



Chap47 에러처리

▼ try...catch..finally문

에러 러치 코드를 미리 등록해 두고 에러가 발생하면 에러 처리 코드로 점프하도록 하는 방법

```
try{
  //실행할 코드(에러가 발생할 가능성이 있는 코드)
}catch(err){
  //try코드 블록에서 에러가 발생하면 이 코드 블록의 코드가 실행된다.
  //err에는 try 코드 블록에서 발생한 Error객체가 전달된다.
}finally{
  //에러 발생과 상관없이 반드시 한 번 실행된다.
}
```

try...catch...finally문으로 에러를 처리하면 프로그램이 강제 종료되지 않는다.

▼ Error 객체

Error 생성자 함수는 에러 객체를 생성한다. → 모두 Error.prototype을 상속받음
자바스크립트에서의 에러 생성자함수를 포함해 7가지 에러 객체 제공

- Error : 일반적인 에러 객체
- SyntaxError : 자바스크립트 문법에 맞지 않는 문을 해석할 때 발생하는 에러 객체
- ReferenceError : 참조할 수 없는 식별자를 참조했을 때 발생하는 에러 객체
- TypeError : 피연산자 또는 인수의 데이터 타입이 유효하지 않을 때 발생하는 에러 객체
- RangeError : 숫자값의 허용 범위를 벗어났을 때 발생하는 에러 객체
- URIError : encodeURIComponent 또는 decodeURI 함수에 부적절한 인수를 전달했을 때 발생하는 에러 객체
- EvalError : eval 함수에서 발생하는 에러 객체

▼ throw 문

에러객체를 생성한다고 에러가 발생하는 것은 아니다. 에러를 발생시키려면 try코드 블록에서 throw문으로 에러 객체를 던져야 한다.

```
try{
  //에러 객체를 던지면 catch 코드 블록이 실행되기 시작한다.
  throw new Error('something wrong');
}catch(error){
  console.log(error);
}
```

▼ 에러의 전파

에러는 호출자 방향으로 전파된다. 즉 콜 스택의 아래 방향(실행 중인 실행 컨텍스트가 푸시되기 직전에 푸시된 실행 컨텍스트 방향)으로 전파된다.

이처럼 throw된 에러를 캐치하지 않으면 호출자 방향으로 전파된다. 이때 throw된 에러를 캐치하여 적절히 대응하면 프로그램을 강제 종료시키지 않고도 코드의 실행 흐름을 복구할 수 있다.

주의할 것은 비동기 함수인 setTimeout이나 프로미스 후속 처리 메서드의 콜백 함수는 호출자가 없다는 것이다. 따라서 에러를 전파할 호출자가 존재하지 않는다.