# 초보자도 할 수 있는
# C# Programming


Fast Campus
Online

한창민 강사.

초보
탈출 I

Fast campus

# 9강
# 예외 처리

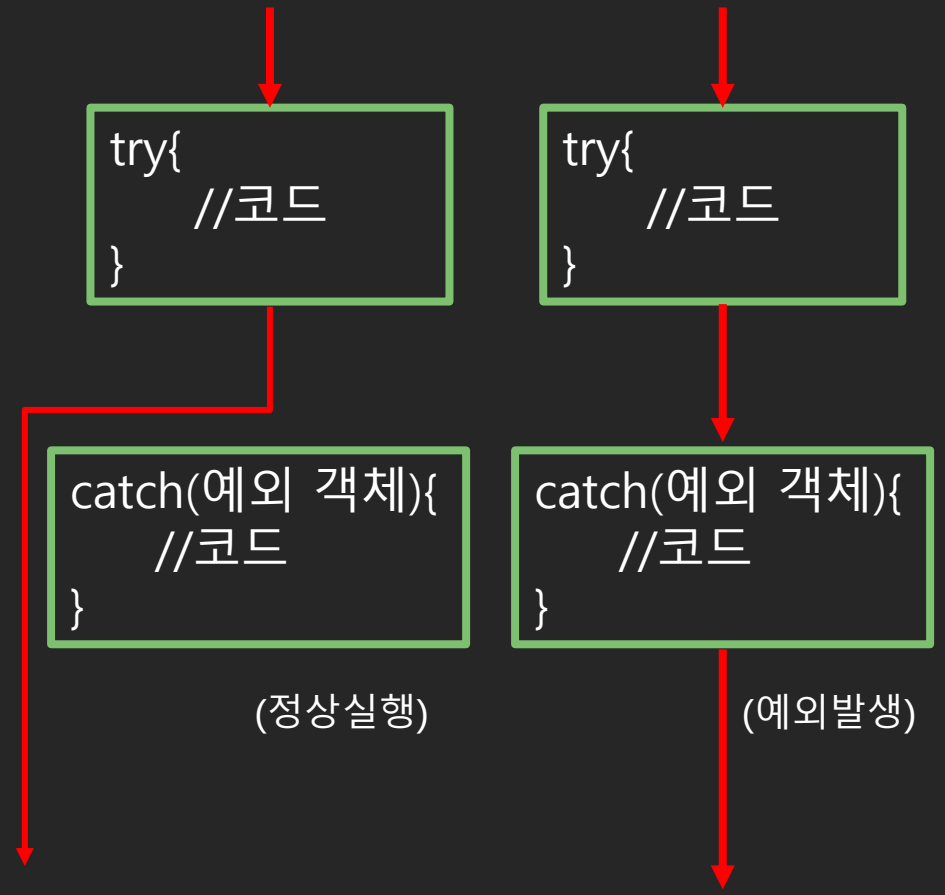
Fast Campus
Online

한창민 강사.

Fast campus

- try ~ catch 키워드
  - 프로그램의 안정성

```
try {
    inputNum = int.Parse(readStr);
    isCorrect = true;
}
catch(FormatException e) {
    Console.WriteLine(e.Message);
}
```




➡ 소스코드 (_111_try_catch)

Fast campus

- System.Exception
  - 예외 클래스의 Base

```
try {
    checked {
        maxNum += Int32.MaxValue;
    }
}
catch(OverflowException e) {
    Console.WriteLine("e.Message: " + e.Message);
    Console.WriteLine("e.Source:" + e.Source);
}
catch(Exception e) {
    Console.WriteLine(e.Message);
}
```



➡ 소스코드 (_112_Exception)

Fast campus

- throw 키워드
  - 상위 호출 메소드로 예외 객체를 전달

```
static void ThrowFunc(int data) {
    if(data > 0) {
        Console.WriteLine("ThrowFunc data: " + data);
    }
    else {
        throw new Exception("data에 0이 입력되었습니다.");
    }
}
```

  - 조건 연산자에서 사용

```
int result = 101;
try {
    int checkNum = (result < 100) ? result : throw new Exception("100 이하만 가능");
}catch(Exception e) {
    Console.WriteLine(e.Message);
}
```

➡ 소스코드 (_113_throw)

Fast campus

- try ~ catch ~ finally 키워드
  - 예외 상황과 관련 없이 무조건 처리

```
try {
    ThrowFunc(0);
}
catch(Exception e) {
    Console.WriteLine(e.Message);
}
finally {
    Console.WriteLine("  finally 무조건 실행  ");
}
```

➡ 소스코드 (_114_finally)

Fast campus

- 사용자 정의 예외 클래스
  - 기존 예외처리 클래스에서 상속

```
class MyException : ApplicationException
{
    public int Num { get; set; }
    public MyException():base() {
    }
    public MyException(int a) {
        Num = a;
    }

    public override string ToString() {
        return "Num: " + Num;
    }
}
```

  - when 키워드, StackTrace 키워드

```
Console.WriteLine("when(e.Num == 0)");
```

```
Console.WriteLine("MyException: " + e.StackTrace);
```

➡ 소스코드 (_115_CustomExcepion)