

프론트엔드 개발자를 위한 Vue.js 시작하기

Vue.js </> 시작하기

ES2015



한국기술교육대학교
온라인평생교육원

학습내용

- ECMAScript Overview
- Babel

학습목표

- **ECMAScript**에 대해 설명할 수 있다.
- **Babel의 역할**에 대해 설명할 수 있다.

ECMAScript Overview

ECMAScript Overview

1 Overview

① 자바스크립트(Javascript) Overview

자바스크립트 Overview

- 1995년 넷스케이프에 의해 개발된 언어
- 자바스크립트는 브라우저에서 실행되는 HTML 페이지에 언어적 기능을 제공하기 위해 개발됨
- 간단한 연산, 유저 입력 데이터 유효성 검증 등이 주 역할
- 스크립트 언어이며 브라우저의 자바스크립트 엔진에 의해 해석되어 실행됨

ECMAScript Overview

1 Overview

② ECMAScript Overview

ECMAScript Overview

- ECMAScript 혹은 ES로 불림
- ECMA International에 의해 표준이 정의되는 언어
- ECMA-254 기술 규격에 따라 정의된 표준화된 스크립트 프로그래밍 언어임
- 자바스크립트를 표준화 하기 위해 만들어 졌음

ECMAScript Overview

ECMAScript Overview

2 ECMAScript 초기 버전

ES3

- 1999년 발표
- 초창기 자바스크립트
- IE8 시절의 자바스크립트

ES5

- 2009년 발표
- 엄격한 오류 검사를 위한 Strict mode 추가
- forEach, map, filter 등의 Collection 관련 함수 추가

ECMAScript Overview

3 자바스크립트의 역할 변화

- 브라우저에서 실행되는 웹 애플리케이션 개발 필요성
- 백엔드(Back-End) 웹 애플리케이션 개발 언어로 진화
- 자바스크립트를 대체할 목적의 다양한 언어 등장

Typescript

Dart

Coffee
Script

ECMAScript Overview

ECMAScript Overview

4 ECMAScript 진화

- 대규모 자바스크립트 애플리케이션 개발을 위한 다양한 기능을 제공하는 정식의 언어로 진화

ECMAScript Overview

4 ECMAScript 진화

1 ES6(ES2015)

- 2015년 발표
- 블록단위 스코프 제공
- 모듈화 지원
- Template Literal 지원

ECMAScript Overview

ECMAScript Overview

4 ECMAScript 진화

2

ES7(ES2016)

3

ES8(ES2017)

4

ES9(ES2018)

5

ES10(ES2019)

Babel

Babel

1 ECMA Script 문제점

1

ECMAScript의 모든 기능이 모든 브라우저에서 정상적으로 실행되지 않음

2

ECMAScript는 표준이며 ECMAScript로 작성된 코드를 실행시켜 주는 것은 브라우저의 자바스크립트 엔진임

Babel

1 ECMA Script 문제점

● ES2015

```
(function () {
  function some(x, y = 10, z = 20) {
    console.log(`x=${x}, y=${y}, z=${z}`)
  }
  some(1)//x=1, y=10, z=20
  some(1, 2)//x=1, y=2, z=20
  some(1, 2, 3)//x=1, y=2, z=3
})();
```

Babel

Babel

1 ECMA Script 문제점

ES2015

Chrome	정상 실행
Edge	정상 실행
IE 11	에러

Babel

1 ECMA Script 문제점

ES2015

```
import { data1, funA, Shape } from './module';
```

Chrome	에러
Edge	에러
IE 11	에러

Babel

Babel

2 Babel

① Babel

- Babel is a Javascript Compiler
- Use next generation Javascript, today



Babel

2 Babel

① Babel

- 개발자가 작성한 파일을 브라우저의 자바스크립트 엔진이 해석할 수 있는 **자바스크립트 코드로 변경**
- 자바스크립트 Compiler(Transpiler로 불리기도 함)
- ES2015 추가된 자바스크립트를 대부분의 브라우저에서 지원하는 **ES5로 변형**이 주목적임
- ECMAScript 이외에 웹에서 이용되는 다양한 형태의 코드를 **브라우저에서 인지할 수 있는 코드로 변형해 줌**

Babel

Babel

2 Babel

② Babel 설치

```
npm install --save-dev @babel/core @babel/cli @babel/preset-env
```

Babel

2 Babel

③ Babel 환경파일 작성

→ babel.config.json 혹은 babel.config.js

```
const presets = [
  [
    "@babel/env"
  ],
];

module.exports = { presets };
```

Babel

Babel

2 Babel

4 Babel Compile

```
node_modules\.bin\babel src --out-dir build
```

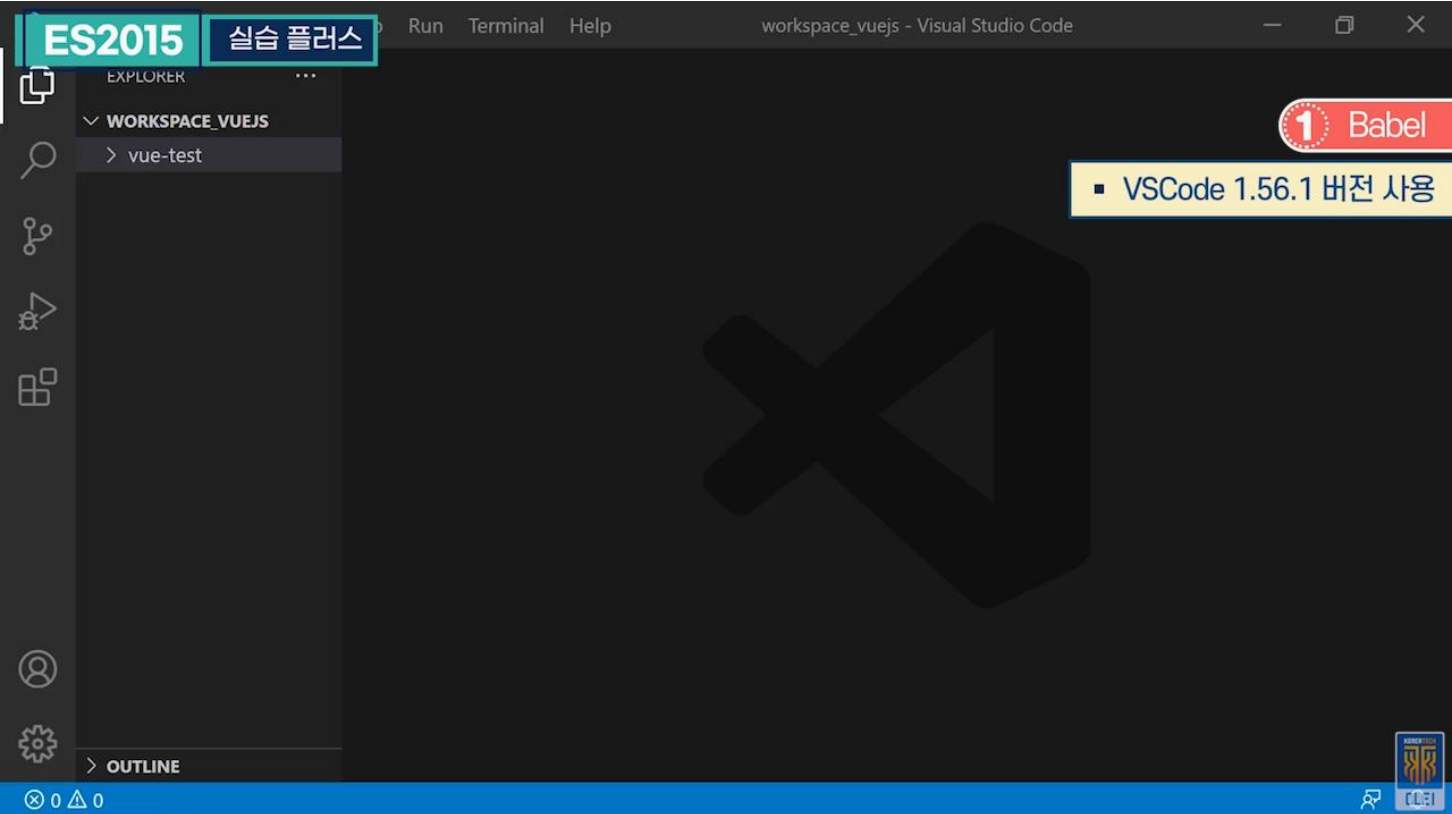
Babel

2 Babel

5 Compile 결과

```
(function () {
  function some(x) {
    var y = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : 10;
    var z = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : 20;
    console.log("x=".concat(x, ", y=").concat(y, ", z=").concat(z));
  }
  some(1); //x=1, y=10, z=20
  some(1, 2); //x=1, y=2, z=20
  some(1, 2, 3); //x=1, y=2, z=3
})();
```

실습 플러스



실습단계
ES2015와 Babel 이용 방법 테스트
ES2015를 테스트하기 위해 새로운 폴더를 만들어서 테스트
Vue 쪽에 프로젝트 포커스가 가지 않도록 새로운 폴더 만들기
ES2015 폴더 제작
ES2015 폴더를 누르고 서브 폴더 제작
ch3 폴더명으로 설정
ch3 폴더에 src 서브 폴더를 만들고 src 폴더 밑에 자바스크립트 파일 생성
src → main.js 자바스크립트 파일 생성
하나의 임의의 함수 정의, 함수가 선언되자마자 바로 호출한 경우라고 할 수 있음
some function 정의, 매개변수 세 개
두 번째, 세 번째 매개변수에 디폴트 값 설정
함수가 호출되었을 때 해당 값을 그대로 콘솔에 출력
매개변수 값을 그대로 콘솔에 출력한 코드
어디선가 some 함수를 Call 해주어야 함

실습 플러스

실습단계
매개변수 하나 주고 Call
매개변수 두 개를 주고 Call
매개변수 세 개를 주고 Call
함수 하나의 디폴트 값 명시, 싱글쿼트가 아니라 Tab 키 위의 backtick
문자열 내에 데이터를 바로 출력한 형태
자바스크립트 파일을 테스트하기 위해 HTML 파일 필요, 자바스크립트를 로딩하여 결과 확인
ch3 폴더에 HTML을 만들어서 테스트
ch3 폴더 → new file, HTML 파일 생성
파일명 : index.html
src → index.html, main.js
스크립트 로딩을 위해 스크립트 태그를 사용, 자바스크립트가 HTML에 의해 로딩되어 실행되게 처리
방금 만들어 놓은 main.js 지칭, src 밑의 main.js가 로딩되게 구성
HTML의 body 부분은 비워놓음
실행을 위해 서버가 필요함, HTML이 서버에서 동작해야 하고 브라우저로 테스트해줘야 함
Vue 프로젝트와 상관없이 ES2015를 살펴보기 위한 것
VS Code에 경량 서버를 설치하고 이용
Extensions → live 검색
쉽게 이용할 수 있는 경량 서버인 Live Server를 설치하고 이용할 수 있음
VS Code 하단 → Go Live, Live Server를 이용해서 HTML 로딩
1. HTML 페이지가 있는 상태에서 하단 Go Live 버튼 선택 2. HTML 페이지에서 오른쪽 마우스 버튼 눌러서 Open with Live Server 메뉴 이용
Live Server에서 HTML을 실행시키면 브라우저가 자동으로 뜸, 컴퓨터에 디폴트로 설정되어 있는 브라우저가 뜨게 됨
콘솔 출력 값을 확인하는 테스트
마우스 오른쪽 버튼 눌러서 콘솔창을 띄워야 함
작성한 자바스크립트 코드가 크롬 브라우저에서 정상적으로 수행됨
자바스크립트 코드가 모든 클라이언트의 브라우저에서 정상적으로 나온다고 볼 수 없음
어떤 브라우저에서 코드 문법이 맞지 않아 에러가 날 수 있음, 그 부분을 해결하기 위해 필요한 것이 Babel
작성했던 코드를 Babel로 컴파일 시켜서 서비스를 진행하는 형태

실습 플러스

실습단계
npm 명령으로 Babel 추가 설치가 필요함
설치하고자 하는 것을 나열하여 작성
Babel이 npm repository 서버로부터 다운로드 되어서 설치가 됨
Babel의 환경파일 명시 필요 → ch3 폴더에 Babel.config.js라는 이름의 파일을 다시 만듦
Babel의 환경파일 → Babel이 어떤 작업을 해야 하는지 명시함
js 파일을 Babel로 컴파일 시켜줘야 함
Babel 명령어를 이용하여 src의 파일을 컴파일 시켜 build 폴더에 컴파일 된 것을 명시할 것
build라는 폴더에 컴파일 된 결과를 출력
ES2015에서 작성된 코드가 모든 브라우저에서 수행된다고 보장할 수 없기 때문에 Babel을 이용하여 컴파일 시킨 후, 서비스하는 형태