



Chap26 ES6 함수의 추가 기능

▼ 함수의 구분

- ES6는 함수를 사용 목적에 따라 3가지 종류로 명확히 구분했다.

ES6 함수의 구분	constructor	prototype	super	arguments
일반함수	O	O	X	O
메서드	X	X	O	O
화살표 함수	X	X	X	X

▼ 메서드

ES6부터 명확한 정의가 생김.

메서드 : 메서드 축약 표현으로 정의된 함수만을 의미

▼ 화살표 함수

function 키워드 대신 화살표를 사용하여 기존의 함수 정의 방식보다 간략하게 함수를 정의

▼ 일반함수와 화살표함수 차이

- 화살표 함수는 인스턴스를 생성할 수 없는 non-constructor다.
- 중복된 매개변수 이름을 선언할 수 없다.
- 화살표 함수는 함수 자체의 `this`, `arguments`, `super`, `new.target` 바인딩을 갖지 않는다.

▼ this

화살표 함수는 함수 자체의 `this` 바인딩을 갖지 않는다. 따라서 화살표 함수 내부에서 `this`를 참조하면 상위 스코프의 `this`를 그대로 참조한다. 이를 `lexical this`라고 한다.

▼ super

화살표 함수는 함수 자체의 `super` 바인딩을 갖지 않는다. 따라서 화살표 함수 내부에서 `super`를 참조하면 `this`와 마찬가지로 상위 스코프의 `super`를 참조한다.

▼ Rest 파라미터

이름 앞에 세개의 점...을 붙여서 정의한 매개변수

함수에 전달된 인수들의 \square 고록을 배열로 전달받는다.