21장. 빌트인 객체

21.1 자바스크립트 객체의 분류

▼ 자바스크립트 객체 종류

1. 표준 빌트인 객체
2. 호스트 객체
3. 사용자 정의 객체

21.2 표준 빌트인 객체

▼ 표준 빌트인 객체란?

모두 인스턴스를 생성할 수 있는 생성자 함수 객체

생성자 함수 객체인 표준 빌트인 객체는 프로토타입 메서드와 정적 메서드를 제공,

생성자 함수 객체가 아닌 표준 빌트인 객체는 정적 메서드만 제공

21.3 원시값과 래퍼 객체

▼ 원시값

원시값은 객체가 아니어서 프로퍼티,메서드를 가질 수 없지만 원시값인 문자열은 객체처럼 동작

→ 마침표 표기법(또는 대괄호 표기법)으로 접근하면 자바스크립트 엔진이 일시적으로 원시값을 연관된 객체로 변환해 주기 때문

▼ 래퍼 객체란?

문자열, 숫자, 불리언 값에 대해 객체처럼 접근하면 생성되는 임시 객체

문자열, 숫자, 불리언, 심벌 이외의 원시값 즉, null과 undefined는 래퍼 객체를 생성하지 않는다.

→ null과 undefined 값을 객체처럼 사용하면 에러 발생

21.4 전역 객체

▼ 전역 객체란?

코드가 실행되기 이전 단계에 자바스크립트 엔진에 의해 어떤 객체보다도 먼저 생성되는 특수한 객체이며, 어떤 객체에도 속하지 않는 최상위 객체

▼ 전역 객체가 최상위 객체라는 것은?

전역 객체 자신은 어떤 객체의 프로퍼티도 아니며 객체의 계층적 구조상 표준 빌트인 객체와 호스트 객체를 프로퍼티로 소유한다는 것

▼ 전역 객체의 특징은?

1. 개발자가 의도적으로 생성할 수 없다.

   = 전역 객체를 생성할 수 있는 생성자 함수가 제공되지 않는다.

2. 전역 객체의 프로퍼티를 참조할 때 window(또는 global)를 생략할 수 있다.
3. 전역 객체는 모든 표준 빌트인 객체를 프로퍼티로 가지고 있다.
4. 자바스크립트 실행 환경에 따라 추가적으로 프로퍼티와 메서드를 갖는다.
5. var 키워드로 선언한 전역 변수와 선언하지 않는 변수에 값을 할당한 암묵적 전역, 그리고 전역 함수는 전역 객체의 프로퍼티가 된다.
6. let이나 const 키워드로 선언한 전역 변수는 전역 객체의 프로퍼티가 아니기에 보이지 않는 개념적인 블록 내에 존재하게 된다.
7. 브라우저 환경의 모든 자바스크립트 코드는 하나의 전역 객체 window를 공유한다.

▼ 빌트인 전역 프로퍼티란?

전역 객체의 프로퍼티를 의미 → 주로 애플리케이션 전역에서 사용하는 값을 제공

▼ Infinity

무한대를 나타내는 숫자값 Infinity를 갖는다.

▼ NaN

NaN 프로퍼티는 숫자가 아님을 나타내는 숫자값 NaN을 갖는다.

▼ undefined

원시 타입 undefined를 값으로 갖는다.

▼ 빌트인 전역 함수란?

애플리케이션 전역에서 호출할 수 있는 빌트인 함수로서 전역 객체의 메서드

▼ eval

자바스크립트 코드를 나타내는 문자열을 인수로 전달받는다.

이때, 전달받은 문자열 코드가 표현식이라면 eval 함수는 문자열 코드를 런타임에 평가하여 값을 생성

표현식이 아닌 문이라면 eval 함수는 문자열 코드를 런타임에 실행

→ 인수로 전달받은 문자열 코드가 여러 개의 문인 경우 마지막 결과값을 반환

eval 함수는 기존의 스코프를 런타임에 동적으로 수정한다.

strict mode에서 eval 함수는 기존의 스코프를 수정하지 않고 eval 함수 자신의 자체적인 스코프를 생성

인수로 전달받은 문자열 코드가 let, const 키워드를 사용한 변수 선언문이라면 암묵적으로 strict mode가 적용된다.

▼ eval 함수 단점

eval 함수를 통해 사용자로부터 입력받은 콘텐츠를 실행하는 것은 보안에 매우 취약함

자바스크립트 엔진에 의해 최적화가 수행되지 않으므로 일반적인 코드 실행에 비해 처리 속도가 느리가

→ eval 함수의 사용은 금지해야 한다.

▼ isFinite

전달받은 인수가 정상적인 유한수인지 검사하여 유한수이면 true를 반환, 무한수이면 false를 반환

만약 전달받은 인수의 타입이 숫자가 아닌 경우, 숫자로 타입을 변환한 후 검사를 수행

→ 이때 인수가 NaN으로 평가되는 값이라면 false를 반환

▼ isNaN

전달받은 인수가 NaN인지 검사하여 그 결과를 불리언 타입으로 반환한다.

전달받은 인수의 타입이 숫자가 아닌 경우 숫자로 타입을 변환한 후 검사를 수행

▼ parseFloat

전달받은 문자열 인수를 부동 소수점 수사, 즉 실수로 해석하여 반환

▼ parseInt

전달받은 문자열 인수를 정수로 해석하여 반환

전달받은 인수가 문자열이 아니면 문자열로 변환한 다음, 정수로 해석하여 반환

▼ encodeURI/decodeURI

encodeURI 함수는 완전한 URI를 문자열로 전달받아 이스케이프 처리를 위해 인코딩한다.

URI는 인터넷에 있는 자원을 나타내는 유일한 주소

인코딩은 URI의 문자들을 이스케이프 처리하는 것을 의미

이스케이프 처리는 네트워크를 통해 정보를 공유할 때 어떤 시스템에서도 읽을 수 있는 아스키 문자 셋으로 변환하는 것

decodeURI 함수는 인코딩된 URI를 인수로 전달받아 이스케이프 처리 이전으로 디코딩한다.

▼ encodeURIComponent/decodeURIComponent

encodeURIComponent 함수는 URI 구성 요소를 인수로 전달받아 인코딩

인코딩은 URI의 문자들을 이스케이프 처리하는 것을 의미

decodeURIComponent 함수는 매개변수로 전달된 URI 구성 요소를 디코딩한다.

▼ encodeURIComponent과 encodeURI 차이

encodeURIComponent 함수: 인수로 전달된 문자열을 URI의 구성요소인 쿼리 스트링의 일부로 간주한다.

encodeURI 함수: 매개변수로 전달된 문자열을 완전한 URI 전체라고 간주한다.

▼ 암묵적 전역

선언하지 않는 식별자에 값을 할당하면 전역 객체의 프로퍼티가 되기 때문이다.