

## 동기식

위에서 아래로 순서대로 실행

```
console.log("1. 시작");

for(let i=0; i<3; i++){
  console.log("반복:", i);
}

console.log("2. 끝");
```

## 비동기식

```
console.log("1. 시작");

setTimeout(() => {
  console.log("비동기 실행 (2초 후)");
}, 2000);

console.log("2. 끝");
```

## 비동기식의 문제점

```
function getNumberAsync(){
  setTimeout(() => {
    const num = 10;
    return num;    // 이 return은 getNumberAsync()에 전달되지 않음
  }, 2000);
}

console.log("시작");

const result = getNumberAsync(); // undefined
console.log("받은 숫자:", result); //   아직 값이 없음 → undefined

console.log("끝");
```

## 콜백함수를 이용한 비동기식 문제점 해결

```
function getNumberAsync(callback){
  setTimeout(() => {
    const num = 10;
    callback(num);    // 결과가 준비되었을 때 콜백 실행
  }, 2000); // 2초 걸리는 작업 가정
}

console.log("시작");

getNumberAsync((result) => {
  console.log("받은 숫자:", result); // 이 시점에서만 결과 사용 가능
});

console.log("끝");
```

## 1. 콜백 함수란?

콜백 함수(Callback Function)는 다른 함수의 인자로 전달되어 실행되는 함수 즉, "함수를 함수에 넘겨서, 나중에 그 함수 안에서 호출되는 것"이 콜백 함수

## 2. 기본 구조

```
function main(callback) {  
  console.log("main 함수 실행 시작");  
  callback(); // 전달받은 함수 실행  
  console.log("main 함수 실행 끝");  
}
```

```
function call(){  
  console.log( "전달 함수")  
}
```

```
main(call);
```

### 예제 1: 기본적인 콜백

```
function greet() {  
  console.log("안녕하세요!");  
}  
  
function runWithCallback(callback) {  
  console.log("콜백을 실행합니다");  
  callback(); // 넘겨받은 함수 실행  
}  
  
runWithCallback(greet);
```

### 예제 2: 익명 함수로 전달

```
function runTask(callback) {  
  console.log("작업 시작");  
  callback();  
  console.log("작업 끝");  
}  
  
runTask(function() {  
  console.log("콜백 함수 실행 중!");  
});
```

### 예제 3: 콜백에 매개변수 전달하기

```
function calculate(num, callback) {  
  console.log("계산 시작");  
  callback(num);  
}  
  
calculate(5, function(n) {  
  console.log(n + 10); // 15
```

```
});
```

#### 예제 4: 문자열 가공

```
function processText(text, callback) {  
  console.log("문자열 처리:");  
  callback(text);  
}  
processText("Hello", function(msg) {  
  console.log(msg + " World!");  
});
```

#### 예제 5: 반복 실행

```
function repeatTask(callback) {  
  for (let i = 1; i <= 3; i++) {  
    callback(i);  
  }  
}  
  
repeatTask(function(n) {  
  console.log("콜백 실행: " + n);  
});
```

#### 예제 6: 조건에 따라 실행

```
function checkNumber(num, callback) {  
  if (num % 2 === 0) {  
    callback("짝수입니다");  
  } else {  
    callback("홀수입니다");  
  }  
}  
  
checkNumber(7, function(result) {  
  console.log(result); // 홀수입니다  
});
```

### 3. 콜백함수 개념 정리

- 콜백은 다른 함수에 전달되는 함수이다.
- 함수 호출 시점과 방식은 호출을 받는 함수가 결정한다.
- 매개변수를 넘겨주거나 조건에 따라 여러 번 실행 가능하다.