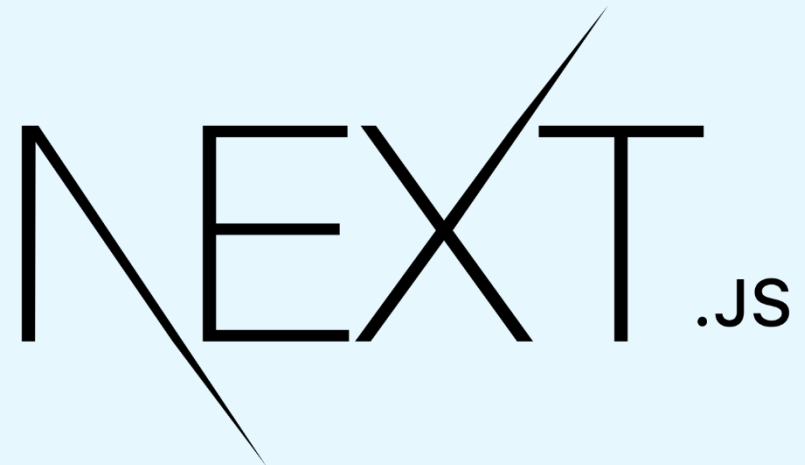
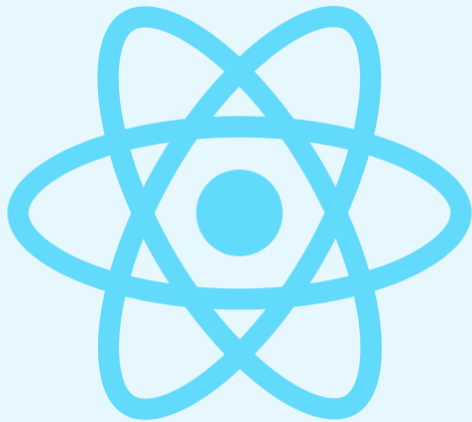


# 프론트엔드 토대 쌓기



# React + TypeScript + Next.js

- ✓ Create Next App을 이용해서 React + TypeScript + Next.js 환경을 구축해보자
- ✓ 앞으로 2학기 프로젝트에서 사용할 프론트엔드 프로젝트의 토대



# 학습 목표



- ✓ 공식 문서를 보면서 React의 기본 개념과 철학을 익힌다
- ✓ next.js를 공부하면서 SSR(Server Side Rendering)의 중요성을 익힌다
- ✓ TypeScript를 사용하면서 프론트엔드의 정적 타입 문법을 사용해본다
- ✓ 개발에 필요한 환경을 구축해보고 기초적인 웹 사용법을 익힌다

# 왜 React인가?



- ✓ React는 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리
- ✓ 국내에서는 이미 Angular나 Vue.js보다 시장 지배적인 위치를 차지
- ✓ 대부분의 개발 회사에서는 프론트엔드 개발을 위해서 React를 선택하는 분위기
- ✓ 취업을 준비하는 입장에서 React를 하지 않을 이유가 없다
- ✓ React는 입문자에게는 학습의 난이도가 있는 편이지만 공식 문서를 학습하면서 천천히 따라간다면 프로젝트에 사용할만큼 이해하는 것이 충분히 가능

# React vs Vue.js



## Vue.js가 무엇인가요?

Vue(/vju:/ 로 발음, **view** 와 발음이 같습니다.)는 사용자 인터페이스를 만들기 위한 **프로그레시브 프레임워크** 입니다. 다른 단일형 프레임워크와 달리 Vue는 점진적으로 채택할 수 있도록 설계하였습니다. 핵심 라이브러리는 뷰 레이어만 초점을 맞추어 다른 라이브러리나 기존 프로젝트와의 통합이 매우 쉽습니다. 그리고 Vue는 **현대적 도구** 및 **지원하는 라이브러리**와 함께 사용한다면 정교한 단일 페이지 응용프로그램을 완벽하게 지원할 수 있습니다.

## React

사용자 인터페이스를 만들기 위한 JavaScript 라이브러리

# 제어의 역전(Inversion Of Control)



- ✓ 객체의 대한 제어를 개발자가 하지 않고 프레임워크가 제어권을 갖는 것
- ✓ Vue.js를 프레임워크로, React를 라이브러리로 부르는 이유는 Vue.js의 경우에는 제어권이 프레임워크에 Vue.js에 있고, React의 경우에는 개발자에게 있기 때문
- ✓ 옳고 그른 것이 아니라 개발의 철학과 사상이 다른 것

# 제어의 역전(Inversion Of Control)



```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
);
```

```
JS  
new Vue({  
  data: {  
    a: 1  
  },  
  created: function () {  
    // `this` 는 vm 인스턴스를 가리킵니다.  
    console.log('a is: ' + this.a)  
  }  
})  
// => "a is: 1"
```

# 왜 TypeScript인가?



- ✓ 높은 생산성, 높은 생산성, 그리고 높은 생산성
- ✓ “아니, Ctrl + Space를 하면 어시스트가 된다고?”
- ✓ “브라우저에서 실행을 안 했는데 오류가 찾아진다고?”
- ✓ “이 족보 없는 언어에 족보가 생겼다고?”
- ✓ Type을 개발 단계에서 알 수 있다
- ✓ low level의 엄격함을 JavaScript에도 도입
- ✓ 100%의 호환성, 빠른 학습

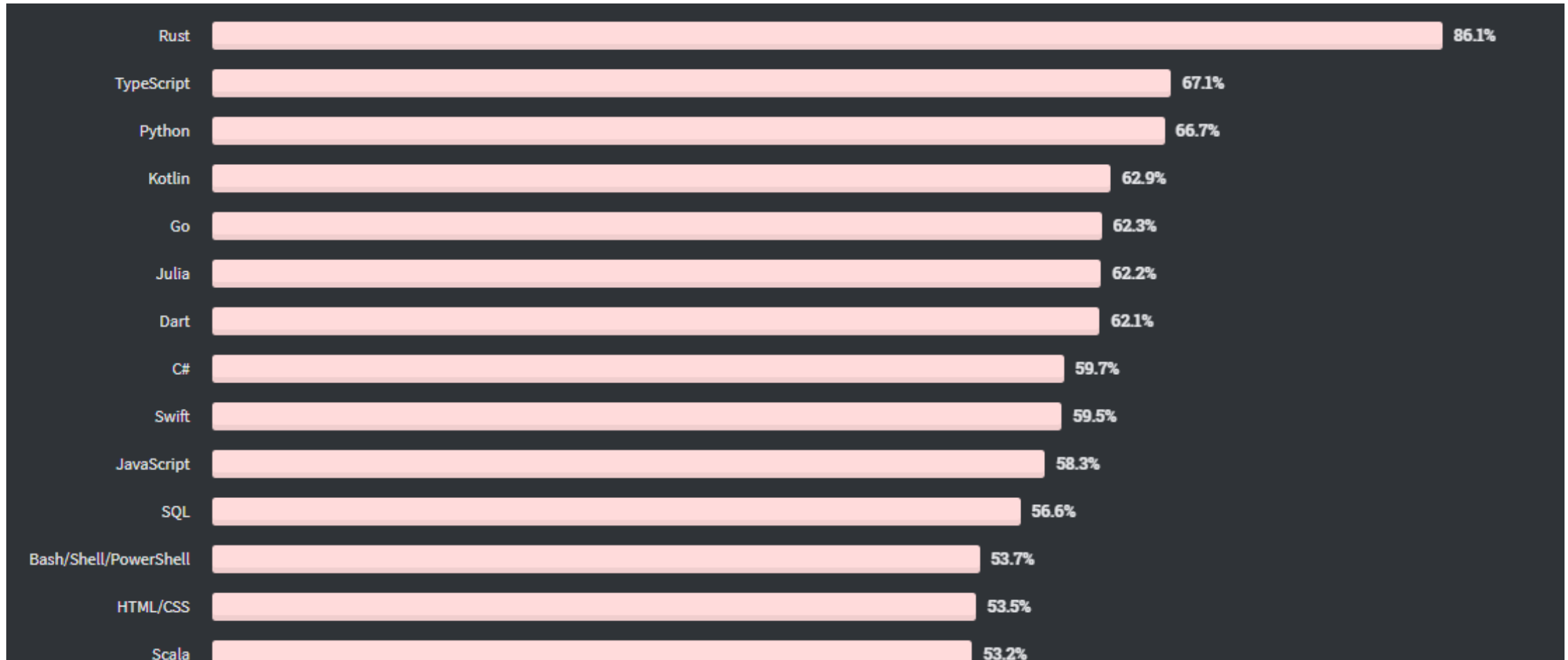


## 퀴즈의 정답을 아시는 분은...



- ✓ let some = '0';
- ✓ some = some \* 1;
- ✓ some = some + '0';
- ✓ some = some \* 1;

# 가장 사랑받는 언어 2위



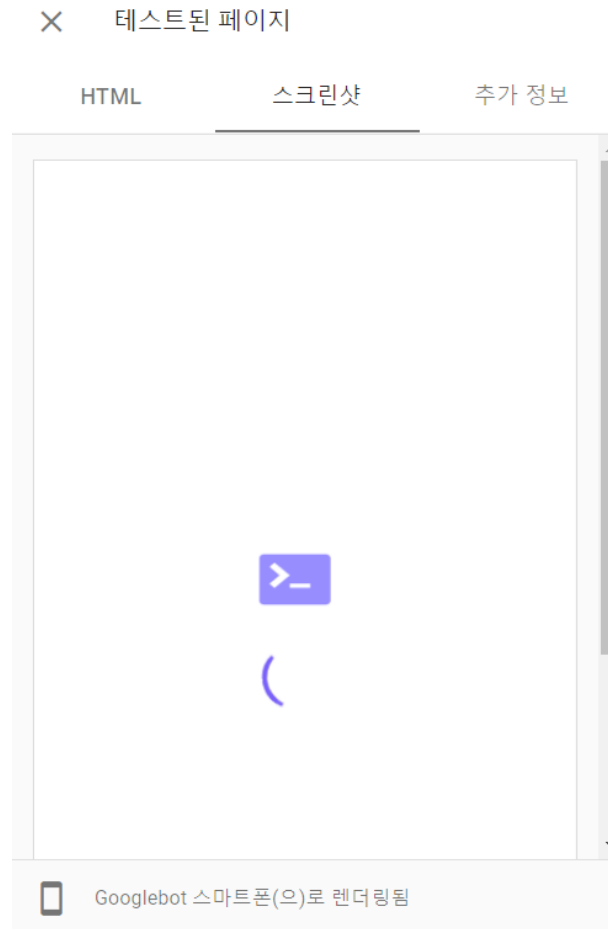
# Next.js



- ✓ 프로젝트에 필수적인 요소는 아니다
- ✓ 그렇지만 오늘은 한번 포함해서 해보자
- ✓ React에 Server Side Rendering을 가능하게 해주는 라이브러리

# 왜 다시 SSR인가?

- ✓ CSR은 화면이 로드 된 이후에 JavaScript 코드에 따라서 렌더링 되는 방식
- ✓ 검색엔진 봇은 언제 렌더링 될 지 예측하지 못하고, 제대로 페이지를 분석하지 못한다
- ✓ 검색의 상위 결과를 차지할 수 없다는 점은 대부분의 서비스에서 치명적인 문제가 된다



오늘 해야 하는 것



# Welcome to Next.js!

Get started by editing `pages/index.js`

## Documentation →

Find in-depth information about Next.js features and API.

## Learn →

Learn about Next.js in an interactive course with quizzes!

## Examples →

## Deploy →

## 교과서(공식 문서) 펴라



- ✓ 오늘 안에 과제를 할 시간은 충분할 것
- ✓ 시작하기에 앞서서 공식 문서를 천천히 읽어보자
- ✓ 모든 내용을 하루 만에 다 이해할 수는 없다
- ✓ 천천히 튜토리얼의 흐름을 따라가보면 된다

# 셸 설치



- ✓ OS에 관계 없이 셸을 사용할 수 있는 환경을 구성해보자
- ✓ 리눅스도 좋고 윈도우 환경도 괜찮음
- ✓ git bash나 파워셸 같은 것들을 활용해보자
- ✓ 이미 깔려있는 경우에는 추가 설치가 필요 없다



PowerShell

# node.js 설치



- ✓ 프로젝트 구동을 위해서 node.js를 설치해본다
- ✓ 셸에서 `node --version` 명령어를 통해서 설치 여부와 현재 버전을 확인 가능하다
- ✓ 되도록 LTS 버전을 설치해야 충돌이 적다



# Create Next App(--ts)



PowerShell ▾

```
npx create-next-app@latest --ts
```

```
# or
```

```
yarn create next-app --typescript
```

## Create Next App 2



```
npm run dev
```

Work like a charm!



# Welcome to Next.js!

Get started by editing `pages/index.js`

## Documentation →

Find in-depth information about Next.js features and API.

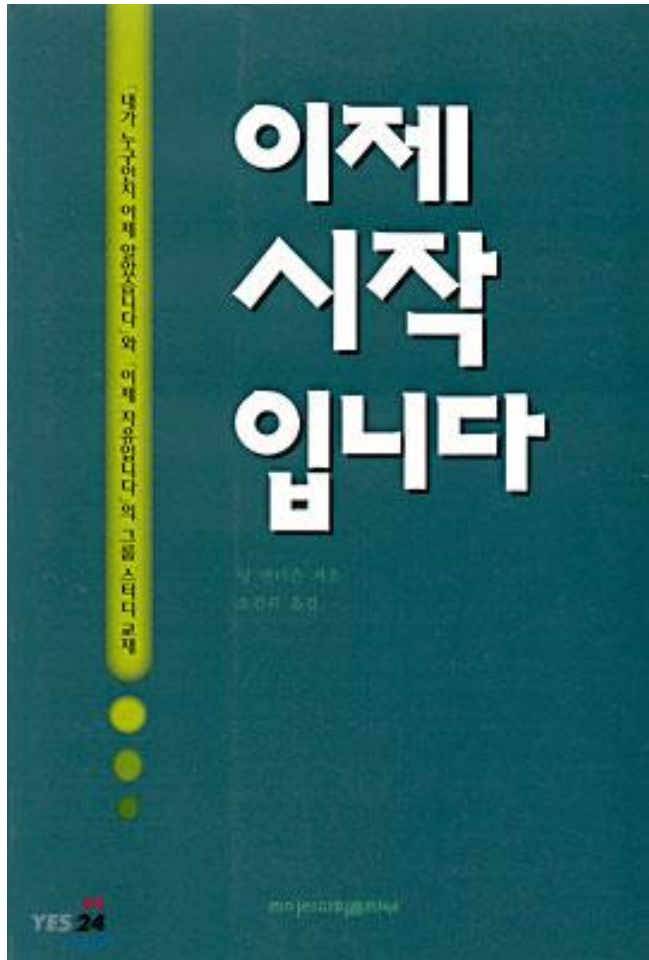
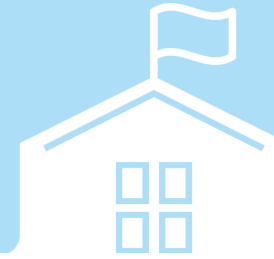
## Learn →

Learn about Next.js in an interactive course with quizzes!

## Examples →

## Deploy →

# 이게 끝인가요?



# IDE로 프로젝트를 열어봅시다



```
TypeError: this.cliEngineCtor is not a constructor
    at ESLintPlugin.invokeESLint (C:\Program Files\JetBr
    at ESLintPlugin.getErrors (C:\Program Files\JetBrain
    at ESLintPlugin.onMessage (C:\Program Files\JetBrain
    at Interface.<anonymous> (C:\Program Files\JetBrains
    at Interface.emit (events.js:375:28)
    at Interface._onLine (readline.js:364:10)
    at Interface._normalWrite (readline.js:509:12)
    at Socket.ondata (readline.js:216:10)|
    at Socket.emit (events.js:375:28)
```



# IDE로 프로젝트를 열어봅시다

- ✓ DE를 활용해서 프로젝트 폴더를 열어봅니다
- ✓ 운이 좋다면 아무런 오류 없이 동작할 수도 있겠지만, 많은 경우에는 오류에 직면하게 될 것입니다
- ✓ IDE에서 프로젝트를 편집하고 실행할 수 있도록 프로젝트 설정을 변경합니다
- ✓ 굉장히 심플한 프로젝트의 폴더 구조를 살펴봅니다

# 공식 문서를 따라가보자



- ✓ 이제 next.js의 바다로 나아갈 준비는 다 되었습니다
- ✓ 공식 문서의 튜토리얼과 예제들을 따라 하면서 하나하나 필요한 것들을 익혀봅시다
- ✓ 어디서부터 시작해야 할 지 막막하다면 Router(<https://nextjs.org/docs/api-reference/next/router>)부터!

질문 있으신 분?

