

화살표 함수를 사용하면 더 짧은 함수 구문을 작성할 수 있습니다.

1. 화살표 함수

구문

```
const 변수명 = (매개변수, ...) => {  
  함수 내용  
}
```

1.1 기본형

변경전

```
const add1 = function(a, b) {  
  let sum = a + b;  
  return sum;  
};
```

변경후

```
const add2 = (a, b) => {  
  let sum = a + b;  
  return sum;  
};
```

1.2 함수 실행문이 1개만 있는 경우 : {} 생략 가능

변경전

```
const add1 = function(a, b) {  
  console.log(a + b);  
}
```

변경후

```
const add2 = (a, b) => console.log(a + b);
```

1.3 함수 실행문이 1개이면서 return문인 경우 : {} 및 return 생략

변경전

```
const add1 = function(a, b) {  
  return a + b;  
}
```

변경후

```
const add2 = (a, b) => a + b;
```

1.4 매개 변수가 1개인 경우 : () 생략 가능

변경전

```
const sqrt1 = function(a) {
  let result = a * a;
  return result;
}
```

변경후

```
const sqrt2 = a => {
  let result = a * a;
  return result;
}
```

1.5 매개 변수가 1개가 아닌(0개 혹은 2개 이상)인 경우 : () 생략 불가

변경전

```
const sqrt1 = function() {
  console.log("Hello React!");
}
```

변경후

```
const sqrt2 = () => console.log("Hello React!");
```

2. Arrow 함수에서의 this

일반 함수와 비교하여 Arrow 함수의 this 처리가 다릅니다.

일반 함수에서 this 키워드는 함수를 호출한 객체를 나타내며 window, document, button 등이 될 수 있습니다.

Arrow 함수에서 this 키워드는 항상 Arrow 함수를 정의한 객체를 나타냅니다.

차이점을 이해하기 위해 두 가지 예를 살펴보겠습니다.

두 예제 모두 페이지가 로드될 때 첫 번째로, 사용자가 버튼을 클릭할 때 다시 한번 메시지를 두 번 호출합니다.

첫 번째 예제는 일반 함수를 사용하고 두 번째 예제는 Arrow 함수를 사용합니다.

결과는 window 개체가 함수의 "소유자"이기 때문에 첫 번째 예제는 두 개의 다른 개체(window와 button)를 반환하고 두 번째 예제는 window 개체를 두 번 반환한다는 것을 보여줍니다.

일반 함수에서 this는 함수를 호출하는 객체를 나타냅니다:

```
// Regular Function:
hello = function() {
  document.getElementById("demo").innerHTML += this;
}

// The window object calls the function:
window.addEventListener("load", hello);

// A button object calls the function:
document.getElementById("btn").addEventListener("click", hello);
```

Arrow 함수의 this는 함수의 소유자를 나타냅니다:

```
// Arrow Function:
hello = () => {
  document.getElementById("demo").innerHTML += this;
}

// The window object calls the function:
window.addEventListener("load", hello);

// A button object calls the function:
document.getElementById("btn").addEventListener("click", hello);
```

함수로 작업 할 때 이러한 차이점을 기억하십시오. 때로는 일반 함수의 동작이 원하는데로 작동하지만 그렇지 않은 경우 **Arrow** 함수를 사용합니다.