



주문식교육의 산실  
**영진전문대학교**

# 구조적 프로그래밍 구성 요소 (디버깅)

일본IT과

정영철 교수

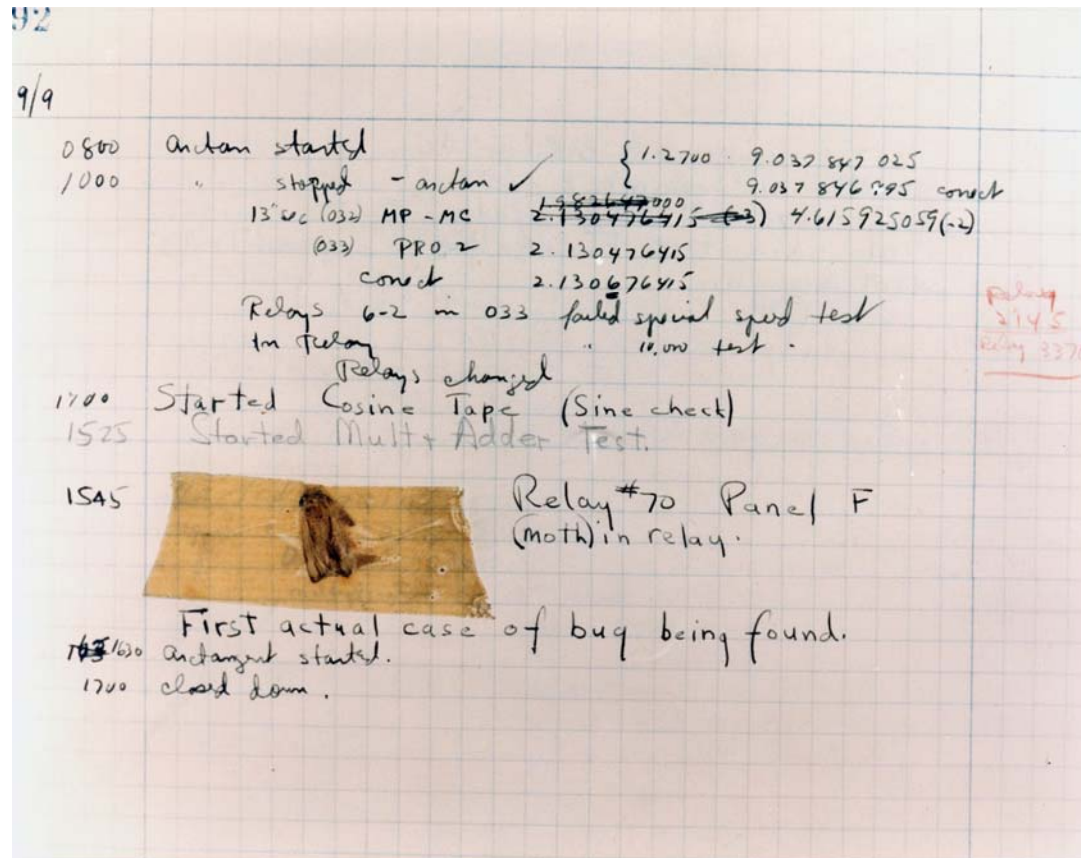


**영진전문대학교 컴퓨터정보계열**

SCHOOL OF COMPUTER INFORMATION

# 디버깅(Debugging)란?

- 프로그램 내 문제점(Bug)을 찾아 해결 하는 것
  - S/W에 내재된 문제점을 Bug라 지칭 하는 이유?
    - Harvard Mark II 컴퓨터 전자 장치에 실제 곤충이 회로에 붙어 문제를 일으켰음.
    - 이때부터 S/W 내 문제점을 Bug라 지칭



# Source Code 디버깅(Debugging) 방법 : 논리적 에러

- 문제점(Bug) 발생 **영역(Area) 예측**
  - 소스코드의 양이 많을 경우 Error를 찾기 위해 모든 소스를 살펴 볼 수 없다.
  - 이에 Error가 예상되는 코드를 부분별로 나누어 판단
    - 예) 코드가 입력 + 처리 + 출력으로 구성되어 있고, 현재 출력 부분에 오류가 의심 된다면, 출력 부분만 오류 검사 실시
- 문제점 발생 예측 영역 **정밀 검사**
  - 코드를 Line-By-Line 실행하여 코드의 논리적 오류 검사 실시
- 정밀검사 시 다양한 **디버깅 소프트웨어 활용**
  - Python Debug S/W + VS Code 활용 디버깅 실습
- Debug S/W 동작 절차
  1. 프로그램의 중단점을 프로그래머가 설정
  2. 디버그 S/W 실행
  3. 디버그 S/W가 설정된 중단점에서 프로그램의 실행을 중단
  4. 실행 중단된 지점부터 Line-By-Line 실행
  5. Line-By-Line 코드 실행으로 코드의 진행 흐름과 변수 값의 변화 확인
  - 위 작업을 통해 프로그래머가 범한 논리적 오류를 확인 ← 디버그 S/W 사용 이유

## 디버깅 방법 (1) : 디버깅 테스트 예제 파일 작성

```
1
2 print("-----")
3
4 for value in range(1, 11):
5     print ("Value : ", value)
6
7     if value % 2 == 0:
8         print("2의 배수 입니다.")
9
10
11 print("-----")
```

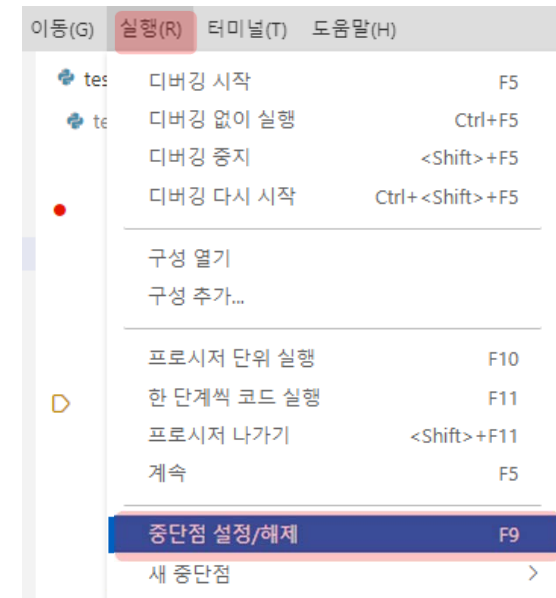
## 디버깅 방법 (2) : 프로그램 중단점 지정

- 가정 : 본 예제에서는 2번 라인에서 프로그램 실행 중단점 설정
- VS Code에서 디버깅을 위한 중단점 삽입 방법
  - 중단 하고자 하는 코드 라인에 마우스 위치 후
    - 메뉴 -> 실행 -> 중단점 설정
    - 또는 F9 단축키 또는 해당 라인 넘버 왼쪽 여백 더블 클릭
  - 경우에 따라 여러 줄에 중단점 삽입 가능

1 2번 라인 마우스 클릭

```
1  
2 print("-----")  
3  
4 for value in range(1, 11):  
5  
6     print ("Value : ", value)  
7  
8     if value % 2 == 0:  
9         print("2의 배수 입니다.")  
10  
11 print("-----")
```

- 3 2번 라인 중단점 삽입 확인
- 중단점이 성공적으로 삽입 되었을 경우 해당 라인 좌측편에 빨간색 점이 표시됨

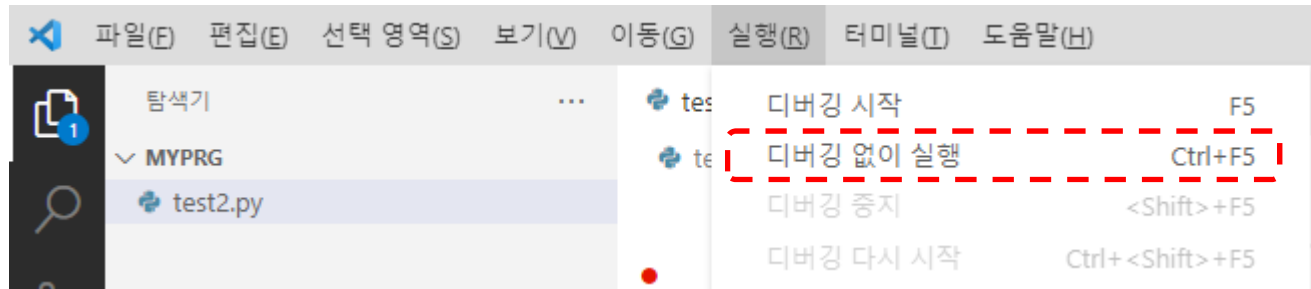


- 2 메뉴->실행->중단점 설정 또는 F9 단축키 실행

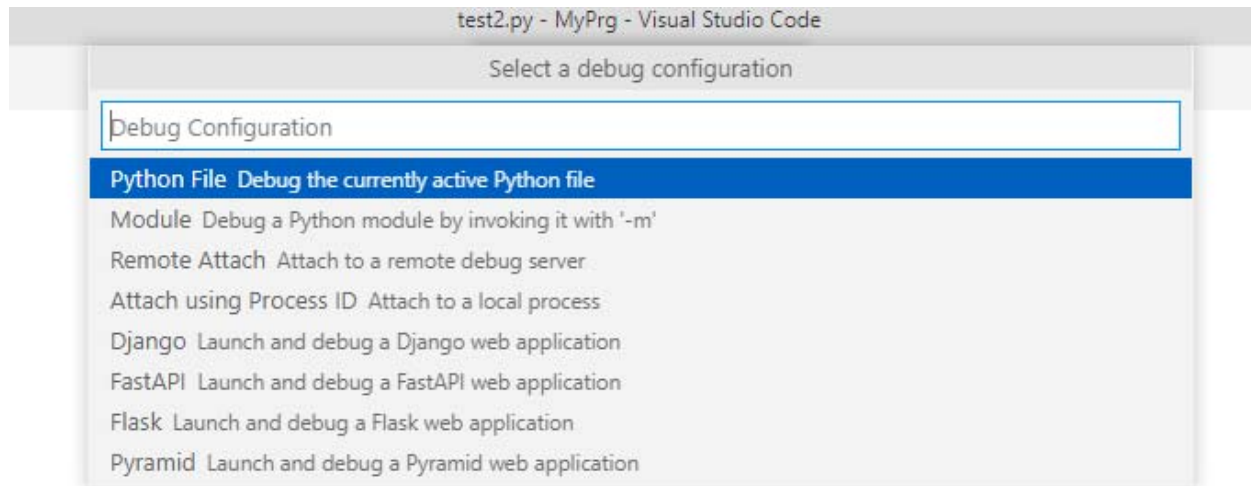
## 디버깅 방법 (3) : 디버그 실행

- 디버그 실행

- **F5 단축키**를 이용하거나, "**메뉴 -> 실행 -> 디버깅 시작**" 선택










- VS Code 에서 디버그 처음 실행 시 아래 설정 모드 선택 창 출력
- 본 예제에서는 "Python File Debug the currently active Python File" 선택



## 디버깅 방법 (4) : Line-By-Line 코드 실행

- 디버그를 실행하면 프로그램이 시작되며,
- 설정된 중단점에서 프로그램이 멈추게 된다.
- 프로그램이 중단점에서 멈춘 후 **디버그 실행 메뉴**를 통해 Line-By-Line 코드 실행



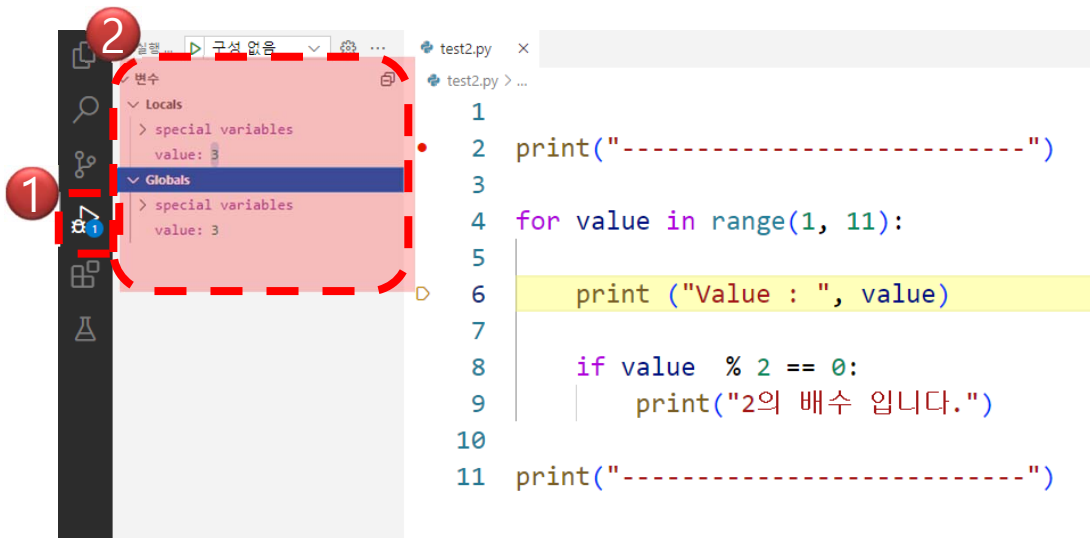
	F5 : 디버깅 실행
	F10 : Line-By-Line 실행 (함수 건너 뛴)
	F11 : Line-By-Line 실행 (함수 내부 탐색)
	Shift+ F11 : 함수 탐색 중 해당 함수 탈출
	Ctrl+Shift+F5 : 디버깅 재실행
	Ctrl+F5 : 디버깅 실행

실행(R)	터미널(T)	도움말(H)
디버깅 시작 F5		
디버깅 없이 실행 Ctrl+F5		
디버깅 중지 <Shift>+F5		
디버깅 다시 시작 Ctrl+<Shift>+F5		
구성 열기		
구성 추가...		
프로시저 단위 실행 F10		
한 단계씩 코드 실행 F11		
프로시저 나가기 <Shift>+F11		
계속 F5		

## 디버깅 방법 (5) : Line-By-Line 코드 실행 중 변수 값 확인

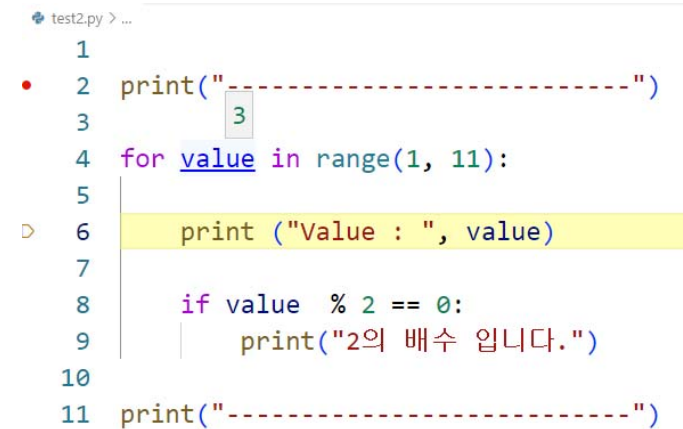
- **F11** 또는 **F10** 단축키를 활용해 코드를 Line-By-Line으로 실행 가능
- 이때 코드의 실행 흐름과 변수들의 값 변화 상태 확인 가능

✓ 네비게이션 창 활용



✓ 마우스 활용

- 해당 변수에 마우스 커서 오버를 시 해당 변수 값 출력



- **F10과 F11 단축키 차이점은 수업 시간 중 실습을 통해 설명**



# Q/A

## 감사합니다



주문식교육의 산실  
**영진전문대학교**