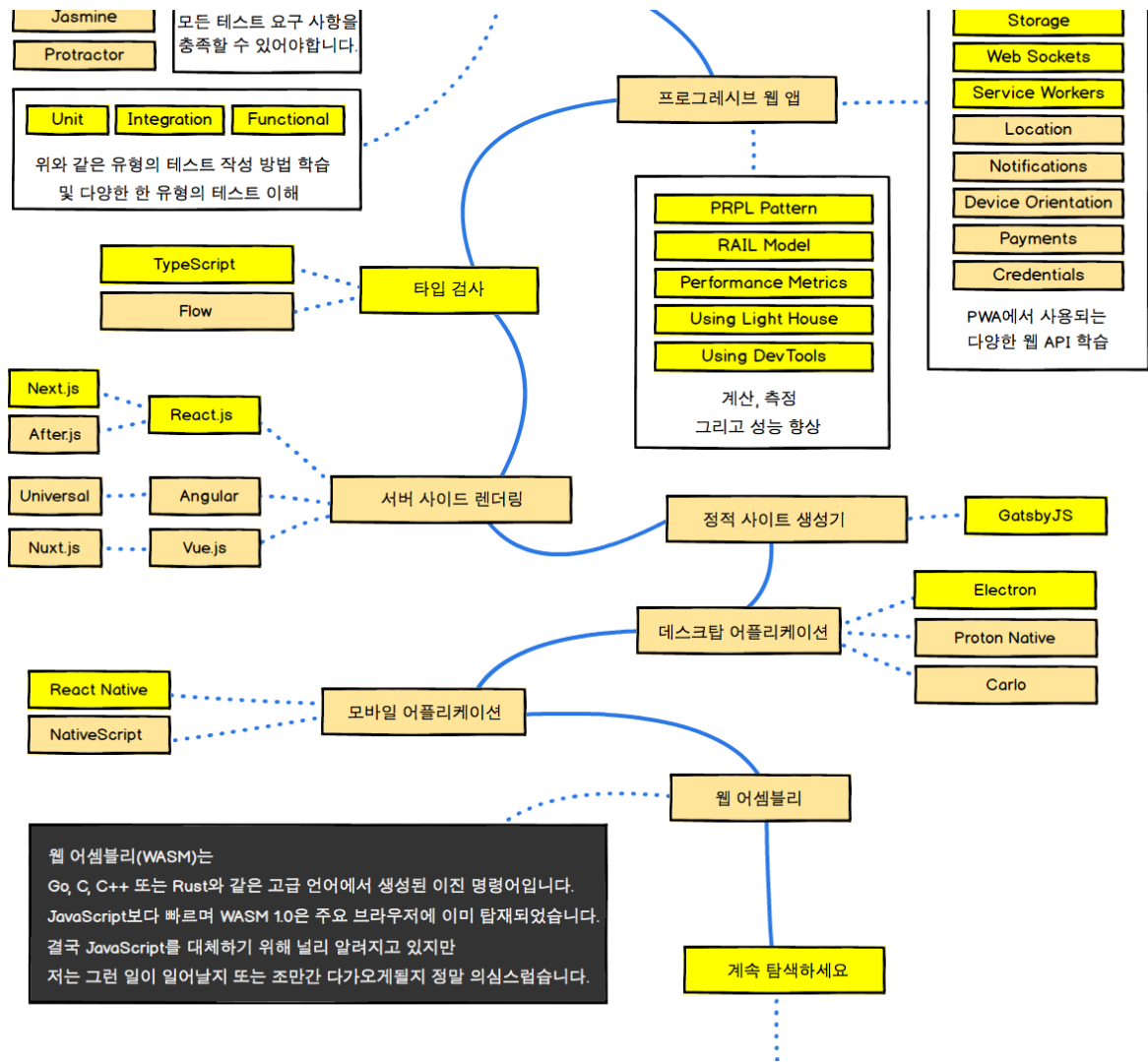
Styled Components  
CSS Modules  
Emotion  
Radium  
Glamorous

Mocha  
Chai  
Ava  
Karma

Jest  
Enzyme  
Cypress

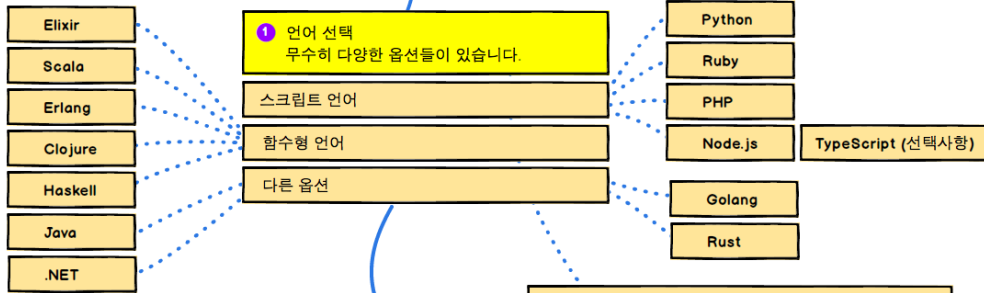
이것들을 통해

앱을 테스트 하세요



## 백엔드 로드맵

# 백엔드



2 배운 것을 실습하세요  
 선택한 언어로 연습하고  
 커맨드 라인(또는 콘솔) 어플리케이션 만들기

백엔드 개발을 시작하는 초보자의 경우  
 스크립트 언어 중 하나를 선택하는 것을 추천합니다  
 쉽고 빠르게 시작하려면 Node.js 또는 PHP를 사용하세요.  
 이미 백엔드에서 스크립트 언어를 사용하고 있다면  
 다른 스크립트 언어를 선택하지 말고  
 Golang, Rust 또는 Clojure를 살펴보세요.  
 새로운 관점을 명확하게 선사할 것입니다.

샘플 아이디어  
 사용하는 몇 가지 명령을 구현하세요.  
 예: 'ls' 레딧 포스팅을 가져 와서 /r/프로그래밍에 저장하는 명령  
 JSON 포맷의 디렉토리 구조를 제공하는 명령  
 위의 JSON을 읽고 매일 수행하는 일부 작업에 대한  
 디렉토리 구조 hink를 작성하고 이를 자동화하려고하는 명령

패키지 관리자는 어플리케이션에서 외부 의존성을 가져와  
 소유한 패키지를 배포하는 데 도움을 줍니다.

3 패키지 매니저를 배우세요  
 선택한 언어에 대해 패키지 관리자를  
 사용하는 방법에 대해 알아봅니다. 예. PHP의 composer,  
 Node.js의 NPM 그리고 yarn, Python의 pip, Ruby의 gems 등

보안 모범 사례에 대해 읽어 보세요.  
 OWASP 가이드라인을 읽고 다양한 보안 문제와  
 OWASP 가이드라인을 이해하며 다양한 보안 문제와 방법을 이해합니다.

4 표준(Standard) 및 모범 사례(Practice)  
 각각의 언어에는 고유한 표준과 모범 사례가 있습니다.  
 여러분이 고른 언어로 그것들을 공부하세요.  
 예를 들어 PHP에는 PHP-FIQ와 PSR이 있으며  
 Node.js에는 커뮤니티 등에 의해 구축되는 여러가지가 있습니다.

선택한 언어에 따라 각각의 용도가 다른 여러가지 사항이 있습니다.  
 Google에서 다양한 사항을 확인하고 필요한 것을 선택하세요.

PHP - [PHPUnit](#), [PHPSpec](#), [Codeception](#)  
 Node.js - [Mocha](#), [Chai](#), [Sinon](#), [Mockery](#), [Ava](#), [Jasmine](#)

다른 언어 사용자의 경우, 어떠한 논쟁도 하고 싶지 않기 때문에  
 여기서는 어떠한 것도 추천하지 않을 것 입니다.  
 주변을 둘러보고 적합한 것을 찾아보세요.

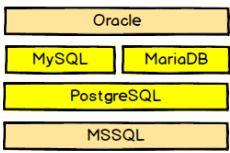
5 패키지와 라이브러리 만들어 배포하기  
 이제 패키지를 만들어 다른 사람이 사용할 수 있도록 배포하고  
 지금까지 배운 표준과 모범 사례를 수행하세요.

오픈 소스 프로젝트 기여  
 github에서 일부 프로젝트를 검색하고  
 오픈소스 프로젝트에서 풀 리퀘스트를 업니다  
 이를 위한 몇가지 아이디어:

학습한 모범 사례를 통하여 리팩토링하고 구현합니다  
 열린 이슈를 조사하고 해결하며 부가적인 기능을 추가하세요.

6 테스트에 대해 알아보기  
 여러가지 테스트 유형이 있지만 지금은 선택한 언어로  
 단위(유닛) 및 통합 테스트를 작성하는 방법에 대해 알아보겠습니다.  
 기타. mock, stub 등과 같은 다른 테스트 용어를 이해합니다.

💡 테스트 커버리지를 계산하는 방법에 대해 알아보세요.



8 관계형 데이터베이스 배우기  
 몇 가지 선택 사항이 있습니다.  
 그러나 만약 한가지를 배운다면 지금 당장  
 MySQL을 선택하세요. 이외에도  
 그것들이 어떻게 다른지와 사용법을 배우세요

7 위의 응용 단계에 대한 테스트 작성  
 먼저 이전 단계에서 구현한 실제 작업에 대한  
 단위 테스트를 작성합니다.

테스트를 작성하고 표준과 모범 사례를 수행하세요.  
 또한 데이터베이스의 경우 색인을 추가하고, 적절한 저장소 엔진을 사용하여  
 어플리케이션에서 쿼리를 사용하기 전에 해당 쿼리를 분석해야 합니다.

9 응용의 시간  
 지금까지 배운 모든 것을 사용하여 간단한 어플리케이션을 만듭니다.  
 가입, 로그인, CRUD가 있어야합니다. 예를 들면 블로그를 만듭니다.  
 누구나 가입할 수 있고 공개 프로필 페이지를 만들면  
 게시물을 작성, 업데이트 및 삭제할 수 있으며  
 공개 페이지는 게시자가 작성한 게시물을 표시합니다.

10 프레임워크 배우기

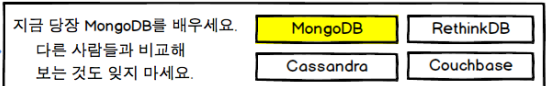
11 응용의 시간  
 원하는 프레임워크로 9에서 만든 것과 동일한 어플리케이션 만들기

12 NoSQL 데이터베이스 배우기  
 먼저 이러한 데이터베이스가 무엇인지, 관계형 데이터베이스와 어떻게 다르고  
 왜 필요한지 이유를 파악합니다. 여러가지 다른 선택 사항이 있습니다.  
 다른 옵션을 살펴보고 차이점을 확인하십시오  
 한가지 선택한다면 MongoDB 선택합니다.

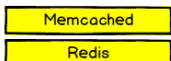
선택한 프로젝트와 언어에 따라 프레임 워크가 필요할 수도 있고  
 없을 수도 있습니다. 여러 가지 선택 사항이 있습니다.

PHP - 마이크로 프레임워크 [Laravel](#) 또는 [Symfony](#) 그리고 Slim, Lumen  
 Node.js - [Express.js](#), [Hapi.js](#)  
 Golang - 프레임워크 없이 코드를 작성하는 것을 선호합니다.

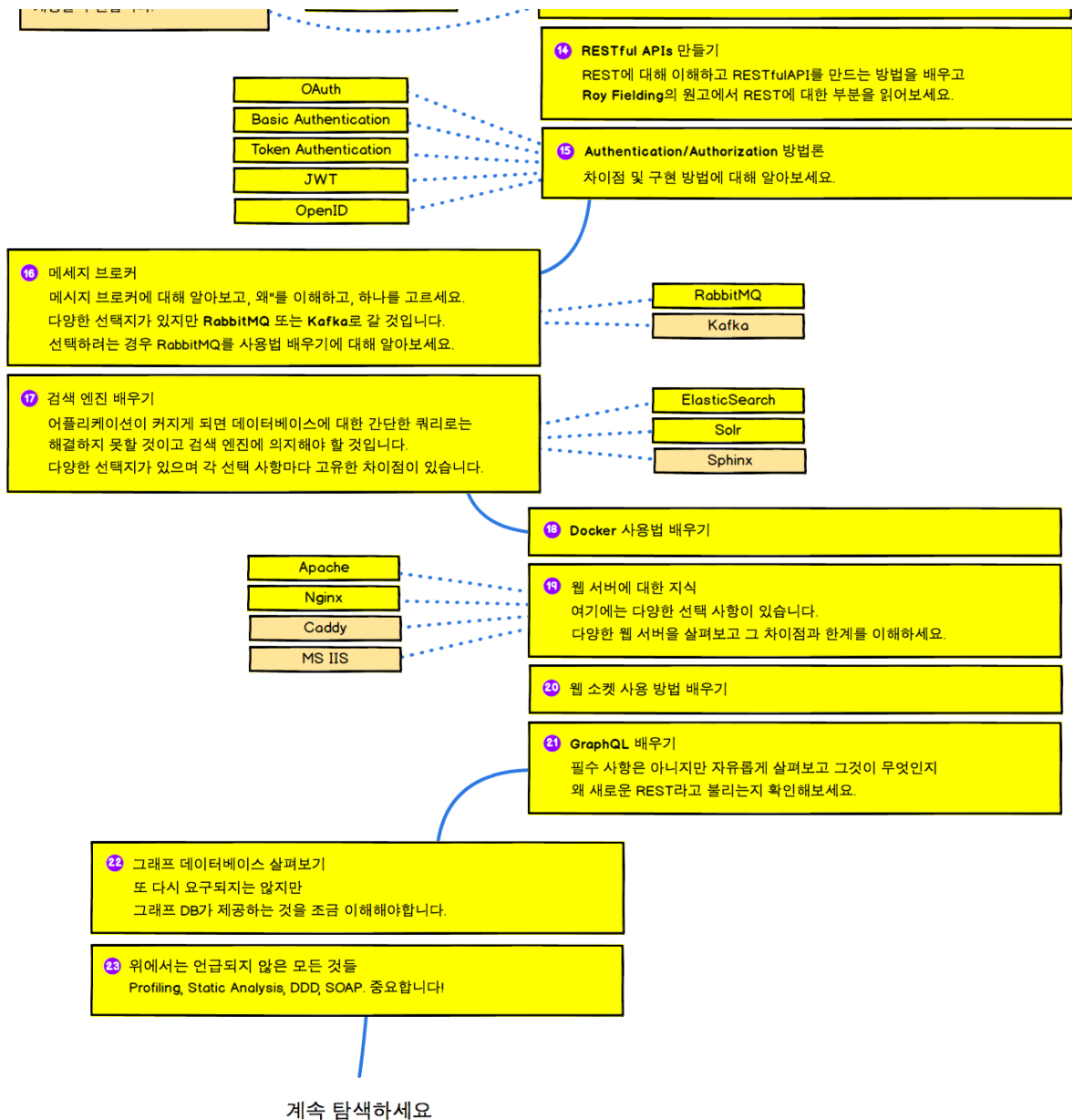
다른 언어 사용자의 경우, 선택한 언어에 적합한 것을 검색하여 찾습니다.



학습이 끝나면 11 단계에서  
 구축한 어플리케이션에  
 캐시를 구현합니다.

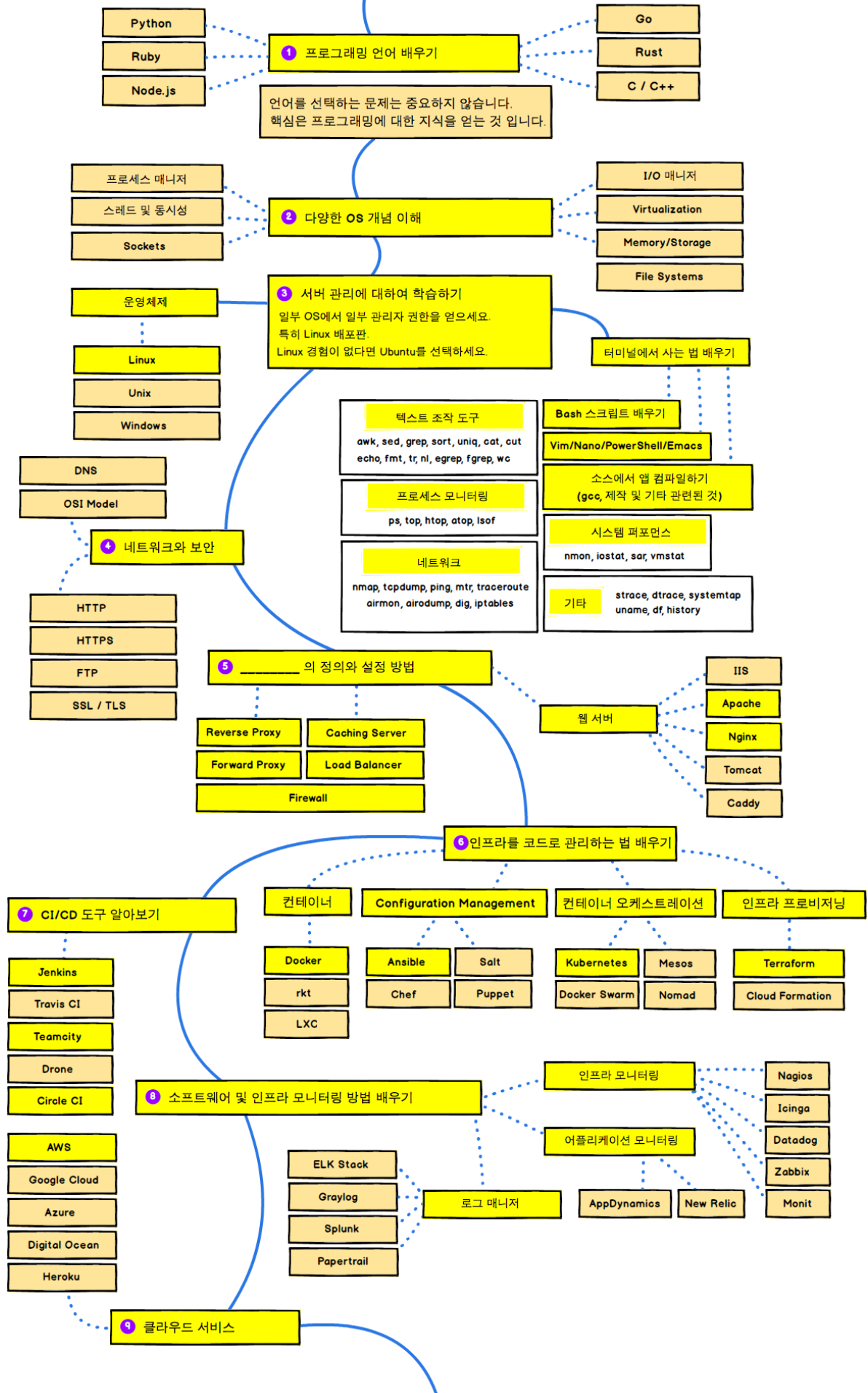


13 캐싱  
 Redis 또는 Memcached를 사용하여 앱 레벨 캐싱을 구현 방법 알아보기



## 데브옵스 로드맵

# 데브옵스



계속 탐색하세요

## 🔧 마무리

로드맵에 개선 할 수 있는 부분이 있다면 `Pull Request` 을 열고 이슈를 제출하세요.

또한 이 로드맵은 계속해서 개선할 예정이기 때문에 이 저장소를 다시 보길 원한다면 `Star` 또는 `Watch` 하여 재방문 할 수도 있습니다.

## 🙌 기여하기

로드맵을 업데이트하는 방법에 대한 [기여 문서](#) 를 살펴보세요.

- 개선사항은 pull request
- 논의사항은 issues
- 널리 알려주세요
- 어떠한 의견이든 제시해주세요 [Follow @kamranahmedse](#)

## 스폰서

- [Hackr.io - 최고의 온라인 프로그래밍 과정 및 튜토리얼 찾기 & 공유](#)

## 라이선스

License CC BY-NC-SA 4.0