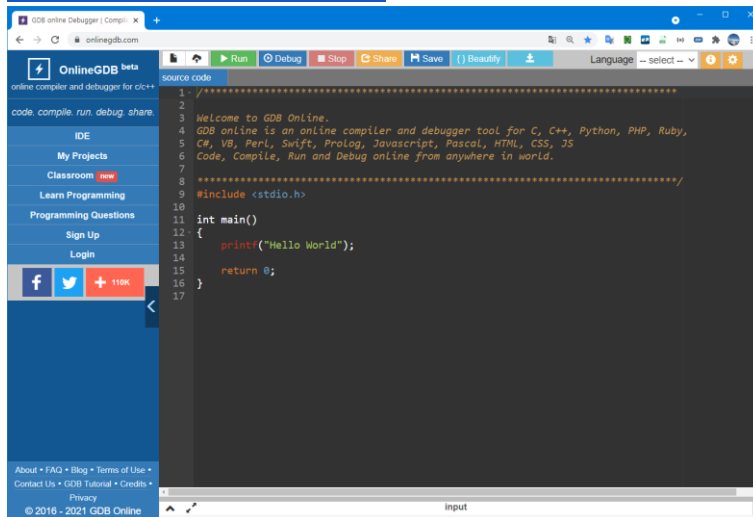


온라인 코딩 환경 소개

개요 인터넷 상에서 사용할 수 있는 다양한 파이썬 코딩 환경이 있습니다. 이중 대표적으로 이용하는 사이트 3가지를 소개 합니다.

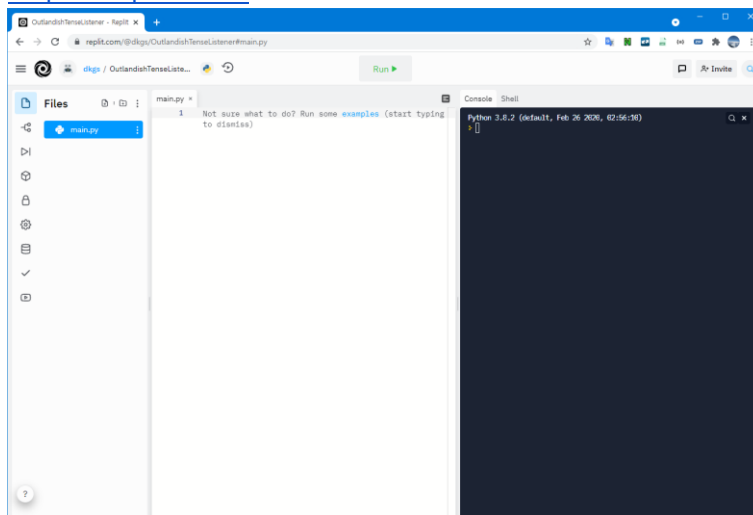
[1] <https://www.onlinegdb.com/>



다양한 언어를 지원하고 직관적이고 단순한 형태의 온라인 코딩 플랫폼 입니다. 로그인 없이 사용이 가능한 장점이 있지만, 패키지를 설치하여 사용할 수 없다는 단점이 있습니다.

사용방법 동영상: <https://youtu.be/6L8MVpAs6DI>

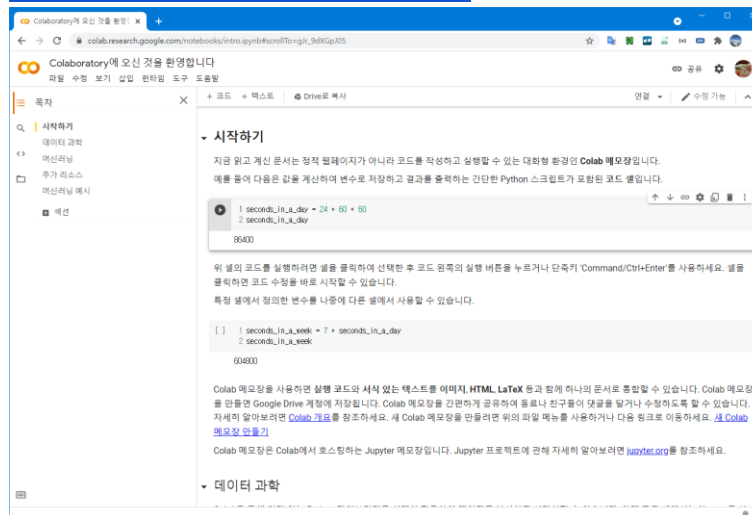
[2] <https://replit.com/>



설치형 코딩 플랫폼과 가장 유사한 형태의 환경을 제공하는 사이트 입니다. 다양한 언어를 제공하고 있으며, 유료 서비스의 경우 완성도 있는 개발 환경을 제공합니다. 가입을 해야만 사용할 수 있으며, 기본 환경에 대한 사용은 무료 입니다. onlinegdb 보다 조금 더 다양한 기능을 즐기는 분들에게 repl.it 을 추천 합니다.

사용방법 동영상: <https://youtu.be/c6XR3at3DSI>

[3] <https://colab.research.google.com/>



구글에서 머신러닝 학습을 위해 제공하는 파이썬 개발환경입니다. 가입을 해야만 사용할 수 있으며, 고성능의 서버 환경을 제공하기 때문에 머신러닝 개발 학습자 분들이 많이 사용하는 사이트 중 하나 입니다. 개발 플랫폼의 형태는 onlinegdb와 repl.it와 달리 주피터 노트북이라고 하는 대화형(하나의 창에 여러개의 코딩 창이 존재하는 형태)환경과 유사한 형태를 제공하고 있습니다.

사용방법 동영상: <https://youtu.be/c6XR3at3DSI?t=104>

설치형 코딩 환경 소개

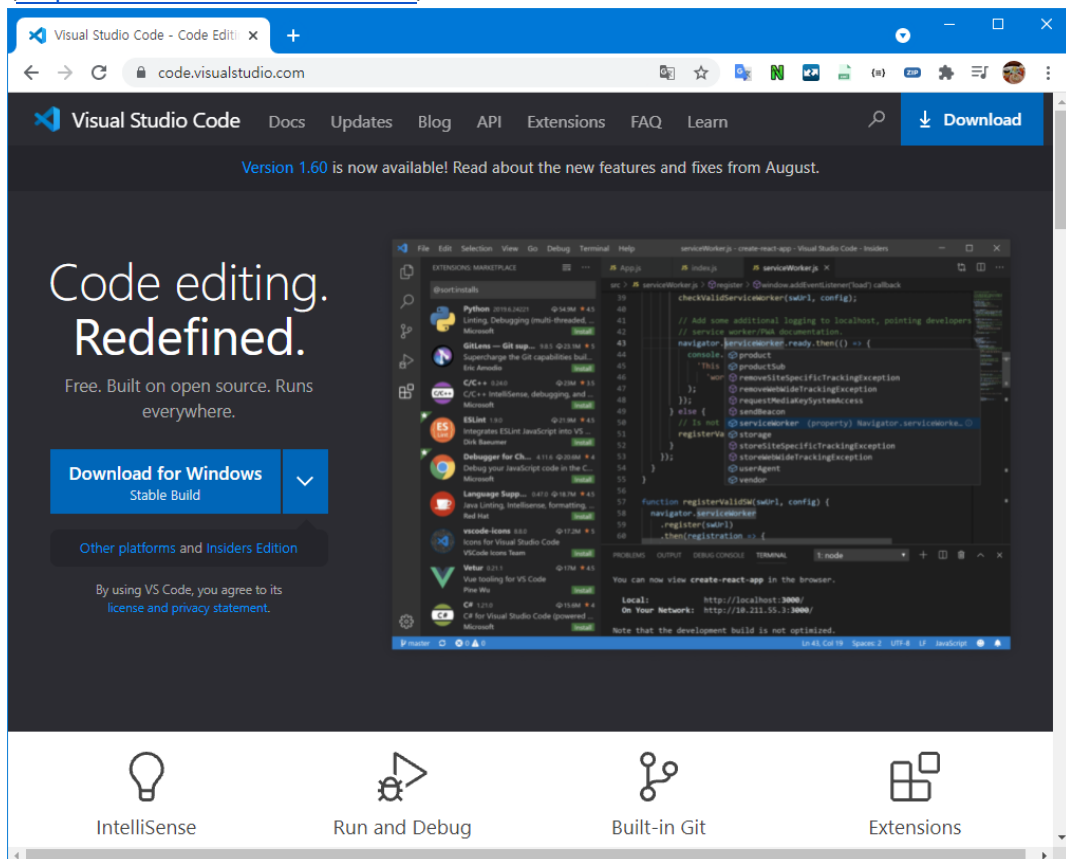
개요 20강의 판다스 실습과 21강의 tkinter 실습을 위해서는 개인 PC에 설치형 코딩환경을 구축해야 합니다. 개인 PC의 환경에 따라, 설치에 실패하는 경우가 종종 있습니다. 따라서 최대한 에러가 발생할 가능성이 적은 방법인 아래 방법으로 설치를 진행하겠습니다. 설치 방법은 윈도우, Mac, 리눅스 모두 동일합니다.

- VS Code 설치
- VS Code에서 Python Extension 설치

설치영상: <https://youtu.be/XQMVbi5ZxiY>

[1] VS Code 사이트 들어가기

(<https://code.visualstudio.com/>)



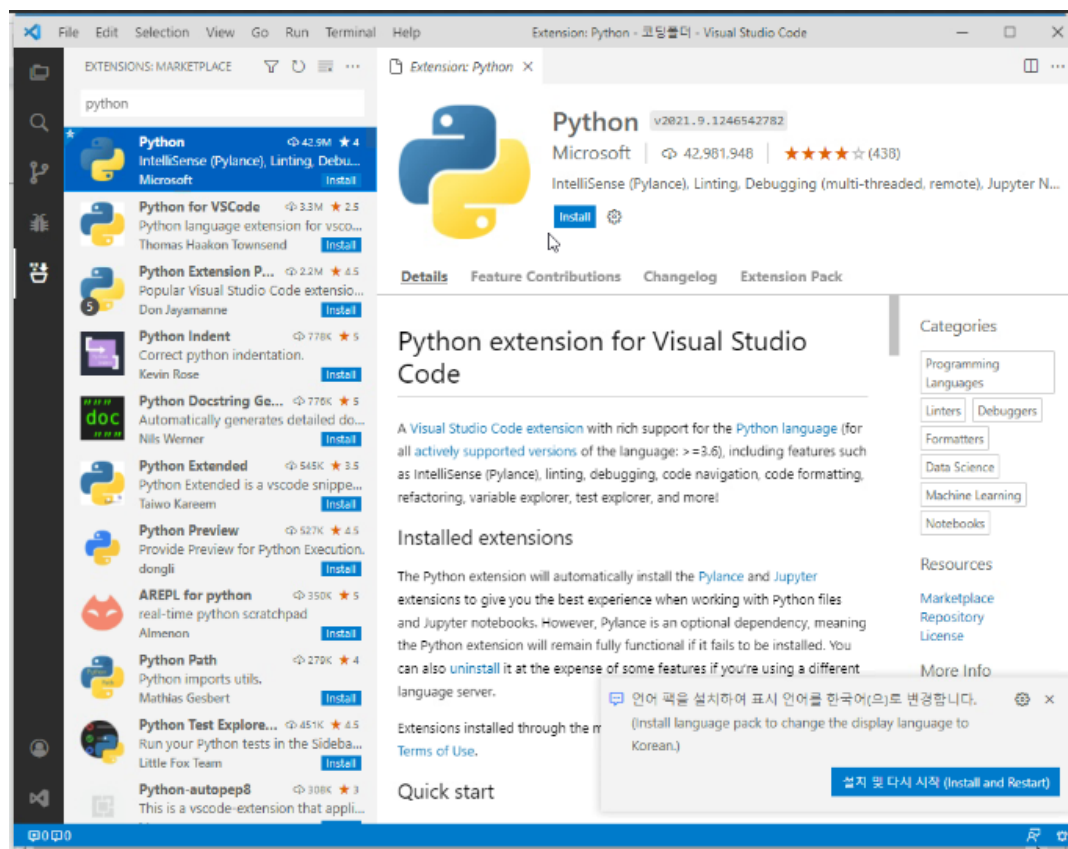
접속된 PC의 환경에 따라 아래와 같이 다운로드 버튼이 다르게 보입니다. 각

PC에 맞는 다운로드 버튼을 클릭 후, 설치를 진행하면 됩니다.

- 윈도우의 경우: Download for Windows
- Mac의 경우: Download for macOS
- 리눅스의 경우: Download for Linux

[2] 파이썬 설치

VS Code를 실행 후, 패키지 설치 버튼을 클릭 한 후, 파이썬 (Python)을 찾아 설치(install) 진행 (공급자가, Microsoft 이어야 함.)



- [3] VS Code는 폴더 단위로 코드를 관리합니다. 작업하고자 하는 폴더를 생성 후, Open Folder로 해당 폴더를 확인 후, 원하는 파이썬 파일(확장자 명 py)를 만들어 작업을 시작할 수 있습니다.

3강. 우리가 컴퓨터와 대화하는 법

실습 코딩실습 01 실습환경 사용하기

문제 실습환경을 처음 사용해보는 연습 문제입니다. 아래 순서에 맞춰 실습 환경인 onlinegdb.com을 사용하는 연습을 해 보세요.

- 아래와 같이 파이썬 코드를 입력 하세요.
- 실행(RUN)을 클릭 한 후, 결과를 확인해 보세요.
- 완성된 코드는 코딩실습01.py로 저장하세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습01](#)

파일 03강_코딩실습01.py

```
print("첫 코딩")  
print("안녕하세요")
```

```
첫 코딩  
안녕하세요
```

4강. 컴퓨터가 기억하는 방법 [1] - 기본 변수형

실습 코딩실습 02 조건문 if ~ else

문제 앞에서 배운 내용을 생각하며 메모리 그릇에 값을 담고, 담은 값을 print()로 출력해 보세요!

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습02](#)

파일 04강_코딩실습02.py

```
파인애플 = 5  
print(파인애플)
```

```
복숭아 = '맛있어'  
print(복숭아)
```

```
키위 = 10  
print(키위)
```

```
파인애플 = 15  
print(파인애플)
```

```
5  
맛있어  
10  
15
```

5강. 컴퓨터가 기억하는 방법 [2] - 문자열

실습 코딩실습 03 문자열 인덱싱, 슬라이싱

문제 '서양식인사' 라는 문자열 메모리 그릇을 만들고, 슬라이싱해서 '짧은서양인사'라는 새로운 문자열 메모리 그릇을 만들교자 합니다. 다음 코드를 입력하고 실행해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습03](#)

파일 05강_코딩실습03.py

```
서양식인사 = '좋은 아침입니다!'
print(서양식인사)
```

```
짧은서양인사 = 서양식인사[0:5] + 서양식인사[-1]
print(짧은서양인사)
```

```
좋은 아침입니다!
좋은 아침!
```

6강. 컴퓨터가 행동하는 방법 [1] - 함수의 기본 형태

실습 코딩실습 04 함수를 만들고 사용하기

문제 return이 있는 함수와 없는 함수의 차이점을 직접 코딩하며 살펴보겠습니다. 다음 코드를 입력하고 실행해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습04](#)

파일 06강_코딩실습04.py

```
def 인사말_만들기(이름):  
    인사말 = '안녕하세요?' + 이름 + '님'  
    return 인사말
```

```
def 인사말_바로하기(이름):  
    인사말 = '안녕하세요?' + 이름 + '님'  
    print(인사말)
```

```
인사말 = 인사말_만들기('준이')  
print(인사말)
```

```
인사말_바로하기('정국')
```

```
안녕하세요?준이님  
안녕하세요?정국님
```


6강. 도전과제!

강의를 통해 함수에 기본값(default)를 입력할 수 있다고 했습니다. 그리고, 실제로는 이런 기본값을 활용해서 다양한 응용을 한다고 했습니다. 이번 도전 과제에서는 함수의 기본값을 만드는 것을 도전해 보겠습니다.

아래와 같이 코딩실습04의 함수에 2개의 값을 입력 받도록 수정해 보겠습니다.

파일: 06강_도전과제.py

```
def 인사말_만들기(이름, 소속):  
    인사말 = '안녕하세요?' + 소속 + '의 ' + 이름 + '님'  
    print(인사말)
```

```
인사말 = 인사말_만들기('준이', '베어유코딩')
```

```
안녕하세요?베어유코딩의 준이님
```

이렇게 만들어진 함수에 기본 값을 아래와 같이 추가하겠습니다.

```
def 인사말_만들기(이름='아무개', 소속='OO회사'):  
    인사말 = '안녕하세요?' + 소속 + '의 ' + 이름 + '님'  
    print(인사말)
```

```
인사말 = 인사말_만들기()  
인사말 = 인사말_만들기(소속= '베어유코딩')  
인사말 = 인사말_만들기(이름= '준이')  
인사말 = 인사말_만들기(소속= '베어유코딩', 이름= '준이')  
인사말 = 인사말_만들기('준이', '베어유코딩')
```

```
안녕하세요?OO회사의 아무개님  
안녕하세요?베어유코딩의 아무개님
```

```
안녕하세요?OO회사의 준이님
안녕하세요?베어유코딩의 준이님
안녕하세요?베어유코딩의 준이님
```

기본값이 지정되면, 함수를 다양하게 응용할 있다는 것이 이해가 되나요?
혹시 이해가 안되시면, 동영상에서 기본값 지정 부분에 대해서 재학습 해주세요! ^^

7강. 컴퓨터가 행동하는 방법 [2] - 입력, 출력함수

실습 코딩실습 05 입력과 출력 연습하기

문제 다음과 같이 나이와 이름을 입력받아서 출력하는 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습05](#)

파일 07강_코딩실습05.py

```
이름 = input('당신의 이름은?')
나이 = int(input('당신의 나이는?'))

print('당신은' + 이름 + '이고 ' + str(나이) + '살입니다.')
print('당신은', 이름, '이고 ', 나이, '살입니다.')
print('당신은 {}이고, {}살 입니다.'.format(이름, 나이))
```

```
당신의 이름은?홍길동
당신의 나이는?20
당신은홍길동이고 20살입니다.
당신은 홍길동 이고 20 살입니다.
당신은 홍길동이고 20살 입니다.
```

8강. 조건문 이해하기[1] - 기본구조

실습 코딩실습 06 조건문 if ~ else

문제 게임 등수에 따라 1등은 TV를 보면서 쉬고, 2등은 설거지를 하게 하려고 합니다.
아래 코드를 직접 작성하고 [RUN]을 눌러보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습06](#)

파일 08강_코딩실습06.py

```
등수 = input('몇 등인가요(1 또는 2)?')

if 등수 == '1':
    print('TV를 보며 편하게 쉬세요.')
else:
    print('설거지 당첨!')
```

```
몇 등인가요(1 또는 2)?2
설거지 당첨!
```

9강. 조건문 이해하기[2] - 응용구조A

실습 코딩실습 07 조건이 2개 일때

문제 학교에서 간식으로 1, 5, 6학년에게는 김밥을, 2, 3, 4학년에게는 햄버거를 주려고 합니다. 비교연산자(>=, <=)와 and를 사용해 조건문 코드를 입력해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습07](#)

파일 09강A_코딩실습07.py

```
학년 = int(input('몇 학년인가요(1~6)?'))
```

```
if 학년 >=2 and 학년 <=4:  
    print('햄버거를 드세요.')  
else:  
    print('김밥을 드세요.')
```

```
몇 학년인가요(1~6)?2  
햄버거를 드세요.
```

9강. 조건문 이해하기[2] - 응용구조B

실습 코딩실습 08 조건문 elif

문제 지영이와 은수가 저녁을 먹으려고 합니다. 피자 가게가 열렸으면 피자를 먹고, 그렇지 않으면 치킨을 먹고, 치킨 가게도 닫았으면, 편의점에서 라면을 먹기로 했습니다. elif를 사용해 저녁 메뉴를 결정하는 조건문 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습08](#)

파일 09강B_코딩실습08.py

```
피자가게 = input('피자 가게가 열렸나요(예/아니요)?')
치킨가게 = input('치킨 가게가 열렸나요(예/아니요)?')
```

```
if 피자가게 == '예':
    print('피자 가게로 가자')
elif 치킨가게 == '예':
    print('치킨 가게로 가자 ')
else:
    print('편의점에서 라면을 먹자')
```

```
피자 가게가 열렸나요(예/아니요)?아니요
치킨 가게가 열렸나요(예/아니요)?아니요
편의점에서 라면을 먹자
```

9장. 도전과제!

피자가게가 열려 있다면, 굳이 치킨 가게가 열려 있는지 확인할 필요 없잖아요? 그럼 피자가게의 입력값이 '아니요'인 경우만 치킨 가게를 물어보며는 코드는 어떻게 되나요? 아래 정답 코드를 보기 전에 한번 도전해 보세요!

파일: 09장_도전과제.py

```
피자가게 = input('피자 가게가 열렸나요(예/아니요)?')

if 피자가게 == '예':
    print('피자 가게로 가자')
else:
    치킨가게 = input('치킨 가게가 열렸나요(예/아니요)?')

    if 치킨가게 == '예':
        print('치킨 가게로 가자 ')
    else:
        print('편의점에서 라면을 먹자')
```

```
피자 가게가 열렸나요(예/아니요)?예
피자 가게로 가자
```

해설: 피자가게의 상태에 따라 치킨가게의 상태 입력을 선택적으로 보여주기 위해서는 피자가게의 조건문 중 else 안에, 치킨가게의 input 을 추가하면 됩니다.

10강. 반복문 이해하기 [1] – 기본구조

실습 코딩 실습 09 반복문 for

문제 입력한 숫자만큼 식권이 출력되는 식권 자판기를 만들려고 합니다. 반복문 for를 사용해 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습09](#)

파일 10강_코딩실습09.py

```
전체식권수 = int(input('식권이 몇 장 필요한가요? '))
```

```
for 출력식권 in range(1, 전체식권수 + 1):  
    print('[식권] 번호:', 출력식권)
```

```
식권이 몇 장 필요한가요? 5
```

```
[식권] 번호: 1
```

```
[식권] 번호: 2
```

```
[식권] 번호: 3
```

```
[식권] 번호: 4
```

```
[식권] 번호: 5
```

11강. 반복문 이해하기 [2] - 응용구조A

실습 코딩 실습 10 반복문 안의 반복문

문제 입력한 두 개의 숫자 사이의 구구단을 출력하는 반복문 속 반복문 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습10](#)

파일 11강A_코딩실습10.py

```
시작단 = int(input('구구단 시작 단을 입력하세요(1~9): '))
끝단 = int(input('구구단 끝 단을 입력하세요(1~9): '))

출력값 = ''

for 앞수 in range(시작단, 끝단+ 1):
    for 뒷수 in range(1, 10):
        지금값 = '{}x{}={}Wn'.format(앞수, 뒷수, 앞수*뒷수)
        출력값 = 출력값 + 지금값

print(출력값)
```

```
구구단 시작 단을 입력하세요(1~9): 3
구구단 끝 단을 입력하세요(1~9): 4
3x1=3
3x2=6
3x3=9
3x4=12
3x5=15
3x6=18
3x7=21
3x8=24
3x9=27
4x1=4
4x2=8
```


4x3=12

4x4=16

4x5=20

4x6=24

4x7=28

4x8=32

4x9=36

11강. 반복문 이해하기 [2] - 응용구조B

실습 코딩 실습 11 조건문 if 와 반복문 for

문제 2단부터 9단까지 구구단을 출력하려고 합니다. 각 단 사이에만 구분선을 원하는 모양으로 넣을 수 있는 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습11](#)

파일 11강B_코딩실습11.py

```
줄스타일 = input('각 단을 구분하는 줄 모양을 입력하세요: ')
```

```
출력값 = ""
```

```
for 앞수 in range(2, 10):
```

```
    for 뒷수 in range(1, 10):
```

```
        지금값 = '{}x{}={}Wn'.format(앞수, 뒷수, 앞수*뒷수)
```

```
        출력값 += 지금값
```

```
    if 앞수 == 9:
```

```
        pass
```

```
    else:
```

```
        출력값 += 줄스타일 + 'Wn'
```

```
print(출력값)
```

```
각 단을 구분하는 줄 모양을 입력하세요: *****
```

```
2x1=2
```

```
2x2=4
```

```
2x3=6
```

```
2x4=8
```

```
2x5=10
```

```
2x6=12
```

```
2x7=14
```

```
2x8=16
```

```
2x9=18
```

$$3 \times 1 = 3$$

$$3 \times 2 = 6$$

$$3 \times 3 = 9$$

$$3 \times 4 = 12$$

$$3 \times 5 = 15$$

$$3 \times 6 = 18$$

$$3 \times 7 = 21$$

$$3 \times 8 = 24$$

$$3 \times 9 = 27$$

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$4 \times 6 = 24$$

$$4 \times 7 = 28$$

$$4 \times 8 = 32$$

$$4 \times 9 = 36$$

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$$6 \times 4 = 24$$

$$6 \times 5 = 30$$

6x6=36

6x7=42

6x8=48

6x9=54

7x1=7

7x2=14

7x3=21

7x4=28

7x5=35

7x6=42

7x7=49

7x8=56

7x9=63

8x1=8

8x2=16

8x3=24

8x4=32

8x5=40

8x6=48

8x7=56

8x8=64

8x9=72

9x1=9

9x2=18

9x3=27

9x4=36

9x5=45

9x6=54

9x7=63

9x8=72

9x9=81

12강. 배열 형태 자료구조[1]- 연결된 자료형A

실습 코딩 실습 13 'in'과 문자형 곱셈

문제 임의의 글을 입력 받은 다음 빈칸을 제외한 글자 수를 세고, 글자 수만큼 입력 받은 글을 복사해서 출력하는 코드를 작성해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습13](#)

파일 12강A_코딩실습13.py

```
문장 = input('아무 글이나 입력하세요: ')
글자수 = 0
```

```
for 한글자 in 문장:
    if 한글자 != ' ':
        글자수 += 1
```

```
print(문장 * 글자수)
```

```
아무 글이나 입력하세요: 첫 코딩
첫 코딩첫 코딩첫 코딩
```

12강. 배열 형태 자료구조[1]- 연결된 자료형B

실습 코딩 실습 14 파이썬의 다양한 자료형 이해하기: 1. 리스트

문제 이번에는 특별히 리스트, 튜플 자료형의 특징을 이해하기 위해 2번의 실습을 연속으로 해 보겠습니다. 먼저 다음 코드를 입력하고 실행하면서 리스트의 특징을 살펴보세요

해설 <https://youtu.be/2FJvHuP1xKI>

파일 12강B_코딩실습14_1.py

(1) 선언, 인덱싱하기

변수 '성적'에 4개의 값을 넣었습니다. 기호로 대괄호 []를 사용했으니 리스트라는 걸 알겠지요? 또한 첫 번째 값의 번호가 0부터 시작하기 때문에 성적[1]을 출력하면 첫 번째 값이 아닌 두 번째 값 '결석'이 출력됨을 확인할 수 있습니다.

```
성적 = [60,'결석', 60, 70]
print(성적[1])
```

결석

(2) 키워드 in으로 값 출력

문자열과 마찬가지로 'in'을 사용해서 변수 '성적' 안의 모든 값을 출력할 수 있습니다.

```
성적 = [60,'결석', 60, 70]
print(성적[1])
```

```
for 안의값 in 성적:
    print(안의값)
```

결석
60
결석
60
70

(3) 인덱싱 후 값 수정하기

변수 '성적'의 네 번째 값도 90으로 변경할 수 있습니다.

```
성적 = [60, '결석', 60, 70]  
print(성적[1])
```

```
for 안의값 in 성적:  
    print(안의값)
```

```
성적[3] = 90  
print(성적[3])
```

```
결석  
60  
결석  
60  
70  
90
```


12강. 배열 형태 자료구조[1]- 연결된 자료형C

실습 코딩 실습 14 파이썬의 다양한 자료형 이해하기: 2. 튜플

문제 이번에는 특별히 리스트, 튜플 자료형의 특징을 이해하기 위해 2번의 실습을 연속으로 해 보겠습니다. 이번에는 리스트 코드에 이어서 튜플을 실습해 보겠습니다. 앞서 입력한 리스트 코드에서 대괄호 []를 소괄호 ()로 바꿔 보세요. 리스트가 튜플로 바뀝니다.

해설 <https://youtu.be/2FJvHuP1xKI?t=133>

파일 12강C_코딩실습14_2.py

```
성적 = (60, '결석', 60, 70)
print(성적[1])
```

```
for 안의값 in 성적:
    print(안의값)
```

```
성적[3] = 90
print(성적[3])
```

```
결석
60
결석
60
70
Traceback (most recent call last):
  File "main.py", line 8, in <module>
    성적[3] = 90
TypeError: 'tuple' object does not support item assignment
```

해설: 리스트와 모든 결과가 동일한데, 마지막 '인덱싱 후 값을 수정하는 코드'에서 오류가 발생합니다. 리스트와 달리 튜플은 선언 후 값을 바꿀 수 없기 때문입니다.

13강. 배열 형태 자료구조[2] – 비연결 자료형A

실습 코딩 실습 14 파이썬의 다양한 자료형 이해하기: 3. 집합

문제 집합의 특성 이해하기

해설 <https://youtu.be/2FJvHuP1xKI?t=178>

파일 13강_코딩실습14_3.py

(1) 12강에서 입력한 튜플 코드에서 소괄호 ()를 다시 대괄호 []로 고치고, 집합을 의미하는 set()으로 감싸 보세요. 집합 자료형으로 바뀝니다.

```
성적 = set([60,'결석',60, 70])
```

```
print(성적[1])
```

```
for 안의값 in 성적:  
    print(안의값)
```

```
성적[3] = 90  
print(성적[3])
```

Traceback (most recent call last):

File "main.py", line 3, in <module>

print(성적[1])

TypeError: 'set' object does not support indexing

해설: [실행]을 누르니 처음부터 오류가 발생합니다! 집합 자료형에는 순서가 없기 때문이에요.

(2) 다음과 같이 인덱스 번호를 사용하는 부분을 모두 주석으로 처리해 보세요.

```
성적 =set([60,'결석',60, 70])
```

```
#print(성적[1])
```

비법노트: 파이썬 코딩 클래스

```
for 안의값 in 성적:  
    print(안의값)
```

```
#성적[3] = 90  
#print(성적[3])
```

```
60  
결석  
70
```

해설:문제가 되는 부분을 모두 주석으로 처리했으니 오류 메시지 없이 결과가 나옵니다.
정상적으로 실행된 부분은 키워드 in을 이용한 값 출력 부분이에요. 앓 그런데 '성적'에 있는 값은 4개인데 값은 3개만 출력되네요! 집합 자료형은 중복된 값을 하나로 생각하기 때문입니다.

13강. 배열 형태 자료구조[2] – 비연결 자료형B

실습 코딩 실습 14 파이썬의 다양한 자료형 이해하기: 4. 딕셔너리

문제 딕셔너리 특성 이해하기

해설 <https://youtu.be/2FJvHuP1xKI?t=252>

파일 13강_코딩실습14_4.py

(1) 이번에는 기호를 중괄호 {}로 변경하고 다음과 같이 각 성적 값에 해당하는 이름을 넣어 보세요. 딕셔너리 자료형으로 바뀝니다.

```
성적 = {'승준':60, '은희':'결석', '태호':60, '지영':70}

print(성적[1])

for 안의값 in 성적:
    print(안의값)

성적[3] = 90
print(성적[3])
```

```
Traceback (most recent call last):
  File "main.py", line 3, in <module>
    print(성적[1])
KeyError: 1
```

해설: 이번에도 에러가 발생했습니다. 딕셔너리는 집합과 마찬가지로 인덱스 번호로 값을 불러오지 않기 때문입니다.

(2) 인덱스 번호 대신에, [1]에 해당하는 key인 '은희'를 입력해 보세요.

```
성적 = {'승준':60, '은희':'결석', '태호':60, '지영':70}

print(성적['은희'])
```

```
for 안의값 in 성적:  
    print(안의값)
```

```
성적[3] = 90  
print(성적[3])
```

```
결석  
은희  
태호  
지영  
승준  
90
```

해설: '은희'의 value인 '결석'이 가장 먼저 출력됩니다. 또한 in 키워드로 '성적'의 값들이 모두 잘 출력됩니다.

(3) 그런데 이상하네요. 분명 인덱스 번호로 호출이 안 된다고 했는데 성적[3]이라고 입력한 마지막 코드가 정상적으로 실행됩니다. 어떻게 된 일일까요? 리스트와 달리 딕셔너리에서 이 코드는 세 번째 값을 호출하는 코드가 아니라, key 값이 3이고 value 값이 90인 '새로운' 메모리 그릇을 만드는 코드이기 때문입니다. 위 출력 결과를 메모리 그릇으로 표현하면 다음과 같습니다.



마지막으로 반복문으로 딕셔너리의 값을 출력했을 때, key가 아니라 value 값을 출력하고 싶다면 아래와 같이 수정하면 됩니다.

```
성적 = {'승준':60, '은희':'결석', '태호':60, '지영':70}
```

```
print(성적['은희'])
```

```
for 안의값 in 성적.values():  
    print(안의값)
```

```
성적[3] = 90  
print(성적[3])
```

```
결석
```

```
60
```

```
결석
```

```
60
```

```
70
```

```
90
```

해설: 리스트, 튜플, 딕셔너리, 집합 자료형의 특징이 이제 이해가 되나요? 비슷하지만 조금은 다른 형태로 메모리에 저장된다는 점, 선언하는 방식은 다르지만 사용하는 방식은 비슷하다는 점을 이해하면 되겠습니다.

13강. 배열 형태 자료구조[2] – 비연결 자료형C

실습 코딩 실습 15 바꿔치기된 메모리 그릇 확인하기

문제 바꿔치기된 메모리 그릇 확인하기

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습15](#)

파일 13강_코딩실습15.py

(1) 아래 코드를 실행해 바꿔치기된 메모리 그릇을 확인해 보세요.

```
숫자 = 90  
print(id(숫자))
```

```
숫자 = 100  
print(id(숫자))
```

```
10107936
```

```
10108256
```

해설: 출력결과는 실습 환경마다 다르게 출력됩니다. 서로 다른 숫자가 나오는 것에 주목하세요!

(2) 이번엔 리스트 자료를 입력해 메모리 그릇의 주소를 확인해 보세요.

```
리스트 = [1, 2, 3]  
print(리스트)  
print(id(리스트))
```

```
리스트[2] = 6  
print(리스트)  
print(id(리스트))
```

```
[1, 2, 3]
```

```
140182914363976
```

```
[1, 6, 3]
```

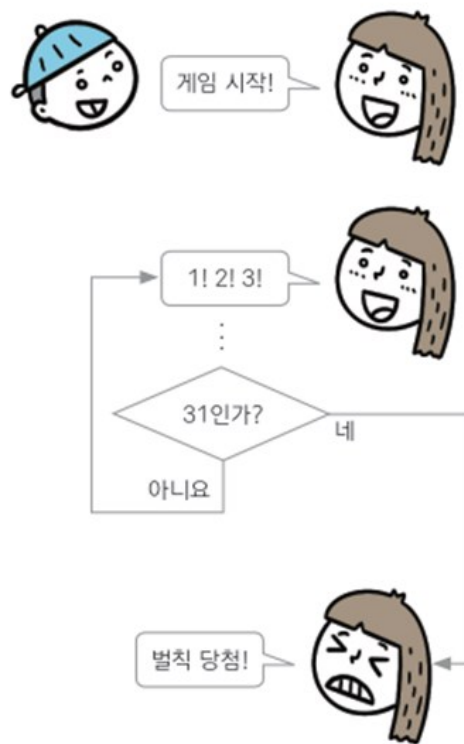
```
140182914363976
```

해설: 정수 자료를 입력했을 때와 달리, 변수 '리스트'의 값을 수정해도 주소가 변경되지 않습니다.

14강 베라31 게임만들기 [1] – 게임 틀 만들기, 랜덤함수A

실습 코딩실습 BR31 Step01

문제 Step01. 기본 틀 만들기



해설 강의동영상 참고

파일 14강A_코딩실습_BR31_STEP01.py

```
게임수 = 0
```

```
while True:
```

```
    나의선택 = int(input('A(1~3)?'))
```

```
    게임수 += 나의선택
```

```
    if 게임수 >= 31:
```

```
        break
```

```
print('벌칙 당첨')
```

```
A(1~3)?3
```

```
A(1~3)?2
```

```
A(1~3)?3
```

```
A(1~3)?3
```

```
A(1~3)?1
```

```
A(1~3)?2
```

```
A(1~3)?3
```

```
A(1~3)?3
```

```
A(1~3)?2
```

```
A(1~3)?1
```

```
A(1~3)?2
```

```
A(1~3)?3
```

```
A(1~3)?1
```

```
A(1~3)?3
```

```
벌칙 당첨
```

14강 베라31 게임만들기 [1] – 게임 틀 만들기, 랜덤함수B

실습 코딩실습 BR31 Step02

문제 Step02. 구호넣기

```
main.py
1 게임수 = 0
2
3 while True:
4     나의선택 = int(input('(1~3)?'))
5
6     for n in range(나의선택):
7         print(게임수 + n + 1, '!')
8
9     게임수 += 나의선택
10
11     if 게임수 >= 31:
12         break
13
14 print('벌칙 당첨')
```



해설 강의동영상 참고

파일 14강B_코딩실습_BR31_STEP02.py

게임수 = 0

while True:

 나의선택 = int(input('A(1~3)?'))

 for n in range(나의선택):
 print(게임수 + n + 1, '!')

 게임수 += 나의선택

 if 게임수 >= 31:
 break

print('벌칙 당첨')

A(1~3)?1

1 !

A(1~3)?2

2 !

3 !

A(1~3)?3

4 !

5 !

6 !

A(1~3)?3

7 !

8 !

9 !

A(1~3)?2

10 !

11 !

A(1~3)?3

12 !

13 !

14 !

A(1~3)?1

15 !

A(1~3)?3

16 !

17 !

18 !

A(1~3)?2

19 !

20 !

A(1~3)?2

21 !

22 !

A(1~3)?1

23 !

A(1~3)?3

24 !

```
25 !  
26 !  
A(1~3)?2  
27 !  
28 !  
A(1~3)?2  
29 !  
30 !  
A(1~3)?1  
31 !  
별칭 당첨
```

14강 베라31 게임만들기 [1] – 게임 틀 만들기, 랜덤함수C

실습 코딩실습 BR31 Step03

문제 Step03. 두번째 참가자 추가하기

```
main.py
1  게임수 = 0
2
3  while True:
4      A선택 = int(input('A(1~3)?'))
5
6      for n in range(A선택):
7          print('A', 게임수 + n + 1, '!')
8
9      게임수 += A선택
10
11     if 게임수 >= 31:
12         print('A 벌칙당침!')
13         break
14
15     B선택 = int(input('B(1~3)?'))
16
17     for n in range(B선택):
18         print('B', 게임수 + n + 1, '!')
19
20     게임수 += B선택
21
22     if 게임수 >= 31:
23         print('B 벌칙당침!')
24         break
25
```



해설 강의동영상 참고

파일 14강C_코딩실습_BR31_STEP03.py

게임수 = 0

while True:

A선택 = int(input('A(1~3)?'))

for n in range(A선택):
 print('A', 게임수 + n + 1, '!')

게임수 += A선택

if 게임수 >= 31:
 print('A 벌칙 당침')
 break

```
B선택 = int(input('B(1~3)?'))
```

```
for n in range(B선택):  
    print('B',게임수 + n + 1, '!')
```

```
게임수 += B선택
```

```
if 게임수 >= 31:  
    print('B 벌칙 당첨')  
    break
```

A(1~3)?3

A 1 !

A 2 !

A 3 !

B(1~3)?2

B 4 !

B 5 !

A(1~3)?1

A 6 !

B(1~3)?3

B 7 !

B 8 !

B 9 !

A(1~3)?2

A 10 !

A 11 !

B(1~3)?3

B 12 !

B 13 !

B 14 !

A(1~3)?3

A 15 !

A 16 !

A 17 !

B(1~3)?2

```
B 18 !  
B 19 !  
A(1~3)?1  
A 20 !  
B(1~3)?2  
B 21 !  
B 22 !  
A(1~3)?2  
A 23 !  
A 24 !  
B(1~3)?3  
B 25 !  
B 26 !  
B 27 !  
A(1~3)?3  
A 28 !  
A 29 !  
A 30 !  
B(1~3)?1  
B 31 !  
B 벌칙 당첨
```


14강 베라31 게임만들기 [1] – 게임 틀 만들기, 랜덤함수D

실습 코딩실습 BR31 Step04

문제 Step04. 두번째 참가자를 컴퓨터로

해설 강의동영상 참고

파일 14강D_코딩실습_BR31_STEP04.py

```
import random

게임수 = 0

while True:

    A선택 = int(input('A(1~3)?'))

    for n in range(A선택):
        print('A', 게임수 + n + 1, '!')

    게임수 += A선택

    if 게임수 >= 31:
        print('A 벌칙 당첨')
        break

    B선택 = random.randint(1,3)

    for n in range(B선택):
        print('컴', 게임수 + n + 1, '!')

    게임수 += B선택

    if 게임수 >= 31:
        print('컴 벌칙 당첨')
```

break

A(1~3)?3

A 1 !

A 2 !

A 3 !

컴 4 !

A(1~3)?2

A 5 !

A 6 !

컴 7 !

컴 8 !

A(1~3)?3

A 9 !

A 10 !

A 11 !

컴 12 !

컴 13 !

A(1~3)?1

A 14 !

컴 15 !

컴 16 !

A(1~3)?2

A 17 !

A 18 !

컴 19 !

컴 20 !

컴 21 !

A(1~3)?3

A 22 !

A 23 !

A 24 !

컴 25 !

컴 26 !

컴 27 !

A(1~3)?2

A 28 !

A 29 !

컴 30 !

컴 31 !

15강 베라31 게임만들기 [2] – 예외처리, 잠재우기A

실습 코딩실습 BR31 Step05

문제 Step05. 문자열 예외 처리하기

```
main.py  BR01_예외처리.py ⋮
1  import random
2
3  게임수 = 0
4
5  while True:
6
7      while True:
8          try:
9              A선택 = int(input('A(1~3)?'))
10             break
11         except:
12             print('[경고]숫자만 입력하세요!')
13
14     for n in range(A선택):
15         print('A', 게임수 + n + 1, '!')
16
```

해설 강의동영상 참고

파일 15강A_코딩실습_BR31_STEP05.py

```
import random
```

```
게임수 = 0
```

```
while True:
```

```
    while True:
```

```
        try:
```

```
            A선택 = int(input('A(1~3)?'))
```

```
            break
```

```
        except:
```

```
            print('[경고]숫자만 입력하세요!')
```

```
    for n in range(A선택):
```

```
        print('A', 게임수 + n + 1, '!')
```

게임수 += A선택

```
if 게임수 >= 31:  
    print('A 벌칙 당첨')  
    break
```

B선택 = random.randint(1,3)

```
for n in range(B선택):  
    print('컴', 게임수 + n + 1, '!')
```

게임수 += B선택

```
if 게임수 >= 31:  
    print('컴 벌칙 당첨')  
    break
```

```
A(1~3)?몰라  
[경고]숫자만 입력하세요!  
A(1~3)?싫어요  
[경고]숫자만 입력하세요!  
A(1~3)?3  
A 1 !  
A 2 !  
A 3 !  
컴 4 !  
컴 5 !  
컴 6 !  
A(1~3)?2  
A 7 !  
A 8 !  
컴 9 !  
컴 10 !  
A(1~3)?글쎄  
[경고]숫자만 입력하세요!  
A(1~3)?3  
A 11 !  
A 12 !  
A 13 !
```

```
컴 14 !  
컴 15 !  
A(1~3)?2  
A 16 !  
A 17 !  
컴 18 !  
A(1~3)?3  
A 19 !  
A 20 !  
A 21 !  
컴 22 !  
컴 23 !  
A(1~3)?5  
A 24 !  
A 25 !  
A 26 !  
A 27 !  
A 28 !  
컴 29 !  
컴 30 !  
A(1~3)?1  
A 31 !  
A 벌칙 당첨
```

15강 베라31 게임만들기 [2] – 예외처리, 잠재우기B

실습 코딩실습 BR31 Step06

문제 Step06. 조건 입력 추가하기

```
main.py
1  import random
2
3  게임수 = 0
4
5  while True:
6
7      while True:
8          try:
9              A선택 = int(input('A(1~3)?'))
10
11              if A선택 == 1 or A선택 == 2 or A선택 == 3:
12                  break
13              else:
14                  print('[경고]1,2,3 숫자만 입력하세요!')
15          except:
16              print('[경고]숫자만 입력하세요!')
17
18      for n in range(A선택):
```

해설 강의동영상 참고

파일 15강B_코딩실습_BR31_STEP06.py

```
import random

게임수 = 0

while True:

    while True:
        try:
            A선택 = int(input('A(1~3)?'))

            if A선택 == 1 or A선택 == 2 or A선택 == 3:
                break
```

```
    else:
        print('[경고]1,2,3 숫자만 입력하세요!')
    except:
        print('[경고]숫자만 입력하세요!')

for n in range(A선택):
    print('A',게임수 + n + 1, '!')

    게임수 += A선택

if 게임수 >= 31:
    print('A 벌칙 당첨')
    break

B선택 = random.randint(1,3)

for n in range(B선택):
    print('컴',게임수 + n + 1, '!')

    게임수 += B선택

if 게임수 >= 31:
    print('컴 벌칙 당첨')
    break
```

```
A(1~3)?물라
[경고]숫자만 입력하세요!
A(1~3)?5
[경고]1,2,3 숫자만 입력하세요!
A(1~3)?2
A 1 !
A 2 !
컴 3 !
컴 4 !
컴 5 !
A(1~3)?3
A 6 !
A 7 !
A 8 !
컴 9 !
```


A(1~3)?1

A 10 !

컴 11 !

A(1~3)?3

A 12 !

A 13 !

A 14 !

컴 15 !

컴 16 !

A(1~3)?6

[경고]1,2,3 숫자만 입력하세요!

A(1~3)?2

A 17 !

A 18 !

컴 19 !

컴 20 !

컴 21 !

A(1~3)?3

A 22 !

A 23 !

A 24 !

컴 25 !

컴 26 !

컴 27 !

A(1~3)?1

A 28 !

컴 29 !

컴 30 !

컴 31 !

컴 벌칙 당첨

15강 베라31 게임만들기 [2] – 예외처리, 잠재우기C

실습 코딩실습 BR31 Step07

문제 Step07. 컴퓨터 생각하듯 만들기

```
main.py
1 import random
2 import time
3
4 게임수 = 0
5
6 while True:
7
8     while True:
9         try:
10             A선택 = int(input('A(1~3)?'))
11
12             if A선택 == 1 or A선택 == 2 or A선택 == 3:
13                 break
14             else:
15                 print('[경고]1,2,3 숫자만 입력하세요!')
16
17         except:
18             print('잘못 입력하셨습니다.')
19
20     B선택 = random.randint(1,3)
21     time.sleep(random.randint(1,4))
22     print('컴(1~3)?{}'.format(B선택))
23
24     if 게임수 >= 31:
25         print('A 벌칙당침!')
26         break
27
28     for n in range(B선택):
29         print('컴', 게임수 + n + 1, '!')
30
31     게임수 += B선택
32
33     if 게임수 >= 31:
34         print('컴 벌칙당침!')
35         break
```

해설 강의동영상 참고

파일 15강C_코딩실습_BR31_STEP07.py

```
import random
import time

게임수 = 0

while True:

    while True:
        try:
            A선택 = int(input('A(1~3)?'))

            if A선택 == 1 or A선택 == 2 or A선택 == 3:
                break
            else:
                print('[경고]1,2,3 숫자만 입력하세요!')

        except:
            print('잘못 입력하셨습니다.')
```

```
except:
    print('[경고]숫자만 입력하세요!')

for n in range(A선택):
    print('A',게임수 + n + 1, '!')

    게임수 += A선택

if 게임수 >= 31:
    print('A 벌칙당침!')
    break

B선택 = random.randint(1,3)
time.sleep(random.randint(1,4))

print('컴(1~3)?{}'.format(B선택))
for n in range(B선택):
    print('컴',게임수 + n + 1, '!')

    게임수 += B선택

if 게임수 >= 31:
    print('컴 벌칙당침!')
    break
```

```
A(1~3)?3
A 1 !
A 2 !
A 3 !
컴(1~3)?2
컴 4 !
컴 5 !
A(1~3)?몰라
[경고]숫자만 입력하세요!
A(1~3)?2
A 6 !
A 7 !
컴(1~3)?2
컴 8 !
컴 9 !
A(1~3)?6
[경고]1,2,3 숫자만 입력하세요!
```

A(1~3)?1

A 10 !

컴(1~3)?1

컴 11 !

A(1~3)?3

A 12 !

A 13 !

A 14 !

컴(1~3)?3

컴 15 !

컴 16 !

컴 17 !

A(1~3)?2

A 18 !

A 19 !

컴(1~3)?3

컴 20 !

컴 21 !

컴 22 !

A(1~3)?3

A 23 !

A 24 !

A 25 !

컴(1~3)?3

컴 26 !

컴 27 !

컴 28 !

A(1~3)?2

A 29 !

A 30 !

컴(1~3)?1

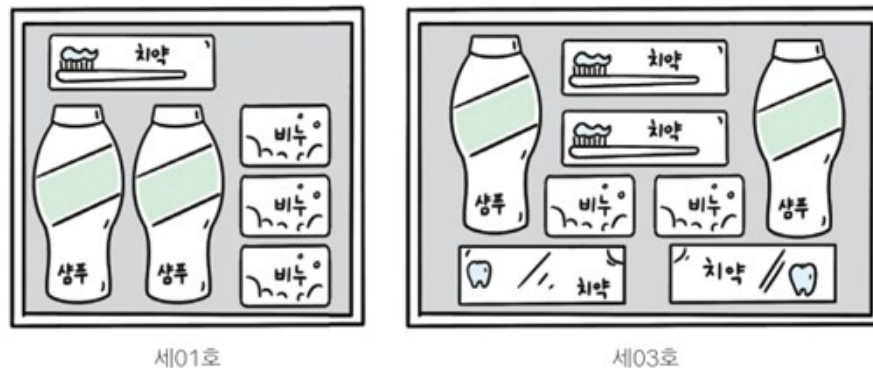
컴 31 !

컴 벌칙당침!

16강. 클래스 이해하기[1] – 기본 구조A

실습 코딩 실습 16 클래스 만들기

문제 치약, 샴푸, 비누로 구성된 세면도구세트를 클래스로 만들고 세01호, 세03호 인스턴스를 만들어서 출력해 보세요.



해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습16](#)

파일 16강A_코딩실습16.py

```
class 세면도구:
```

```
    치약 = 0
```

```
    샴푸 = 0
```

```
    비누 = 0
```

```
세01호 = 세면도구()
```

```
세01호.치약 = 1
```

```
세01호.샴푸 = 2
```

```
세01호.비누 = 3
```

```
세03호 = 세면도구()
```

```
세03호.치약 = 4
```

```
세03호.샴푸 = 2
```

```
세03호.비누 = 2
```

```
print('세면도구세트 01호 내용물')
```

```
print('치약:', 세01호.치약, '샴푸:', 세01호.샴푸, '비누:', 세01호.비누)
```

비법노트: 파이썬 코딩 클래스

```
print('세면도구세트 03호 내용물')  
print('치약:', 세03호.치약, '샴푸:', 세03호.샴푸, '비누:', 세03호.비누)
```

```
세면도구세트 01호 내용물  
치약: 1 샴푸: 2 비누: 3  
세면도구세트 03호 내용물  
치약: 4 샴푸: 2 비누: 2
```

16강. 클래스 이해하기[1] – 기본 구조B

실습 코딩 실습 17 메서드 만들기

문제 세면도구세트에서 내용물의 개수를 출력하는 메서드를 만들고 출력해 보세요.



세01호



세03호

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습17](#)

파일 16강B_코딩실습17.py

```
class 세면도구:
    치약 = 0
    샴푸 = 0
    비누 = 0

    def 내용물출력(self, 세트이름):
        print(세트이름)
        print(' 치약:', self.치약)
        print(' 샴푸:', self.샴푸)
        print(' 비누:', self.비누)
```

세01호 = 세면도구()

세01호.치약 = 1

세01호.샴푸 = 2

세01호.비누 = 3

세03호 = 세면도구()

세03호.치약 = 4

세03호.샴푸 = 2

세03호.비누 = 2

세01호.내용물출력('세면도구세트 01호')

세03호.내용물출력('세면도구세트 03호')

세면도구세트 01호

치약: 1

샴푸: 2

비누: 3

세면도구세트 03호

치약: 4

샴푸: 2

비누: 2

17강. 클래스 이해하기[2] – 생성자, 상속A

실습 코딩 실습 18 생성자 만들기

문제 세면도구세트에서 생성자를 이용해 '세01호'와 '세03호' 인스턴스를 만들고 세면도구의 내용을 개수를 출력해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습18](#)

파일 17강A_코딩실습18.py

```
class 세면도구:
    def __init__(self, 치약, 샴푸, 비누):
        self.치약 = 치약
        self.샴푸 = 샴푸
        self.비누 = 비누
```

```
    def 내용물출력(self, 세트이름):
        print(세트이름)
        print(' 치약:', self.치약)
        print(' 샴푸:', self.샴푸)
        print(' 비누:', self.비누)
```

```
세01호 = 세면도구(1,2,3)
```

```
세03호 = 세면도구(4,2,2)
```

```
세01호.내용물출력('세면도구세트 01호')
```

```
세03호.내용물출력('세면도구세트 03호')
```

```
세면도구세트 01호
```

```
치약: 1
```

```
샴푸: 2
```

```
비누: 3
```

```
세면도구세트 03호
```

```
치약: 4
```

```
샴푸: 2
```

```
비누: 2
```


17강. 클래스 이해하기[2] – 생성자, 상속B

실습 코딩 실습 19 상속하기

문제 이번 명절에는 특별히 면도기가 포함된 특별세트를 만들려고 합니다. 특별세트를 만드는 기계를 새로 제작하는 것보다 기존에 사용하던 '세면도구세트' 기계를 변형해 사용하는 게 비용이 덜 들겠죠?



특01호

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습19](#)

파일 17강B_코딩실습19.py

(1) 오버라이드 메소드 없이 상속받은 결과 출력하기

```
class 세면도구:
```

```
    def __init__(self, 치약, 샴푸, 비누):
        self.치약 = 치약
        self.샴푸 = 샴푸
        self.비누 = 비누
```

```
    def 내용물출력(self, 세트이름):
        print(세트이름)
        print(' 치약:', self.치약)
        print(' 샴푸:', self.샴푸)
        print(' 비누:', self.비누)
```

```
class 특별세트(세면도구):
```

```
def __init__(self, 치약, 샴푸, 비누, 면도기):  
    super().__init__(치약, 샴푸, 비누)  
    self.면도기 = 면도기
```

```
특01호 = 특별세트(1, 2, 3, 1)  
특01호.내용물출력('명절 특별세트 01호')
```

명절 특별세트 01호

치약: 1

샴푸: 2

비누: 3

해설: 결과 화면이 이상하지 않나요? 면도기 값이 출력되지 않았습니다. 그 이유는 치약, 비누, 샴푸 개수만 화면에 출력하는 ‘세면도구’ 클래스의 ‘내용물출력()’ 메서드를 사용했기 때문입니다.

(2) 면도기 값까지 화면에 나타내려면 ‘내용물출력()’ 메서드를 오버라이딩하면 됩니다. 다음과 같이 기존 ‘내용물출력()’ 메서드는 상속받아 사용하고, 면도기 값을 출력하는 코드를 추가해 보세요.

class 세면도구:

```
def __init__(self, 치약, 샴푸, 비누):  
    self.치약 = 치약  
    self.샴푸 = 샴푸  
    self.비누 = 비누
```

```
def 내용물출력(self, 세트이름):  
    print(세트이름)  
    print(' 치약:', self.치약)  
    print(' 샴푸:', self.샴푸)  
    print(' 비누:', self.비누)
```

class 특별세트(세면도구):

```
def __init__(self, 치약, 샴푸, 비누, 면도기):  
    super().__init__(치약, 샴푸, 비누)  
    self.면도기 = 면도기
```

```
def 내용물출력(self, 세트이름):  
    super().내용물출력(세트이름)  
    print(' 면도기:', self.면도기)
```

특01호 = 특별세트(1,2,3,1)

특01호.내용물출력('명절 특별세트 01호')

명절 특별세트 01호

치약: 1

삼푸: 2

비누: 3

면도기: 1

18강. 클래스 이해하기[3] - 패키지, 스레드

실습 코딩 실습 22 달리기 경주

문제 별 표시로 달리기 경주하는 아래 코드를 입력해 보세요.

해설 [Do It! 첫 코딩 with 파이썬 - 5-05 한번에 여러가지 하는 척 하기 스프레드](#)

파일 18강_코딩실습22.py

(1) 스레드 없이 출력하기

```
import time
```

class 오른쪽으로달리기:

```
def __init__(self, 나의이름):
    self.이동거리 = 나의이름 + '!'
```

```
def 달리기(self):
    for 숫자 in range(0, 200):
        self.이동거리 = self.이동거리 + '*'
        print(self.이동거리)
```

수호 = 오른쪽으로달리기('수호')

세훈 = 오른쪽으로달리기('세훈')

수호.달리기()

세훈.달리기()

[illegible]

비법노트: 파이썬 코딩 클래스

비법노트: 파이썬 코딩 클래스

비법노트: 파이썬 코딩 클래스

[illegible]

(2) 그런데 두 사람이 동시에 뛰지 않아서 공평해 보이지 않네요! 스레드를 이용해 수호와

비법노트: 파이썬 코딩 클래스

세훈이가 있는 힘껏 달리기 경기를 하는 상황을 만들어 보겠습니다.
아래 빨간색 부분의 코드를 수정하세요.

참고: 스레드는 적은 반복횟수에도 동작을 하나, 온라인 컴파일 환경에서 발생하는 네트워크 딜레이로 인해 적은 반복 횟수에서 스레드가 나타나지 않는 경우가 있습니다. 이 때문에 반복횟수를 비교적 큰 200으로 실시하였습니다. 동영상 설명서에서는 반복 횟수를 늘리는 대신에, sleep을 적용하였습니다.

```
import threading

class 오른쪽으로달리기(threading.Thread):

    def __init__(self, 나의이름):
        threading.Thread.__init__(self)
        self.이동거리 = 나의이름 + ':'

    def run(self):
        for 숫자 in range(0, 200):
            self.이동거리 = self.이동거리 + '*'
            print(self.이동거리)

수호 = 오른쪽으로달리기('수호')
세훈 = 오른쪽으로달리기('세훈')

수호.start()
세훈.start()
```

[illegible]

비법노트: 파이썬 코딩 클래스

비법노트: 파이썬 코딩 클래스

비법노트: 파이썬 코딩 클래스

19강. 클래스 이해하기[4] – 자료형 클래스의 매서드A

실습 코딩 실습 20 리스트에서 메서드 사용하기

문제 리스트 자료형은 대량의 자료를 다룰 때 많이 사용됩니다. 하루에 할 일을 리스트로 만들어서 리스트의 메서드를 연습해 보세요.

1. 처음 계획
 - a. 기상
 - b. 아침 식사
 - c. 책읽기
 - d. 점심 식사
2. [하나 추가] 점심 식사 후에 운동을 넣어야 겠다.
3. [두개 추가] 운동 후에 저녁 일정 '게임', '잠' 도 넣자.
4. [중간 삽입]아침 먹고 '샤워'
5. [삭제] '게임' 할 시간은 없으니 삭제
6. [삭제] 오늘은 바쁘니, '아침 식사'는 거르자
7. [확인1] 저녁 일정 부터 거꾸로 출력
8. [확인2] 가,나,다 순으로 일정 출력

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습20](#)

파일 19강A_코딩실습20.py

```
할일 = ['기상', '식사', '책읽기', '식사'] # 건강 식단, 하루 두 끼

print("'책읽기'의 인덱스는 {}입니다.".format(할일.index('책읽기')))
print("할일 중 '식사'는 {}번 있습니다.".format(할일.count('식사')))

할일.append('운동') # 식사 후 운동이 빠졌네!
print(할일)

할일.extend(['게임', '잠']) # 운동 후 저녁 일정도 넣자
print(할일)
```

비법노트: 파이썬 코딩 클래스

```
할일.insert(2, '샤워') # 참, 아침엔 샤워를 해야지  
print(할일)
```

```
할일.remove('게임') # 게임할 시간은 없을 것 같아  
print(할일)
```

```
할일.pop(1) # 오늘은 바쁘니 아침 식사는 빼자  
print(할일)
```

```
할일.reverse() # 저녁 일정부터 반대로 정리  
print(할일)
```

```
할일.sort() # 가나다 순으로 정리  
print(할일)
```

'책읽기'의 인덱스는 2입니다.

할일 중 '식사'는 2번 있습니다.

['기상', '식사', '책읽기', '식사', '운동']

['기상', '식사', '책읽기', '식사', '운동', '게임', '잠']

['기상', '식사', '샤워', '책읽기', '식사', '운동', '게임', '잠']

['기상', '식사', '샤워', '책읽기', '식사', '운동', '잠']

['기상', '샤워', '책읽기', '식사', '운동', '잠']

['잠', '운동', '식사', '책읽기', '샤워', '기상']

['기상', '샤워', '식사', '운동', '잠', '책읽기']

19강. 클래스 이해하기[4] – 자료형 클래스의 매서드B

실습 코딩 실습 21 딕셔너리에서 메서드 사용하기

문제 딕셔너리는 짝을 이루는 자료를 저장할 때 사용됩니다. 이번 실습에서는 딕셔너리의 메서드 연습을 위해 개인정보에 관한 짧은 딕셔너리를 만들어 보겠습니다.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습21](#)

파일 19강B_코딩실습21.py

```
이름 = input('이름: ')
나이 = int(input('나이: '))

개인정보 = {'이름':이름, '나이':나이}

for 키만 in 개인정보.keys():
    print('- 키: ', 키만)

for 벨류만 in 개인정보.values():
    print('- 값:', 벨류만)

주소 = input('주소: ')
개인정보['주소'] = 주소
print(개인정보)

삭제할키 = input('삭제할 키: ')
개인정보.pop(삭제할키)
print(개인정보)
```

```
이름: 홍길동
나이: 25
- 키: 이름
- 키: 나이
- 값: 홍길동
- 값: 25
주소: 한국
{'이름': '홍길동', '나이': 25, '주소': '한국'}
```


삭제할 키: 나이
{'이름': '홍길동', '주소': '한국'}

20강. 데이터 처리 맛보기 – 판다스

실습 코딩 실습 23 판다스로 시간표 만들기

문제 아래 학원 시간표를 판다스 DataFrame으로 저장하고, 빈칸을 'x'로 변경한 다음, CSV파일로 저장해 보세요. (CSV 파일은 코드파일의 폴더와 동일한 폴더에 저장됩니다.)

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습23](#)

파일 20강_코딩실습23.py

```
import pandas as pd

# 데이터 생성
월 = pd.Series({'1교시':'국어','2교시':'수학'})
수 = pd.Series({'1교시':'영어','2교시':'영어','3교시':'수학'})
금 = pd.Series({'2교시':'수학','3교시':'국어'})

학원시간표 = pd.DataFrame({
    '월요일':월,
    '수요일':수,
    '금요일':금
})

print('--- 학원시간표 ---')
print(학원시간표)

# 데이터 엑셀로 저장
학원시간표.to_csv('학원시간표.csv')

# 엑셀자료 호출
학원시간표_정리 = pd.read_csv('학원시간표.csv', index_col=0, header=0)

# 데이터 값 변경
학원시간표_정리.loc['3교시']['월요일'] = 'X'
학원시간표_정리.iloc[0][2] = 'X'
```

비법노트: 파이썬 코딩 클래스

```
print ('Wn--- 수정된 학원시간표 ---')
print (학원시간표_정리)

print ('Wn--- 수요일 영어만 있는 시간표 ---')
print (학원시간표_정리[ 학원시간표_정리['수요일'] == '영어'])
```

```
--- 학원시간표 ---
    월요일 수요일 금요일
1교시  국어  영어  NaN
2교시  수학  영어  수학
3교시  NaN  수학  국어

--- 수정된 학원시간표 ---
    월요일 수요일 금요일
1교시  국어  영어  X
2교시  수학  영어  수학
3교시  X  수학  국어

--- 수요일 영어만 있는 시간표 ---
    월요일 수요일 금요일
1교시  국어  영어  X
2교시  수학  영어  수학
```

csv 파일을 열어보면 다음과 같이 콤마로 구분되어 있는 것을 확인 할 수 있습니다.

학원시간표.csv
,월요일,수요일,금요일
1교시,국어,영어,
2교시,수학,영어,수학
3교시,,수학,국어

20장. 도전과제!

그리고 이번에는 csv 파일 대신에 엑셀파일(xls)로 저장 해 보겠습니다.

파일: 20장_도전과제.py

```
import pandas as pd

# 데이터 생성
월 = pd.Series({'1교시':'국어','2교시':'수학'})
수 = pd.Series({'1교시':'영어','2교시':'영어','3교시':'수학'})
금 = pd.Series({'2교시':'수학','3교시':'국어'})

학원시간표 = pd.DataFrame({
    '월요일':월,
    '수요일':수,
    '금요일':금
})

print('--- 학원시간표 ---')
print(학원시간표)

# 데이터 엑셀로 저장
학원시간표.to_excel('학원시간표.xlsx',encoding='utf-8')

# 엑셀자료 호출
학원시간표_정리 = pd.read_excel('학원시간표.xlsx', index_col=0, header=0,
encoding='utf-8')

# 데이터 값 변경
학원시간표_정리.loc['3교시']['월요일'] = 'X'
학원시간표_정리.iloc[0][2] = 'X'

print('Wn--- 수정된 학원시간표 ---')
print(학원시간표_정리)

print('Wn--- 수요일 영어만 있는 시간표 ---')
print(학원시간표_정리[학원시간표_정리['수요일'] == '영어'])
```

비법노트: 파이썬 코딩 클래스

--- 학원시간표 ---

 월요일 수요일 금요일
1교시 국어 영어 NaN
2교시 수학 영어 수학
3교시 NaN 수학 국어

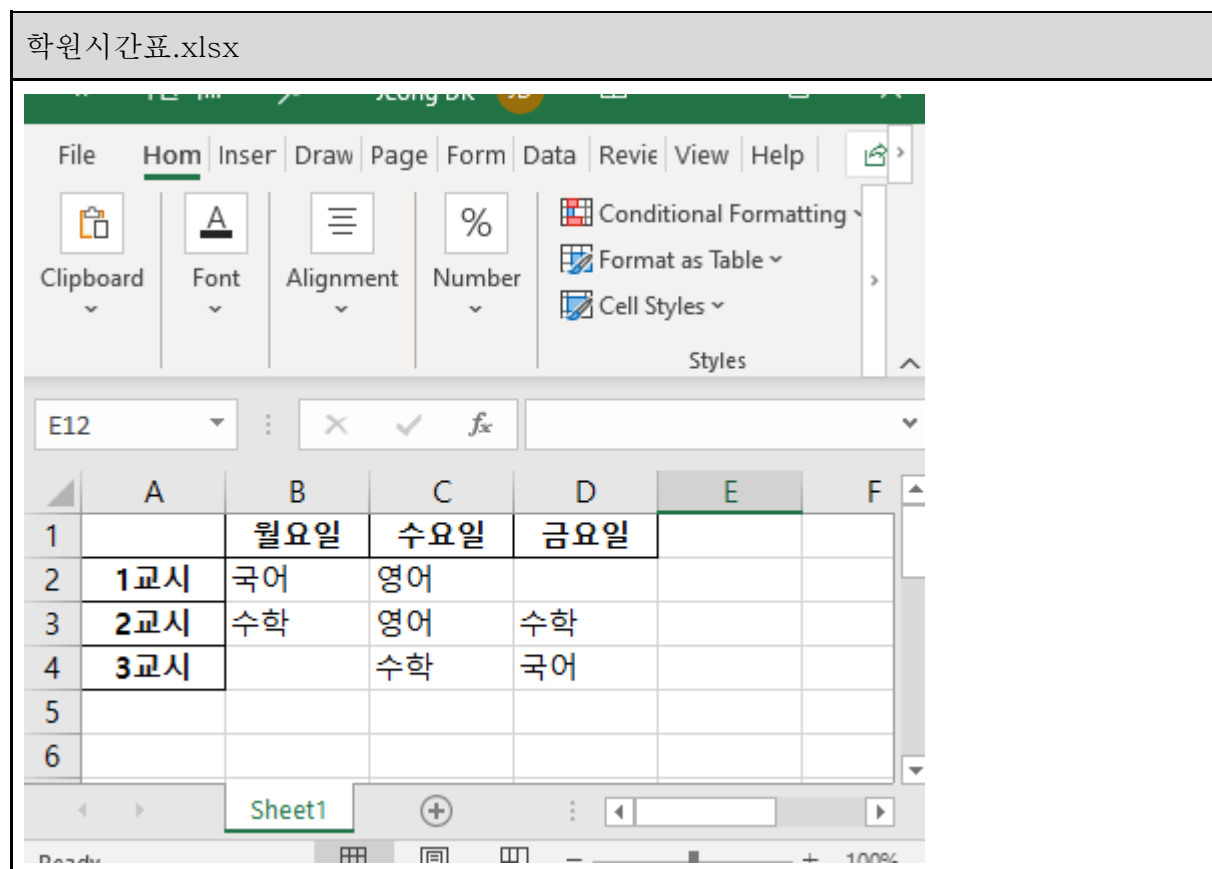
--- 수정된 학원시간표 ---

 월요일 수요일 금요일
1교시 국어 영어 X
2교시 수학 영어 수학
3교시 X 수학 국어

--- 수요일 영어만 있는 시간표 ---

 월요일 수요일 금요일
1교시 국어 영어 X
2교시 수학 영어 수학

엑셀 파일을 열어보면 다음과 같이 엑셀의 기본 시트(sheet1)에 데이터 값이 저장된 것을 확인할 수 있습니다.



21강. 그래픽 코딩 맛보기 – tkinter

실습 코딩 실습 24 목찌빠 게임 만들기

문제 학습한 목찌빠 게임을 조금만 변경해서 직접 만들어 보겠습니다.

해설 [Do It! 첫 코딩 with 파이썬 - 코딩실습24](#)

파일 21강_코딩실습24.py

(1) 틀 만들기

- Mac 환경에서는 윈도우의 모양과 메시지 화면이 조금 다르게 보입니다.

```
import tkinter as tk

게임화면 = tk.Tk()
게임화면.title('목-찌-빠 게임')
게임화면.geometry("250x350")

레이블1 = tk.Label(
    master = 게임화면,
    text = '컴퓨터',
    font = ('Arial', 15))

레이블1.place(
    x = 0,
    y = 0,
    width = 100,
    height = 50)

레이블2 = tk.Label(
    master = 게임화면,
    text = '나',
    font = ('Arial', 15))

레이블2.place(
    x = 150,
```

```
y = 0,  
width = 100,  
height = 50)
```

```
레이블3 = tk.Label(  
    master = 게임화면,  
    text = '-',  
    font = ('Arial', 30),  
    bg = "red",  
    fg = "white")
```

```
레이블3.place(  
    x = 0,  
    y = 50,  
    width = 100,  
    height = 300)
```

```
레이블4 = tk.Label(  
    master = 게임화면,  
    text = 'vs',  
    font = ('Arial', 20))
```

```
레이블4.place(  
    x = 100,  
    y = 50,  
    width = 50,  
    height = 250)
```

```
버튼1 = tk.Button(  
    master = 게임화면,  
    text = '묵',  
    font = ('Arial', 10))
```

```
버튼1.place(  
    x = 150,  
    y = 50,  
    width = 100,  
    height = 100)
```

```
버튼2 = tk.Button(  
    master = 게임화면,
```

비법노트: 파이썬 코딩 클래스

```
text = '찌',
font = ('Arial', 10))

버튼2.place(
    x = 150,
    y = 150,
    width = 100,
    height = 100)

버튼3 = tk.Button(
    master = 게임화면,
    text = '빠',
    font = ('Arial', 10))

버튼3.place(
    x = 150,
    y = 250,
    width = 100,
    height = 100)

게임화면.mainloop()
```

(2) 로직 추가 및 이벤트 처리하기

```
import tkinter as tk

게임화면 = tk.Tk()
게임화면.title('묵-찌-빠 게임')
게임화면.geometry("250x350")

레이블1 = tk.Label(
    master = 게임화면,
    text = '컴퓨터',
    font = ('Arial', 15))

레이블1.place(
    x = 0,
    y = 0,
```



```
width = 100,  
height = 50)
```

```
레이블2 = tk.Label(  
    master = 게임화면,  
    text = '나',  
    font = ('Arial', 15))
```

```
레이블2.place(  
    x = 150,  
    y = 0,  
    width = 100,  
    height = 50)
```

```
레이블3 = tk.Label(  
    master = 게임화면,  
    text = '-',  
    font = ('Arial', 30),  
    bg = "red",  
    fg = "white")
```

```
레이블3.place(  
    x = 0,  
    y = 50,  
    width = 100,  
    height = 300)
```

```
레이블4 = tk.Label(  
    master = 게임화면,  
    text = 'vs',  
    font = ('Arial', 20))
```

```
레이블4.place(  
    x = 100,  
    y = 50,  
    width = 50,  
    height = 250)
```

```
버튼1 = tk.Button(  
    master = 게임화면,  
    text = '폭',
```

```
font = ('Arial', 10),
command = lambda:컴퓨터선택(1))

버튼1.place(
    x = 150,
    y = 50,
    width = 100,
    height = 100)

버튼2 = tk.Button(
    master = 게임화면,
    text = '찌',
    font = ('Arial', 10),
    command = lambda:컴퓨터선택(2))

버튼2.place(
    x = 150,
    y = 150,
    width = 100,
    height = 100)

버튼3 = tk.Button(
    master = 게임화면,
    text = '빠',
    font = ('Arial', 10),
    command = lambda:컴퓨터선택(3))

버튼3.place(
    x = 150,
    y = 250,
    width = 100,
    height = 100)

from random import *
from tkinter import messagebox as mb

def 컴퓨터선택(사용자의선택):
    global 레이블3

    묵찌빠 = {1:'묵', 2:'찌', 3:'빠'}
```

비법노트: 파이썬 코딩 클래스

```
컴퓨터의선택 = randint(1, 3)
```

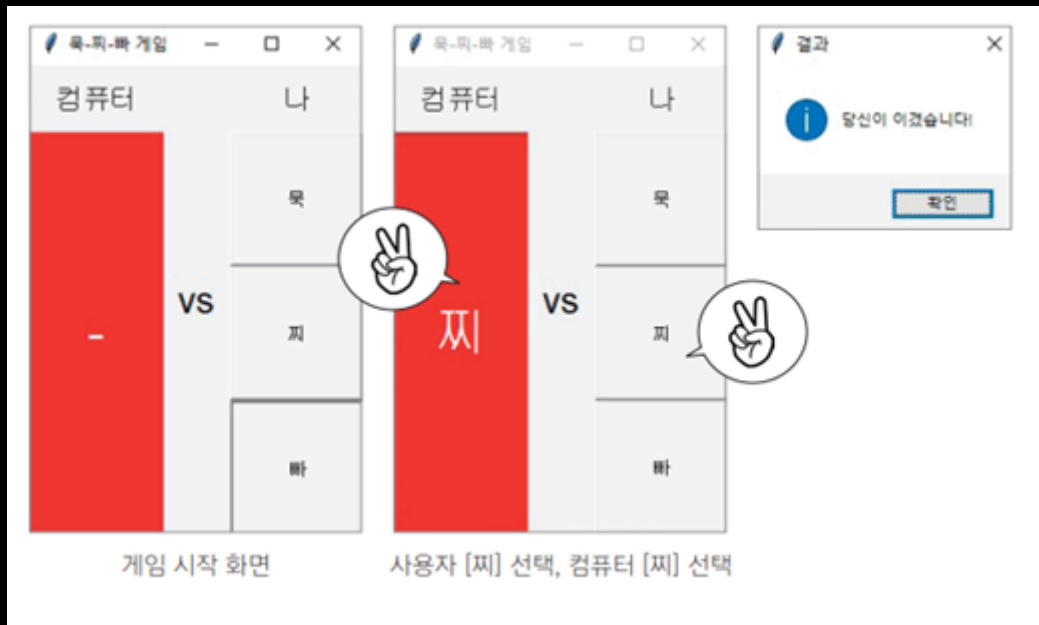
```
레이블3.config(text = 목찌빠[컴퓨터의선택])
```

```
if 컴퓨터의선택 == 사용자의선택:
```

```
    mb.showinfo(title='결과',message = '당신이 이겼습니다!')
```

```
    레이블3.config(text = '-')
```

```
게임화면.mainloop()
```



XX강. -

실습 -

문제 -

해설 -

파일 -.py

(1) -

-

-

해설: -

XX강. -

실습 -

문제 -

해설 -

파일 -.py

(1) -

-

-

해설: -

XX강. -

실습 -

문제 -

해설 -

파일 -.py

(1) -

-

-

해설: -