

함수(**function**)

함수

- function
 - 특정 기능을 처리하는
자바스크립트 코드 묶음
- 함수 형태: [코드 1](#)

함수 구성 요소

- **function** 키워드 Keyword 코드 1
- 함수 이름
- 파라미터 Parameter
 - 매개 변수, 인자, 아규먼트로도 부름
 - 강좌에서는 파라미터로 표기
 - 파라미터 작성은 선택
- 함수 Body
 - 중괄호{} 안에 작성한 코드
 - 함수 코드, 소스 텍스트로도 부름
 - 강좌에서는 함수 코드로 표기
 - 함수 코드 작성은 선택

함수 이름 규칙

- 첫 문자
 - 영문자, \$, 언더바(_): 사용 가능
 - 숫자, &, *, @, +: 사용 불가 [코드 1](#)
- 함수 이름 작성 권장
 - 함수 코드를 읽지 않더라도
 - 함수 이름과 파라미터로 기능을 알 수 있도록
 - 시맨틱(의미, 뜻)을 부여하여 작성

함수 이름 관례

- `calculatePoint()`처럼 동사로 시작
 - 포인트를 계산한다. [코드 1](#)
- 두 개 이상의 단어를 사용할 때
 - 두 번째 단어부터 명사 사용
 - 명사의 첫 문자를 대문자로 사용
 - **CamelCase** 형태라고 부름
- 동사 + 명사 형태로 동적인 모습
 - 강좌에서는 설명 편리를 위해 간단하게 작성합니다.

함수 호출

호출받는 함수

- 함수는 호출되어야 실행됩니다. [코드 1](#)
- 호출받는 함수
 - 함수가 호출되었을 때 실행되는 함수
 - 함수라고 하면 호출받는 함수를 지칭
- 파라미터
 - 호출한 함수에서 넘겨준 값을 받음
 - (one, two)처럼
소괄호() 안에 파라미터 이름 작성

함수 호출

- `setValue()` 형태로 호출 [코드 1](#)
 - 함수 이름과 소괄호()`()`를 작성
 - `setValue`만 작성하면 호출되지 않음
- 파라미터
 - 호출된 함수에 넘겨줄 값 작성
 - `setValue(10, 20)`처럼 소괄호()`()` 안에 작성
 - JS에서 지원하는 타입 작성
 - 콤마(`,`)로 구분하여 다수 작성 가능

return

- 형태:
return 표현식opt;
- 표현식의 평가 결과 반환 [코드 1](#)
- return 또는 표현식을 작성하지 않으면
undefined 반환 [코드 2](#)
- return과 표현식을 한 줄에 작성 [코드 3](#)

주식 작성 목적

주석 작성의 궁극적인 목적

- 코드가 목적을 달성하기 위한
 - 기능, 시나리오 등을 **생각하고 정리하는 것**
- 이것을 문법에 맞추어 작성한 것이 코드
- 코드를 작성하기 전에
 - 우선 먼저 정리한 생각을 주석으로 작성
 - 어렵지만 처음 배울 때부터 습관화
- 주석 작성 기준
 - 코드의 목적, 결과가 미치는 영향을 작성
 - 다른 사람이 알 수 있도록 자세하게 작성

주석 사례

```
/**
 * @function calculateAmount
 * 1. 파라미터로 받은 수량에 단가를 곱해
 *   금액을 계산하고 결과 값을 반환한다.
 * 2. 수량과 단가의 0 체크는
 *   함수를 호출하기 전에 하므로 체크하지 않는다.
 * @param {Number} quantity, 수량
 * @param {Number} price, 단가
 * @return {Number} 계산 결과
 */
function calculateAmount(quantity, price){
  // 주석을 자세하게 작성하기 위해
  // 함수 코드를 분리하는 경우도 있습니다.
};
```