

함수

왜 함수를 사용할까

- 프로그래밍에서 가장 중요한 것은 문제를 분석하는 것
- 주어진 문제를 여러 개의 작은 문제로 나눈 후 작은 문제를 하나씩 해결하면서 최종적으로 주어진 문제를 끝낸다.
- 가장 작은 단위로 나눈 것을 함수로 작성한다.
- 코드 재사용
- 코드 줄이기

함수 선언 & 실행

- 함수를 선언할 때에는 function이라는 예약어를 사용하고
- 함수 이름을 적은 후 중괄호 안에 실행할 여러 명령들을 묶는다.
- 함수를 실행(호출)할 때는 함수 이름 뒤에 중괄호 ()를 붙인다.

함수 선언

```
function 함수명() {  
    명령(들)  
}
```

함수 실행

```
함수명()
```

선언적함수

```
function 함수() {  
}
```

함수 선언

```
function 함수명() {  
    실행문;  
    실행문;  
    실행문;  
    실행문;  
    실행문;  
}
```

함수 실행

→ 함수명();

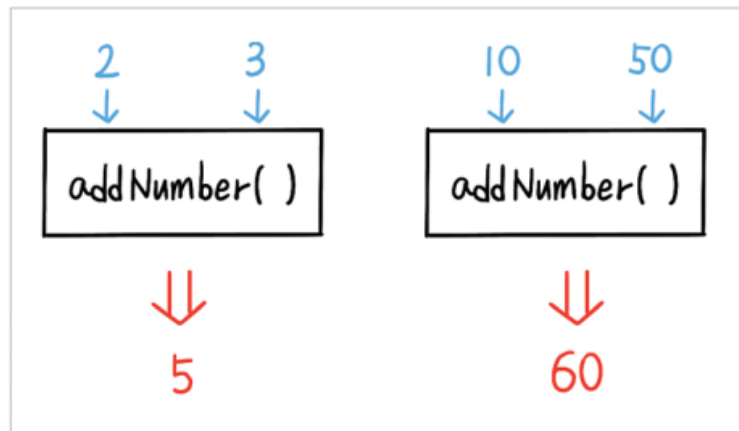
함수 선언 소스는 어디에 넣어도 상관없다.
하지만, 일반적으로 함수를 선언하는 소스를
실행 소스보다 앞에 넣는다.

매개변수와 인수

함수를 여러 번 사용할 수 있게 해보자

```
<script>  
  function addNumber(매개변수, 매개변수) {  
    실행문;  
    실행문;  
  }  
  addNumber(1, 5);  
</script>
```

← 항상 같은 값이 출력됨



매개변수와 인수

매개변수와 인수를 통틀어서 '인자'라고도 함

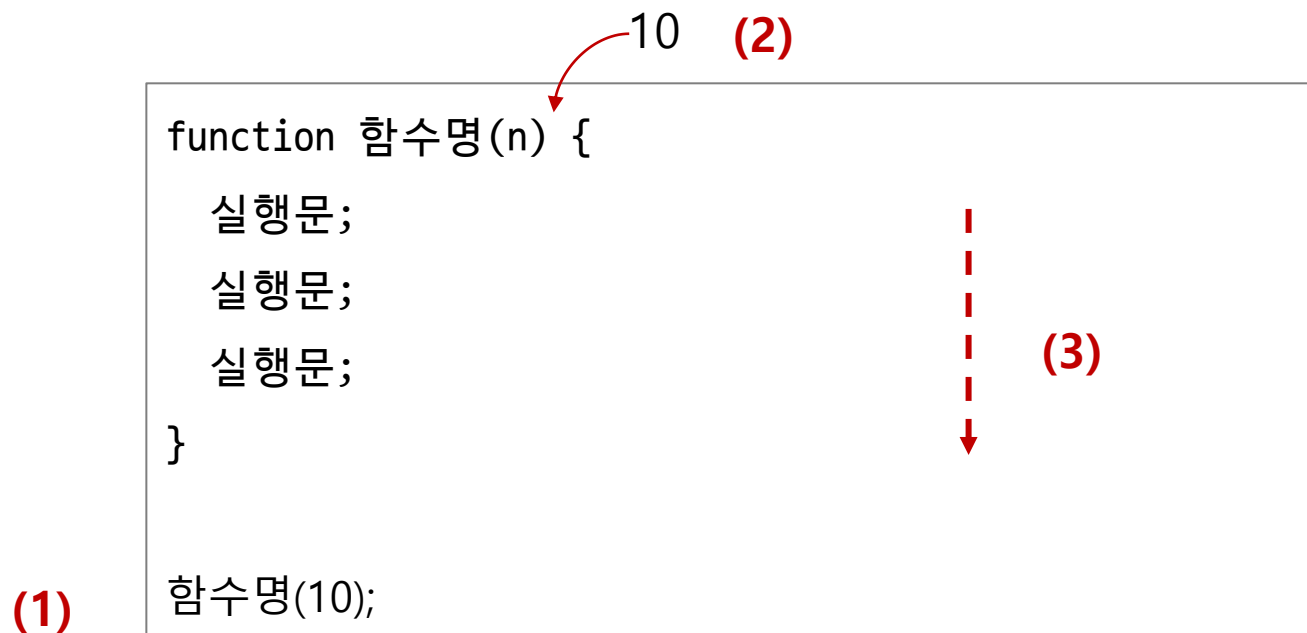
매개변수

- 함수를 선언할 때 외부에서 값을 받는 변수
- 함수 이름 옆의 괄호 안에 매개변수 이름을 넣어준다
- 매개변수에 이름을 붙이는 방법은 일반적인 변수 이름을 붙이는 방법과 같다.
- 매개변수는 선언된 함수에서만 사용한다.
- 함수에 여러 개의 매개변수가 필요할 때에는 매개변수 사이에 쉼표(,)를 찍으면서 나열한다.

인수

매개변수가 있는 함수를 실행할 때, 매개변수로 값을 넘겨주는 변수

자바스크립트에서 함수 처리 순서



자바스크립트는 function이라는 예약어를 만나면 함수를 선언한다는 것을 알고 일단 메모리 어딘가에 함수명() 함수의 내용을 저장해둔다.

- 1) 함수 선언 다음에 오는 함수명(10) 명령을 만나면 함수명() 함수를 실행한다.
- 2) 이 때 소괄호 안에 있는 값은 함수를 선언할 때 지정한 매개변수 n으로 넘겨진다.
- 3) 그 값을 사용해서 함수에 있던 명령을 차례대로 실행한다.

return문

- 함수 안에서 실행하고 그 결과를 함수 밖에서 받아 처리해야 할 경우도 많다.
- 함수의 실행 결과를 함수를 실행한 시점으로 넘겨주는 것을 '결과값을 반환한다^{return}'라고 한다.
- 함수를 실행한 후 결과를 반환할 때는 예약어 return 다음에 넘겨줄 값이나 변수를 지정한다.

```
function 함수명(n) { // n: 매개변수
    실행문;
    실행문;
}

let 변수 = 함수명(값)
document.write(변수);
```

기본 매개변수

ES6에는 기본 매개변수가 있어서 함수를 정의할 때 매개변수의 기본값을 지정할 수 있다.

→ 함수를 실행할 때 인수가 부족하면 기본값을 사용한다.

```
function multiple(a, b = 5, c = 10 ) {  
    return a * b + c;  
}  
  
multiple(5, 10, 20)  
multiple(10, 20)  
multiple(10);
```


스코프

스코프(scope) : 선언한 변수의 적용 범위를 가리킨다.

- var 키워드를 사용한 변수는 함수 레벨 스코프
- let, const 키워드를 사용한 변수는 블록 레벨 스코프

익명함수

```
let 변수 = function() {  
    ...  
}
```

```
let 변수 = function(매개변수) {  
    ...  
}
```

즉시 실행 함수

지금까지는 함수를 선언한 후 실행 → 같은 함수를 여러번 실행할 수 있음
한번만 사용하는 함수라면 → 선언과 동시에 실행할 수 있다.

```
(function() {  
    ...  
})();
```

```
(function(매개변수) {  
    .....  
})(인수);
```

화살표 함수

화살표 함수(애로우 펑션) : ES6 버전부터는 =>을 사용해 함수를 좀 더 간단하게 선언함.

화살표 함수는 함수 표현식을 사용할 경우에만 사용할 수 있음.

```
( ) => {함수 내용}
```

```
(매개변수) => {함수 내용}
```

콜백 함수

콜백 함수는 다른 함수의 인자로 사용하는 함수