



Ch.9 타입 변환과 단축 평가

▼ 🌟🌟 명시적 타입 변환(타입 캐스팅)

개발자가 의도적으로 값의 타입을 변환하는 것

▼ 🌟🌟🌟 암묵적 타입 변환(타입 강제 변환)

개발자의 의도와 상관없이 표현식을 평가하는 도중에 자바스크립트 엔진에 의해 암묵적으로 타입이 자동 변환되는 것.

p110(e144) 암묵적 타입 변환이 발생하면 원시타입 중 하나로 타입을 자동 변환한다.

타입변환이란 기존 원시 값을 사용해 다른 타입의 새로운 원시 값을 생성하는 것이다.

p109(e143)

자바스크립트 엔진은 표현식 $x + ' '$ 를 평가하기 위해 x 변수의 숫자 값을 바탕으로 새로운 문자열 값 '10'을 생성하고 이것으로 표현식 '10+'를 평가한다. 이때 암묵적으로 생성된 문자열 '10'은 x 변수에 재할당되지 않는다.

즉, 암묵적 타입 변환은 기존 변수 값을 재할당 하여 변경하는 것은 아니다. 자바스크립트 엔진은 표현식을 에러 없이 평가하기 위해 피연산자의 값을 암묵적 타입 변환해 새로운 타입의 값을 만들어 단 한 번 사용하고 버린다.

▼ 숫자타입으로 암묵적 타입 변환

p113(e147) 빈 문자열(' '), 빈 배열([]), **null**, **false**는 0으로 변환된다.

객체와 빈 배열이 아닌 배열, **undefined**는 변환되지 않아 NaN이 된다는 것에 주의하자.

▼ 🌟🌟 Falsy값

자바스크립트 엔진은 불리언 타입이 아닌 값을 Truthy값, Falsy값으로 구분한다.

Falsy값

- **false**
- **undefined**
- **null**
- **0, -0**
- **NaN**
- **''(빈 문자열)**

+p114(e148) 예제 09-13 빈문자열이 아닌 문자열은 Truthy 값이다. '0'은 Truthy.

Falsy가 아닌 모든 값은 Truthy다. 따라서 빈 배열, 빈 객체는 Truthy 다.
p115(e149) 예제 09-13

▼ 함수

p115(2149)

어떤 작업을 수행하는 데 필요한 문들의 집합을 정의한 코드 블록이다.

▼ 🌟🌟🌟 명시적 타입 변환 방법

- 표준 빌트인 생성자 함수를 new 연산자 없이 호출 (String, Number, Boolean)
- 빌트인 메서드 사용
- 암묵적 타입 변환 이용

+p115(e149) **표준 빌트인 생성자 함수**와 **표준 빌트인 메서드**는 자바스크립트에서 기본 제공하는 함수다. 표준 빌트인 생성자 함수는 객체를 생성하기 위한 함수이며 new 연산자와 함께 호출한다. 표준 빌트인 메서드는 자바스크립트에서 기본 제공하는 빌트인 객체의 메서드다.

Object.prototype.toString메서드

Object {

```
.....
toString: f
....
}
```

▼ 문자열 타입으로 명시적 변환

- String 생성자 함수를 new 연산자 없이 호출 p115(e149) 예제 09-14
- Object.prototype.toString 메서드를 사용
- 문자열 연결 연산자 이용

▼ 숫자 타입으로 명시적 변환

- Number 생성자 함수를 new 연산자 없이 호출 p116(e150) 예제 09-15
- parseInt, parseFloat 함수 사용(문자열만 숫자 타입으로 변환 가능)
- + 단항 산술 연산자 이용
- * 산술 연산자 이용

▼ 불리언 타입으로 변환

- Boolean 생성자 함수를 new 연산자 없이 호출 p117(e151) 예제 09-16
- ! 부정 논리 연산자를 두 번 사용

▼ 🌟🌟🌟 단축 평가

단축평가는 **표현식을 평가하는 도중에 평가 결과가 확정된 경우 나머지 평가 과정을 생략**하는 것을 말한다.

논리곱(&&) 연산자와 논리합(||) 연산자는 논리 연산의 결과를 결정하는 피연산자를 타입 변환하지 않고 그대로 반환한다. p119(e153)

단축평가를 사용하면 if문을 대체할 수 있다.

단축평가는 다음과 같은 상황에서 유용하다. p121(e155)

- 객체를 가리키기를 기대하는 변수가 **null** 또는 **undefined**가 아닌지 확인하고 프로퍼티를 참조할 때
- 함수 매개변수에 기본값을 설정할 때
(ES6부터는 매개변수에 **=** 로 바로 기본값 지정 가능)

▼ 🌟 옵셔널 체이닝 연산자

?. p122(e156)

좌항의 피연산자가 **null** 또는 **undefined**인 경우 **undefined**를 반환하고 그렇지 않으면 우항의 프로퍼티 참조를 이어간다.

+프로퍼티 참조: 변수를 통해 변수값을 참조하듯 객체의 프로퍼티에 접근해 프로퍼티 값을 참조하는 것.

▼ 🌟🌟 null병합 연산자

??

좌항의 피연산자가 **null** 또는 **undefined**인 경우 우항의 피연산자를 반환하고 그렇지 않으면 좌항의 피연산자를 반환한다.