

# Do it! Vue.js 입문 - 07장

# Vuex

- 복잡한 규모의 애플리케이션에서 컴포넌트의 데이터 변화를 효과적으로 관리할 수 있는 라이브러리
- 상세한 튜토리얼은 아래 링크를 활용

[뷰엑스 튜토리얼 - Vuex와 State](#)

[뷰엑스 튜토리얼 - Getters와 Mutations](#)

[뷰엑스 튜토리얼 - Actions와 폴더 구조화](#)

# Vue Reactivity

- 뷰의 데이터가 변화할 때 자동으로 화면을 다시 그리는 라이브러리 특성
- 인스턴스가 생성될 때 data에 정의된 속성들은 모두 getter, setter 변환 작업을 거침
- 데이터의 변화가 감지되면 watcher에서 화면에 다시 그리라는 신호를 보냄

[Vue.js 공식 사이트 Vue Reactivity 설명](#)

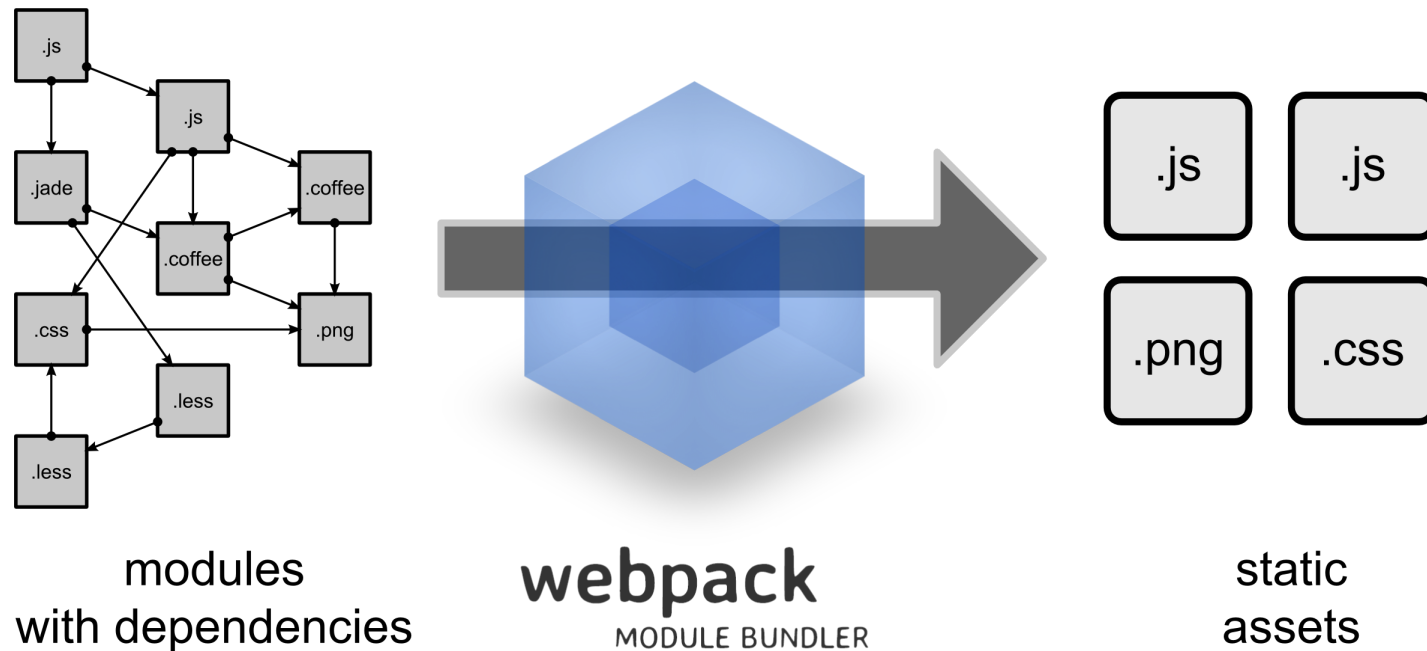
## 서버 사이드 렌더링

- Nuxt.js : 뷰에서 공식적으로 지원하는 서버 사이드 렌더링 라이브러리
- 클라이언트 사이드 렌더링과 서버 사이드 렌더링의 차이점을 인지하고 목적에 따라 적합한 방식을 적용

[Nuxt.js 공식 문서](#)

# 웹팩이란?

- 웹 애플리케이션 자원의 연관 관계를 파악하여 1개의 모듈로 변환해주는 도구
- 웹 성능 최적화, 모듈화와 밀접한 관계가 있음



## 웹팩의 주요 속성

- entry : 웹팩으로 빌드할 대상 파일들의 시작점
- output : 웹팩으로 빌드한 결과 파일의 설정 속성
- loader : 비 자바스크립트 파일을 자바스크립트로 변환해주는 기능
- plugin : 웹팩으로 빌드한 결과물에 추가 기능을 제공하는 속성
- resovle : 모듈 및 라이브러리를 해석하는 속성

## 웹팩 데브 서버

- 뷰 CLI 도구로 프로젝트를 생성할 때 사용하는 기본 웹팩 플러그인
- 인 메모리 방식으로 파일을 빌드하여 파일 시스템에서는 빌드한 파일이 확인되지 않음
- 기본적으로 파일의 변경에 따라 자동으로 다시 빌드

## ES6

- 최신 자바스크립트 스펙(문법)을 의미
- ECMAScript2015와 동일한 의미
- 자바스크립트로 효율적인 코드를 짤 수 있게 문법과 기능이 추가됨



## ES6 - const & let

- ES5의 변수 정의 방식 var의 업그레이드 버전
- const : 값의 재할당이 불가능
- let : 변수를 재 선언할 수 없음

```
const a = 10;  
a = 20; // Uncaught TypeError: Assignment to constant variable.  
  
let b = 10;  
let b = 20; // Uncaught SyntaxError: Identifier 'b' has already been declared
```

## ES6 - 화살표 함수

- ES5의 함수 정의 방식을 간소화

```
// ES5  
var add = function () {  
  // 로직 정의  
};  
  
// ES6  
var add = () => {  
  // 로직 정의  
}
```

## import & export

- 자바스크립트 파일을 모듈화 할 수 있는 기능
- 기존에 라이브러리로 제공되던 기능을 언어 차원에서 지원

```
// ./app/index.js
export const id = 10;

// ./main.js
import { id } from './app/index.js';
console.log(id); // 10
```

# NPM

- 노드 패키지 매니저의 약자로 웹 앱에 사용되는 라이브러리를 관리해주는 도구
- 전역 설치 : 라이브러리를 시스템 레벨에 설치
- 지역 설치 : 라이브러리를 해당 폴더에 설치
- 명령어를 사용자가 임의로 정의하여 사용할 수 있음

끝