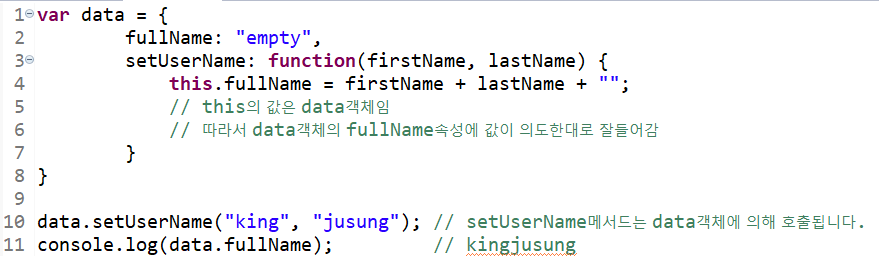
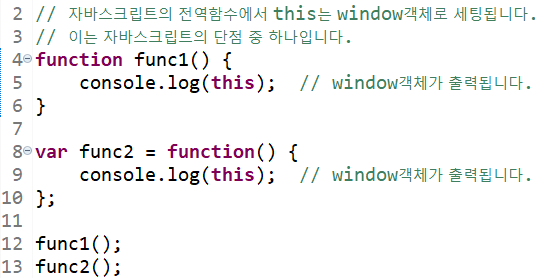
< 콜백 함수 사용 시 유의점 >

< 코드 1 > this 키워드 예제1



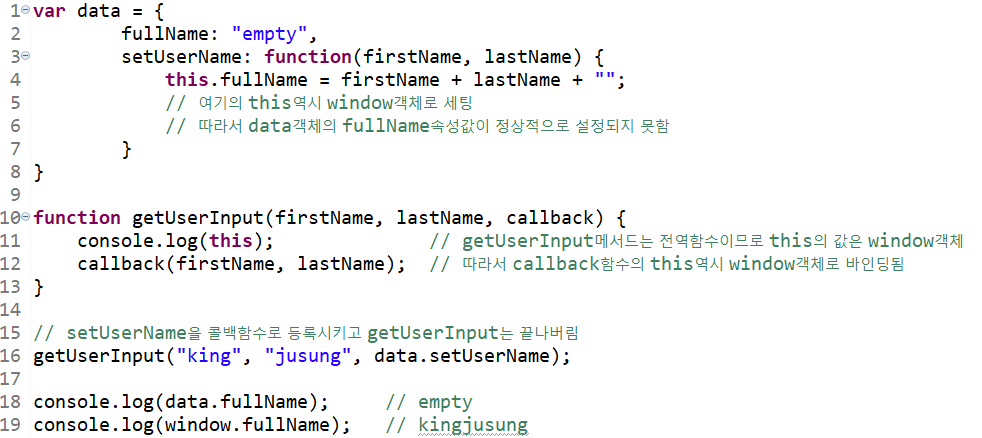
※ this는 자신을 호출한 객체가 저장되는 변수입니다. 아시다시피 this키워드의 값은 어디에 정의되어 있는지에는 영향을 받지 않습니다(즉, 렉시컬 스코프의 영향을 받지 않습니다). this키워드는 어떻게 호출되었는지에 따라 그 값이 결정됩니다. 그리고 위 코드의 setUserName메서드는 data객체에 의해 호출되었기 때문에 setUserName메서드 내부의 this의 값은 data객체가 됩니다. 따라서 data의 fullName값은 의도한대로 세팅이 됩니다.

< 코드 2 > this 키워드 예제2



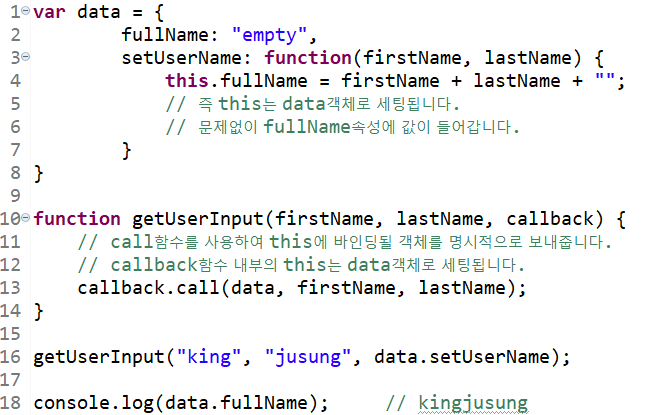
※ 자바스크립트의 단점 중 하나인 코드입니다. 자바스크립트의 전역함수 내부의 this는 window객체로 세팅됩니다. 사실 이 자체로는 큰 문제가 되지 않지만 만약 전역함수에서 콜백함수를 호출할 경우에는 문제가 발생합니다.

< 코드 3 > 전역함수 내부의 콜백함수 1



※ 위의 코드에서 보실 수 있듯이 getUserInput이라는 전역함수 내부에서 콜백함수를 호출할 경우, 콜백함수 내부의 this역시 window객체로 바인딩되는 문제가 발생합니다. 따라서 data객체의 fullName속성의 값이 정상적으로 설정되지 않습니다. 엉뚱한 window객체의 속성으로 들어갈 뿐이죠. 위와 같은 문제는 콜백함수 호출 시 빈번하게 발생되는 문제입니다.

< 코드 4 > 전역함수 내부의 콜백함수 2



※ 해결책은 call또는 apply함수를 활용하는 것입니다. call, apply함수는 특정함수를 호출 시 그 함수 내부의 this를 명시적으로 바인딩시키는 함수입니다. 즉 어떤 함수를 호출할 때 그 함수 내부의 this를 마음대로 세팅할 수 있다는 뜻입니다. 위 코드에서는 callback함수를 call함수를 사용하여 호출함으로서 callback함수 내부의 this를 data객체로 강제로 바인딩 시킴으로서 < 코드 3 > 과 같은 문제를 해결하였습니다.