스코프

스코프란? 변수 이름, 함수 이름, 클래스 이름과 같은 식별자가 본인이 선언된 위치에 따라 다른 코드에서 자신이 참조될 수 있을지 없을지 결정되는 것.

스코프 체인? Inner->outer->전역 순서대로 변수를 참조한다. 단방향성(하위>상위, 그 반대는 성립 불가).

블록 레벨 스코프: 모든 코드 블록이 지역 스코프를 생성.

함수 레벨 스코프: var 키워드로 선언된 변수는 오로지 함수 코드 블록만을 함수 지역 스코프로 인정.

Es6 이후로 let, const 키워드를 통해 블록 레벨 스코프를 이용한다.

동적 스코프: 함수가 호출되는 시점에 따라 상위 스코프가 결정되는 것.

정적 스코프: 렉시컬 스코프. 함수가 정의되는 시점에서 결정되는 것. 함수가 생성되면서 내부 슬롯에 상위 스코프 참조를 저장한다.

함수 호출 > 실행 컨텍스트 생성 > 실행 컨텍스트 스택에 push > 렉시컬 환경 생성 > 실행 컨텍스트 스택에서 pop

클로져: 자유 변수가 있고, 본인의 외부 함수보다 더 오래 살아있는 경우 중첩 함수를 부르는 이름.

\*자유 변수: 클로져에 의해 참조되는 지역 변수

\*렉시컬 환경: 포함하는 식별자, 식별자에 바인딩 된 값, 상위 렉시컬 환경에 대한 참조.

\*중첩 함수: 함수 내부 함수.

\*외부 함수: 중첩 함수를 가진 함수.