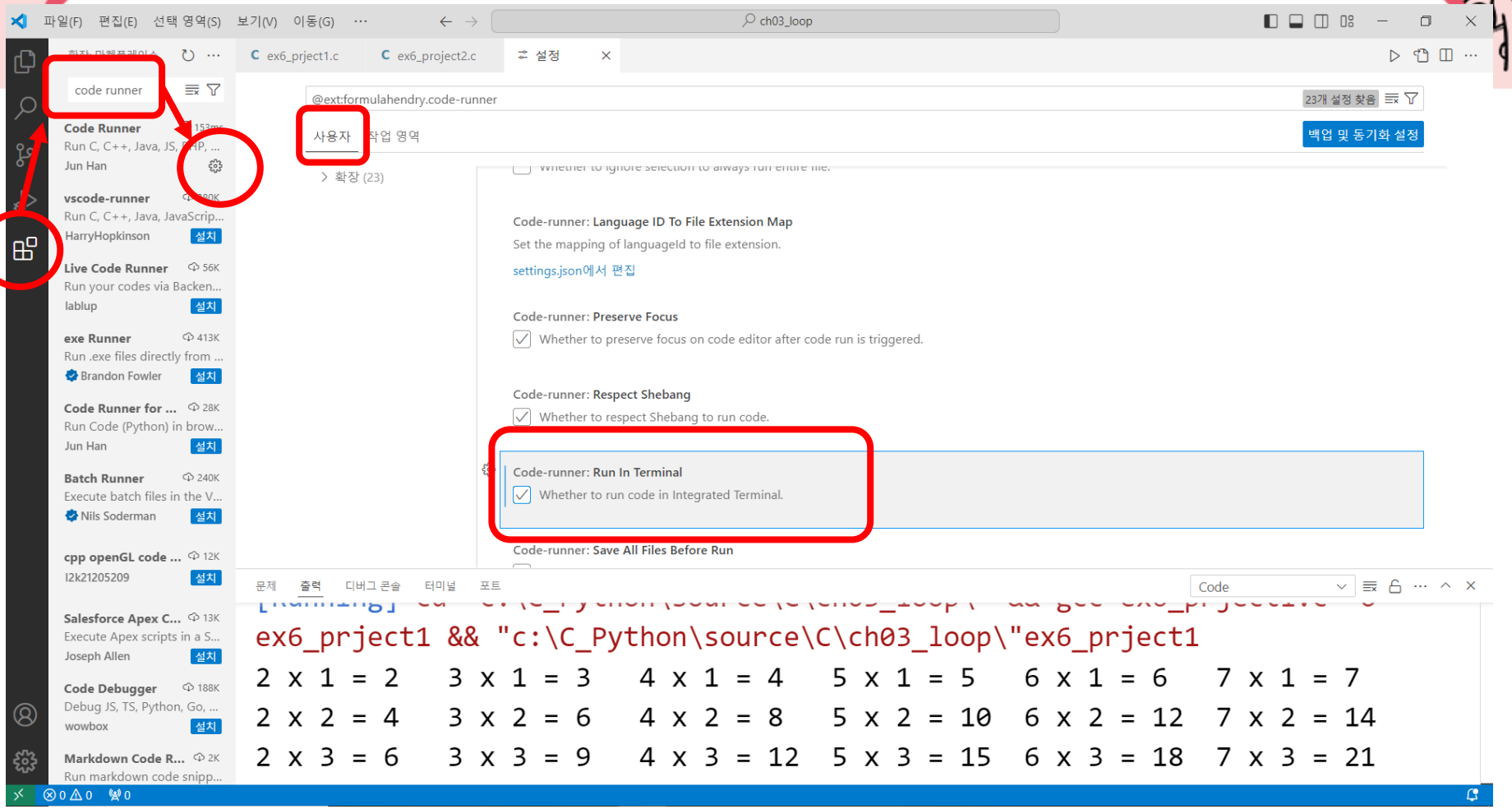
# 4장 조건에 따라 다른 일하기
## : 조건문

1. 이 장에서 만드는 프로그램
2. 조건에 따라 분기하기
3. 실행 중단하기
4. 여러 방향으로 분기하기
5. 프로젝트: 숫자 맞히기


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) ···
ch03_loop
code runner
Code Runner
Run C, C++, Java, JS, ...
Jun Han
vscode-runner
Run C, C++, Java, JavaScrip...
HarryHopkinson
설치
Live Code Runner
56K
Run your codes via Backen...
lablup
exe Runner
413K
Run .exe files directly from ...
Brandon Fowler
Code Runner for ...
28K
Run Code (Python) in brow...
Jun Han
Batch Runner
240K
Execute batch files in the V...
Nils Soderman
cpp openGL code ...
12K
I2k21205209
Salesforce Apex C...
13K
Execute Apex scripts in a S...
Joseph Allen
Code Debugger
188K
Debug JS, TS, Python, Go, ...
wowbox
Markdown Code R...
2K
Run markdown code snipp...
ex6_prject1.c
ex6_project2.c
설정
@ext:formulahendry.code-runner
23개 설정 찾음
사용자
작업 영역
백업 및 동기화 설정
확장 (23)
Code-runner: Language ID To File Extension Map
Set the mapping of languageId to file extension.
settings.json에서 편집
Code-runner: Preserve Focus
Whether to preserve focus on code editor after code run is triggered.
Code-runner: Respect Shebang
Whether to respect Shebang to run code.
Code-runner: Run In Terminal
Whether to run code in Integrated Terminal.
Code-runner: Save All Files Before Run
문제 출력 디버그 콘솔 터미널 포트
Code
ex6_prject1 && "c:\C_Python\source\C\ch03_loop\"ex6_prject1
2 x 1 = 2 3 x 1 = 3 4 x 1 = 4 5 x 1 = 5 6 x 1 = 6 7 x 1 = 7
2 x 2 = 4 3 x 2 = 6 4 x 2 = 8 5 x 2 = 10 6 x 2 = 12 7 x 2 = 14
2 x 3 = 6 3 x 3 = 9 4 x 3 = 12 5 x 3 = 15 6 x 3 = 18 7 x 3 = 21

사용자 작업 영역

> 확장 (23)

Code-runner: **Language ID To File Extension Map**

Set the mapping of languageId to file extension.

settings.json에서 편집

Code-runner: **Preserve Focus**

☑ Whether to preserve focus on code editor after code run is triggered.

Code-runner: **Respect Shebang**

☑ Whether to respect Shebang to run code.

Code-runner: **Run In Terminal** *(다음에서도 수정됨 사용자)*

☑ Whether to run code in Integrated Terminal.

Code-runner: **Save All Files Before Run**

# 1. 이 장에서 만드는 프로그램

그림 4-1 숫자 맞히기 게임 구성 1

# 1. 이 장에서 만드는 프로그램

그림 4-2 숫자 맞히기 게임 구성 2

# 2. 조건에 따라 분기하기

## (1) if-else 문

- 조건문 : 프로그램을 실행하는 중에 조건에 따른 분기가 필요할 때 사용

형식

```
if (조건) {
    // 조건을 만족할 때 수행할 문장
} else {
    // 조건을 만족하지 않을 때 수행할 문장
}
```

# 2. 조건에 따라 분기하기

## (2) if-else if-else 문

- ⊖ if 문에서 조건을 만족하면 문장을 수행하고 조건문을 끝낸다.
- ⊖ if 문에서 조건을 만족하지 않으면 다음에 있는 else if 문에서도 조건을 확인하고 만족하면 수행하고 조건문을 끝낸다.
- ⊛ 모두 만족하지 않으면 마지막에 else 문으로 가서 해당 문장을 수행하고 끝낸다.

형식

```
if (조건) {
    // 수행할 문장
} else if (조건) {
    // 수행할 문장
} else if (조건) {
...
} else {
    // 수행할 문장
}
```

# 2. 조건에 따라 분기하기

## (2) if-else if-else 문

⊖ 점수에 따른 학점 출력하기

형식

```
if (조건) {
    // 수행할 문장
} else if (조건) {
    // 수행할 문장
} else if (조건) {
...
} else {
    // 수행할 문장
}
```

# 2. 조건에 따라 분기하기

## (3) switch 문

- 조건으로 어떤 값을 받고 이 값과 일치하는 case 문의 문장 수행

형식

```
switch (조건) {
    case 값1:
        // 수행할 문장
        break;
    case 값2:
        // 수행할 문장
        break;
    ...
    default:
        // 어떤 값도 해당하지 않을 때 수행할 문장
}
```

# 2. 조건에 따라 분기하기

(4) 실습: 점수에 따른 학점 출력하기(if, switch)

# 3. 여러 방향으로 분기하기

## (1) 난수 생성하기

- rand() 함수 : 난수를 생성하는 함수

형식 `rand() % 어떤 수;`

- 전처리기 지시문에 time.h, stdlib.h 파일 추가
- '어떤 수'는 내가 뽑고 싶은 숫자의 범위를 지정
- 난수 초기화

# 3. 여러 방향으로 분기하기

## (1) 난수 생성하기

4.4.1 난수.c

```c
int main(void) {
    printf("난수 초기화 이전...\n");
    for (int i = 0; i < 10; i++) {
        printf("%d ", rand() % 10);
    }
    srand(time(NULL)); // 난수 초기화
    printf("\n\n난수 초기화 이후...\n");
    for (int i = 0; i < 10; i++) {
        printf("%d ", rand() % 10);
    }
    return 0;
}
```

실행결과

```
난수 초기화 이전...
1 7 4 0 9 4 8 8 2 4

난수 초기화 이후...
0 1 2 3 4 9 7 6 2 7
```

# 4. 프로젝트: 숫자 맞히기

㉠ 헤더 파일 추가, 난수 초기화

㉡ 난수 생성하기(1~100 숫자 반환) : rand() 함수

㉢ 실습이므로 정답 출력

④ 답변 기회를 5번으로 설정 : 변수 chance를 선언하고 5를 저장

# 4. 프로젝트: 숫자 맞히기

⑤ 답변 기회 차감 : chance에서 1씩 차감

⑥ 숫자 입력받기
- 안내문구를 printf()로 출력
- scanf_s()로 값을 입력받기

⑧ 정답을 맞힐 때까지 반복 : while 문

⑨ 정답과 입력한 숫자 비교
- 정답보다 입력한 숫자가 작을 때 : UP ↑
- 정답보다 입력한 숫자가 클 때 : DOWN ↓
- 예외 상황 : 알 수 없는 오류가 발생했어요.

⑩ 5회 이내에 정답을 맞히지 못했을 때
- 실패 안내 후 종료
- 실패 안내문구 : 모든 기회를 사용했어요. 아쉽게도 실패했습니다.