

Ch5 표현식과 문

값

값은 식이 평가되어 생성된 결과를 말한다.

모든 값은 데이터 타입을 가지고, 메모리에 2진수 (비트의 나열) 로 저장된다.

```
//변수에는 10 +20이 평가되어 생성된 숫자 값 30이 할당된다.  
var sum = 10 + 20;
```

리터럴

리터럴은 사람이 이해할 수 있는 문자 또는 약속된 기호를 사용해 값을 생성하는 표기법을 말한다.

자바스크립트 엔진은 코드가 실행되는 시점인 런타임에 리터럴을 평가해 값을 생성한다.

표현식

표현식은 값으로 평가될 수 있는 문으로 표현식이 평가되면 새로운 값을 생성하거나 기존 값을 참조한다.

→ 리터럴도 표현식이다.

```
// 리터럴 표현식  
10  
'Hello'  
  
// 식별자 표현식(선언이 이미 존재한다고 가정)  
sum  
person.name  
arr[1]  
  
// 연산자 표현식  
10 + 20  
sum = 10  
sum !== 10  
  
// 함수/메서드 호출 표현식(선언이 이미 존재한다고 가정)  
square()  
person.getName()
```

문

문은 프로그램을 구성하는 기본 단위이자 최소 실행 단위다. 문은 여러 토큰으로 구성되는데 **토큰**이란 문법적인 의미를 가지며 문법적으로 더 이상 나눌 수 없는 코드의 기본 요소를 의미한다.

문은 명령문이라고도 부르고 선언문, 할당문, 조건문, 반복문 등으로 구분할 수 있다.

```
// 변수 선언문  
var x;  
  
// 표현식 문(할당문)  
x = 5;  
  
// 함수 선언문  
function foo () {}
```

```
// 조건문
if (x > 1) { console.log(x); }

// 반복문
for (var i = 0; i < 2; i++) { console.log(i); }
```

세미콜론과 세미콜론 자동 삽입 기능

- **세미콜론(;)은** 문의 종료를 나타내기 때문에 문을 끝낼 때는 세미콜론을 붙여야 한다.
- 단, 0개 이상의 문을 중괄호로 묶은 코드 블록 뒤에는 세미콜론을 붙이지 않는다.
 - 자체 종결성을 갖기 때문
- 문의 끝에 붙이는 세미콜론은 옵션이므로 생략 가능하다.
 - 세미콜론 자동 삽입 기능이 암묵적으로 수행되기 때문
 - 동작이 다를 수 있음

표현식인 문과 표현식이 아닌 문

표현식인 문과 표현식이 아닌 문을 구별하는 방법은 변수에 할당해 보는 것인데 표현식인 문은 값으로 평가되므로 변수에 할당할 수 있지만 표현식이 아닌 문은 값으로 평가할 수 없으므로 변수에 할당하면 에러가 발생한다.

※ 완료 값

크롬 개발자 도구에서 표현식이 아닌 문을 실행하면 언제나 `undefined`를 출력하는데 이를 완료 값이라 한다. 완료 값은 표현식의 평가 결과가 아니므로 다른 값과 같이 변수에 할당할 수 없고 참조할 수도 없다.