

## 04 제어문(반복문)

- 주사위를 100번 던져 평균값을 구해야 한다면?
- 1부터 1000까지 더해야 한다면?
- 최근 1년의 주가 데이터를 가져와 분석해야 한다면?

## 04 제어문(반복문)

- $1+2+3+4+ \dots\dots\dots + 1000$

## 04 제어문(반복문\_for)

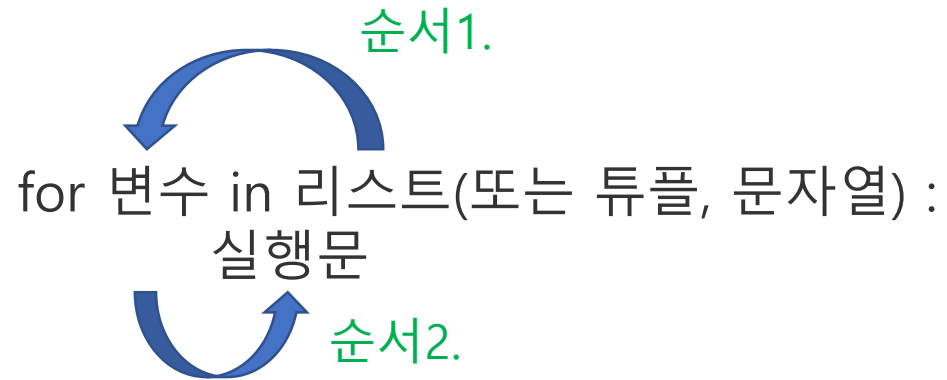
### ■ for문의 기본 구조

✓ for 변수 in 반복할 횟수(리스트, 튜플, 문자열) :

실행문

✓ 반복문이라기 보다, 특정 모음 안의 값들을 하나씩 꺼내서 변수에 담  
는 개념

## 04 제어문(반복문\_for)



### ■ for문의 기본 구조

- ✓ 리스트에 있는 값들을 하나씩 꺼내담을 때 마다 실행문 실행,
- ✓ 그리고 다시 리스트에 다른 값을 꺼낸다.
- ✓ 동작의 반복

## 04 제어문(반복문\_for)

- 다른 언어(java)에서의 for문의 기본 구조
  - ✓ for (int i=0, i<lista.length, i++):  
    lista.get(i)

## 04 제어문(반복문\_for)

### ■ For문 예제를 통한 이해1

✓ `str1 = 'for 반복문 연습'`

`for i in str1 :`

`print(i)`

## 04 제어문(반복문\_for)

### ■ For문 예제를 통한 이해2

✓ `list_cookies = ['새우깡', '홈런볼', '초코파이']`

`for cookie in list_cookies :`

`print(cookie)`

## 04 제어문(반복문\_for)

### ■ for문 예제 01

- ✓ 리스트 변수 list\_num에 'one', 'two', 'three' 값이 담겨있다.
- ✓ 하나씩 꺼내서 print하라.



## 04 제어문(반복문\_for)


### ■ for문 예제 02

- ✓ 리스트 변수 lista에 학생들의 점수(90, 25, 67, 45, 80)가 담겨있다.
- ✓ For문을 이용하여 60점이 이상이라면 "n번째 학생은 합격입니다."를
- ✓ 미만이라면 "불합격입니다."를 출력하는 프로그램을 작성하시오

## 04 제어문(반복문\_for)

### ■ for문의 다양한 형태

✓ for 변수 in 리스트(튜플로 구성된 리스트)



실행문

✓ 튜플 단위로 변수에 담는다.

## 04 제어문(반복문\_for)

- 범위가 감당 범위를 벗어난다면?

- ✓ 지금까지는 작성 가능한 범위였다.

- ✓ 하지만 만번 이상, 백만번 이상 반복하는 프로그램이라면?

- ✓ [1,2,3,4,~ 10000]이라고 작성해야 할까?

## 04 제어문(반복문\_for)

- For문과 사용되는 range 함수

- ✓ range(a, b) << 의미 : a이상 b미만의 리스트
  - ✓ for i in range(a, b) : << a부터 b-1까지의 숫자를 차례로 i에 담는다.
- ✓ [예제]
- ✓ for num in range(1, 101) : << 슬라이싱 범위와 같은 개념
  - ✓ print(num)

## 04 제어문(반복문\_for)

### ■ For문과 사용되는 range 함수 예제

- ✓ for문과 range함수를 사용하여 1부터 100까지 더한 값을 출력하라.

## 04 제어문(반복문\_for)

- 다른 언어(java)에서의 for문 range 구조

- ✓ for (int i=0, i<100, i++):

- print(i)

## 04 제어문(반복문\_for)

### ■ Continue문

- ✓ 특정조건을 만나면 건너뛰고 다음 반복작업으로 넘어가라
- ✓ continue 선언 이후 하위 코딩을 무시하고 for문의 시작으로 돌아가고 싶을 때 사용

## 04 제어문(반복문\_for)

### ■ Break문

- ✓ 조건문과 반복문을 사용중 특정 조건시 반복을 강제 종료 시키고 싶을 때 사용
- ✓ [예시]
- ✓ 10만번 주가 데이터를 가져와라 그 과정 중 어떠한 값을 받아오면 강제로 종료하라



## 04 제어문(반복문\_for)

### ■ For문과 break

- ✓ While문과 사용법 동일
- ✓ 예제)a형인 고객을 선착순으로 이벤트에 당첨시켜주고자 한다.
- ✓ Lista = ['b', 'b', 'ab', 'a', 'b', 'b'] 이렇게 접수한 순서대로

## 04 제어문(반복문\_for)

### ■ 연습문제

- ✓ 1~100 사이에서 3의 배수이면서 2의 배수가 아닌 수를 출력하라
- ✓ 그 합을 역시 출력하라.

[출력 화면]

3 9 15 21 ... 99

누적합 : 867

## 04 제어문(반복문\_for)

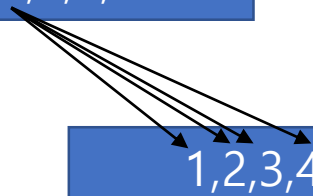
- 2중 For문에 대해 알아보자. 그 이상의 for문도 가능

✓ for a in lista :

1,2,3,4

for b in listb :

1,2,3,4



✓ 두번째 for문은 첫번째 for문의 변수값이 달라질때마다 다시 시작.

## 04 제어문(반복문\_for)

### ■ 이중 For문의 활용

✓ 구구단을 2단부터 9단까지 출력하라. 출력 형태는 아래와 같다

✓  $2 \times 2 = 4$

✓  $2 \times 3 = 6$

✓ ...

✓  $9 \times 9 = 81$

## 04 제어문(반복문\_for)

### ■ For문을 활용하여 리스트에 각종 값을 담기 01

- ✓ 1~100까지의 숫자중 3의 배수를 리스트에 담은 후 해당 리스트를 print 하라.
- ✓ Hint 리스트에 담는다 = append함수 사용

## 04 제어문(반복문\_for)

### ■ For문을 활용하여 리스트에 각종 값을 담기 02

- ✓ 구구단을 2단부터 9단까지 계산하여 모두 리스트에 담은 후
- ✓ 해당 리스트를 print 하라.

## 04 제어문(반복문\_for)

### ■ 연습문제

- ✓ Position = ['과장', '부장', '대리', '사장', '대리', '과장']
- ✓ 위 리스트를 대상으로 중복되지 않은 직위와 직위별 인원 수를 출력하라
- ✓ 중복되지 않는 직위 출력의 순서는 무시해도 된다.

[출력 화면]

중복되지 않은 직급 : ['사장', '부장', '대리', '과장']

각 직급별 인원수 : {'사장': 1, '부장': 1, '대리': 2, '과장': 2}

## 04 제어문(반복문\_for)

- 리스트 내포 포함 시킬까말까 고민...



## 04 제어문(반복문\_while)

### ■ For문과 while문의 차이

- ✓ **For**문은 반복 횟수를 지정

정해진 반복 범위 만큼 실행한다.

- ✓ **While**문은 조건을 지정

조건을 지정해 두고 그 조건이 만족할 때까지 계속 실행한다.

조건을 증가, 감소시키는 문장이 필요

만약 조건을 변경하는 문장을 입력하지 않으면 어떻게 될까?

## 04 제어문(반복문\_while)

### ■ while문의 기본구조

✓ while 조건문 : << True면 반복, False면 종료

실행문

조건을 변화시키는 문장

## 04 제어문(반복문\_while)

### ■ while문 예제를 통한 이해1

✓ `count = 1`

`while count < 5 :`

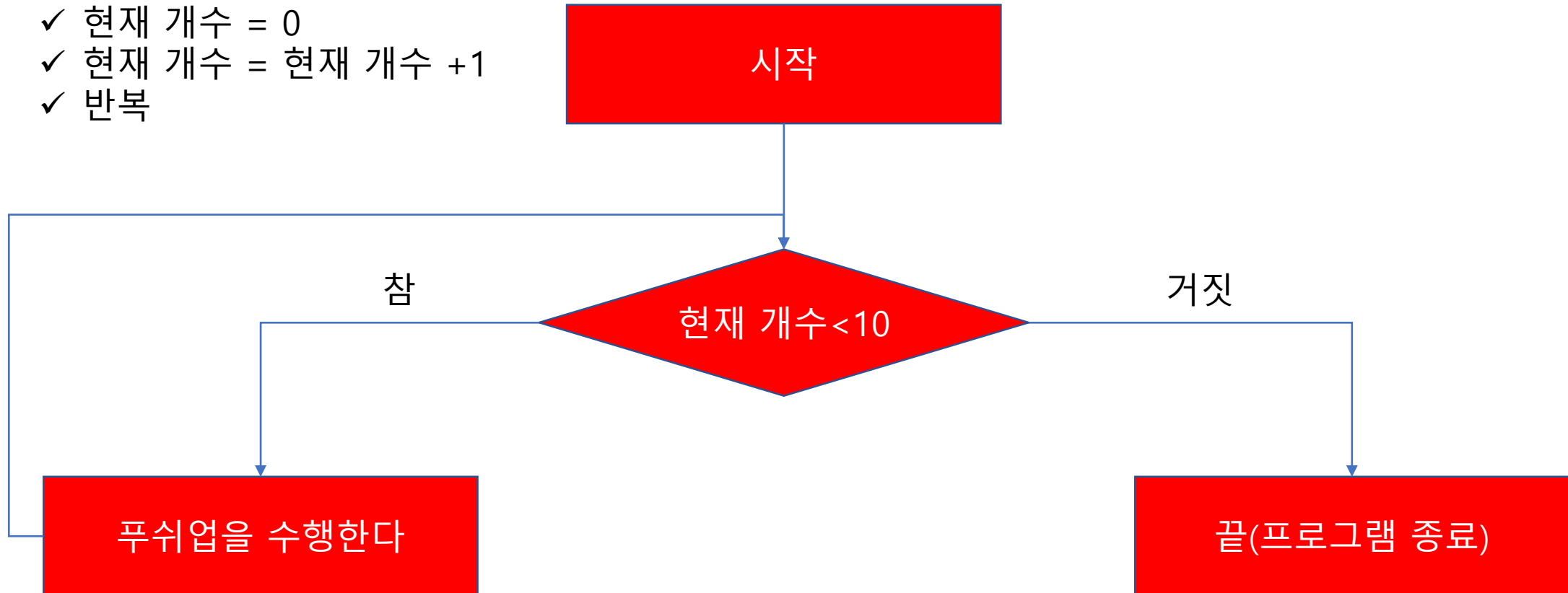
`print(count)`

`count += 1`

## 04 제어문(반복문\_while)

### ■ 푸시업을 10번 한다.

- ✓ 현재 개수 = 0
- ✓ 현재 개수 = 현재 개수 + 1
- ✓ 반복



## 04 제어문(반복문\_while)

### ■ 아래 문구를 print 하시오.

- ✓ 푸쉬업 1회 수행했습니다.
- ✓ 푸쉬업 2회 수행했습니다.
- ✓ 푸쉬업 3회 수행했습니다.
- ✓ ...
- ✓ 푸쉬업 10회 수행했습니다.

## 04 제어문(반복문\_while)

### ■ 아래 문구를 print 하시오.

- ✓ 푸쉬업 1회 수행했습니다.
- ✓ 푸쉬업 2회 수행했습니다.
- ✓ 푸쉬업 3회 수행했습니다.
- ✓ ...
- ✓ 푸쉬업 10회 수행했습니다.
- ✓ 오늘 운동을 끝마칩니다.

## 04 제어문(반복문\_while)

### ■ 변수의 범위

- ✓ A를 0부터 1씩 늘려가며 출력할 때
- ✓ While True:

a=0

print(a)

a = a+1

## 04 제어문(반복문\_while)

### ■ 연습문제

- ✓ While문을 이용하여 1부터 100까지 더하는 프로그램을 작성하라
- ✓ 숫자는 키보드로 입력 받지 않고 `num = 0` 이라 변수 할당 후 시작한다.
- ✓ [결과화면]  
합산 결과는 다음과 같습니다. 5050



## 04 제어문(반복문\_while)

### ■ 연습문제

- ✓ 아래와 같은 결과화면이 나오게 프로그래밍 하여라.
- ✓ 단, 4를 입력하면 프로그램은 종료된다.

✓ [결과화면]

1. Add

2. Del

3. List

4. Quit

Enter Number :

## 04 제어문(반복문\_while)

### ■ 연습문제 : 110p 문제2 A형

- ✓ 리스트의 크기를 키보드로 입력받아 그 크기만큼 임의 숫자를 리스트에 추가하고, 리스트의 크기와 값 전체를 출력하라. 모든 값은 키보드로 입력을 받고, list의 크기는 함수를 통해 구하라. 단, 리스트의 크기는 10 이하로 입력하라.

- ✓ [출력화면 예시]
- ✓ 리스트의 크기를 정하세요!30
- ✓ 리스트의 크기를 10 이하로 다시 할당하세요!9
- ✓ 리스트에 값을 할당해 보세요! 1
- ✓ 리스트에 값을 할당해 보세요! 2

## 04 제어문(반복문\_while)

### ■ Break문

- ✓ 조건문과 반복문을 사용중 특정 조건시 반복을 강제 종료 시키고 싶을 때 사용
- ✓ [예시]
- ✓ 10만번 주가 데이터를 가져와라 그 과정 중 어떠한 값을 받아오면 강제로 종료하라

## 04 제어문(반복문\_while)

### ■ Break문 연습문제 01

- ✓ 1~100까지의 합을 출력하라.
- ✓ 단, while 조건문은 아래 형태이고, break문을 사용하여 프로그래밍 하라.
- ✓ while True:
- ✓ Hint1 -> print(f"1부터 {num}까지의 덧셈 종료")

[결과화면 예시]

1부터 100까지의 덧셈 종료

결과 : 5050

## 04 제어문(반복문\_while)

### ■ Break문 연습문제 02

- ✓ 푸쉬업 예제를 While True: 형태와 break를 사용해서 출력

## 04 제어문(반복문\_while)

### ■ Continue문

- ✓ 특정조건을 만나면 건너뛰고 다음 반복작업으로 넘어가라
- ✓ continue 선언 이후 하위 코딩을 무시하고 While문의 시작으로 돌아가고 싶을 때 사용

## 04 제어문(반복문\_while)

### ■ Continue문 예제 01

- ✓ Continue를 사용해서 1~100 중 홀수만 출력하라

## 04 제어문(반복문\_while)

### ■ Continue문 예제 02

- ✓ 푸쉬업 예제를 break와 continue 모두를 사용해서 출력



## 04 제어문(반복문\_while)

### ■ 연습문제 1

✓ 1~10 사이의 난수 하나를 발생시켜 변수에 담고, 예상되는 숫자를 입력하여 맞추는 프로그램을 작성하라. 값이 일치하면 '성공', 입력한 값이 난수보다 크면 '더 작은 값을 입력하세요' 난수보다 작으면 '더 큰 값을 입력하세요' 라고 출력하라.

✓ [hint]

✓ import random

✓ num\_random = random.randint(1,10) # 1~10 사이의 난수 발생

## 04 제어문(반복문\_while)

### ■ 연습문제 2

- 아래의 문자열 객체를 이용하여 단어를 추출하고 단어의 개수를 출력하라.

✓ 비 갠 뒤에 비애 대신 a happy end

비스듬히 씹 비웃듯 칠색 무늬의 무지개

철없이 철 지나 철들지 못해 (still)

[화면출력 예시]

비

갠

...

못해

(still)

단어 개수 : 19

## 04\_별첨 알고리즘

- 어떤 문제를 해결하기 위한 일련의 절차.
- 흔히 말하는 프로그램의 로직(Logic)

## 04\_별첨 알고리즘(최대/최소값)

### ■ 최대값 최소값을 구하라.

✓ 1~100의 정수중 10개를 뽑아 리스트로 담고 최대값과 최소값을 구하라

✓ [hint]

✓ 1. `num_random = random.randint(1,100)` # 1~100 사이의 난수 발생

✓ 2.

✓ 3.

[출력화면 예시]

[57, 9, 27, 34, 58, 79, 61, 64, 15, 98]

최대값 = 98, 최소값 = 9

## 04\_별첨 알고리즘(선택정렬[selection sort])

- 선택정렬 알고리즘은 특정 요소(값)을 기준으로 나머지 모든 원소를 비교해가며 정렬하는 방법이다.

[출력화면 예시]

초기 리스트 : [3, 5, 1, 2, 4]

1번째 정렬 : [1, 5, 3, 2, 4]

2번째 정렬 : [1, 2, 5, 3, 4]

3번째 정렬 : [1, 2, 3, 5, 4]

4번째 정렬 : [1, 2, 3, 4, 5]

