

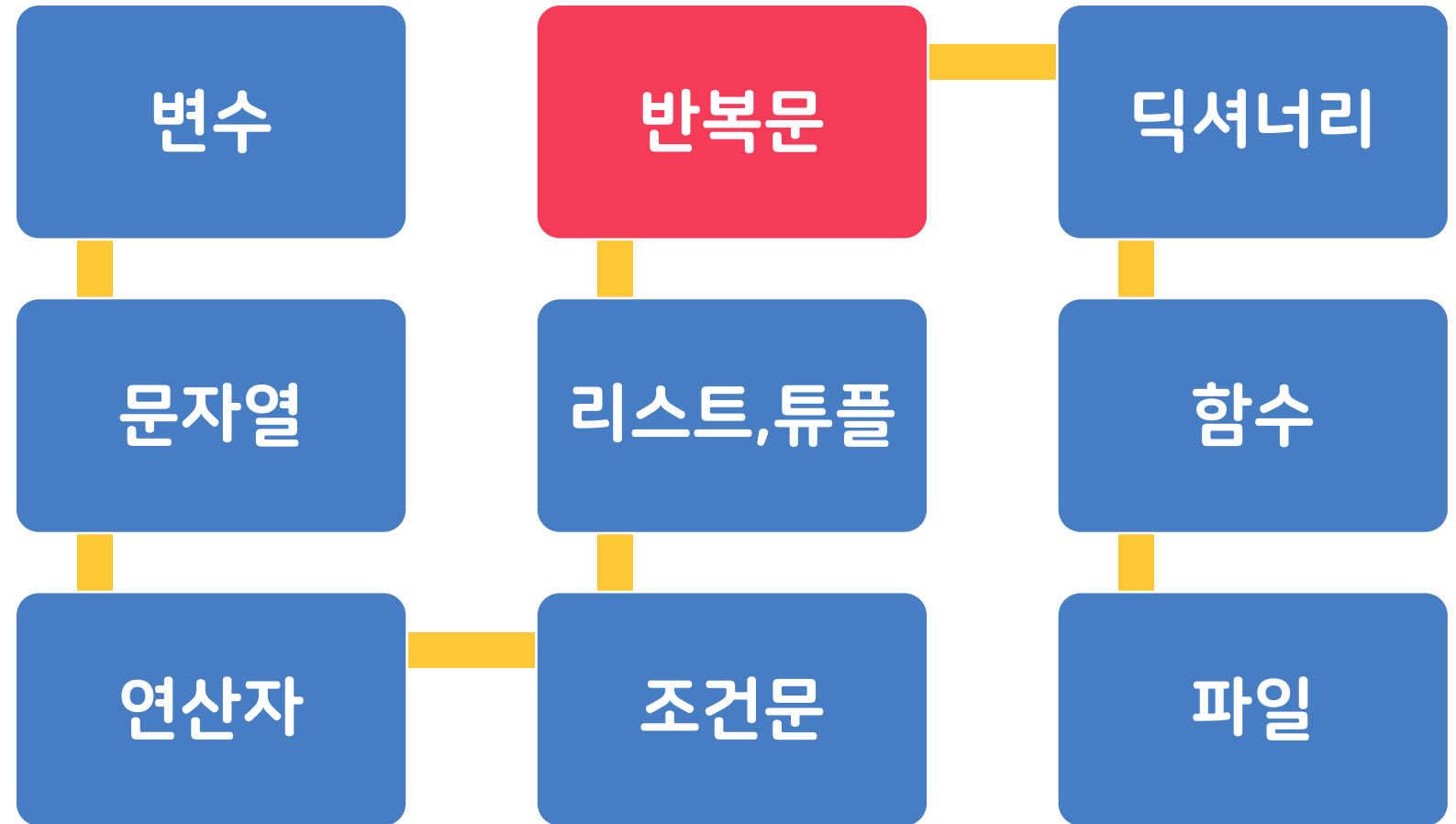


스마트인재개발원
Smart Human Resources Development

손 지 영 강사



수업 진행방향





학습목표

- 반복문에 대해 알 수 있다.
- 반복문 종류와 특징을 알 수 있다.
- 반복문으로 예제 문제를 풀 수 있다.

python 반복문 이란?

- 프로그램 내에서 똑같은 명령을 일정 횟수만큼 **반복하여 수행**하도록 제어하는 명령문
- 반복문 종류는 **while**문, **for**문이 있다.



while : 반복 횟수가 명확하지 않을 때

for : 반복 횟수가 명확할 때

문자열 또는 리스트 또는 튜플이 들어갔을 때 안에 있는 요소를 **하나씩 반복**

```
for 변수 in 문자열(or 리스트 or 튜플):  
    print(변수)
```

들여쓰기 (Tab, Space*4)

(colon, 콜론)

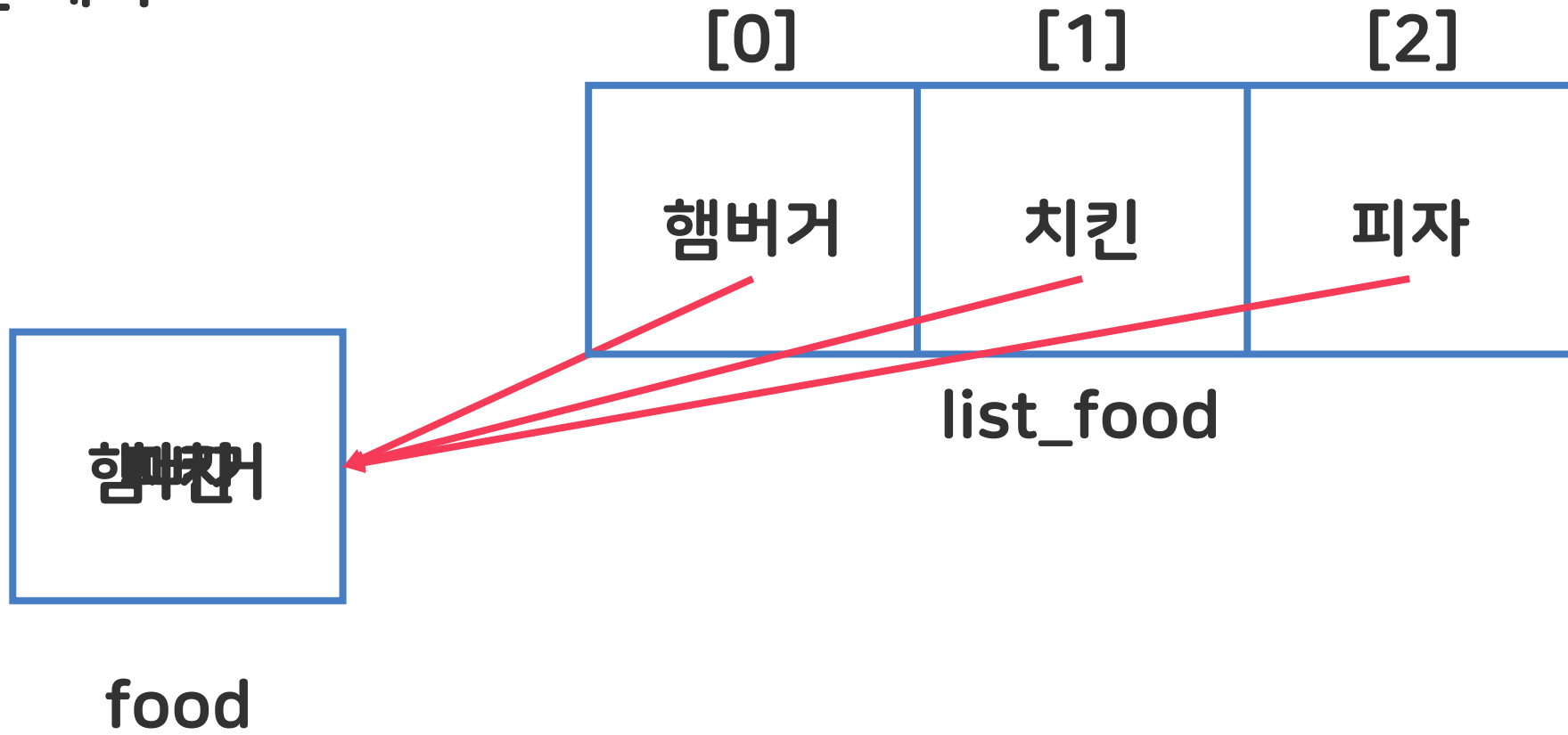
for문 예시

```
list_food = ["햄버거", "치킨", "피자"]  
for food in list_food:  
    print(food)
```



The diagram illustrates the execution of the for loop. Red boxes highlight the list elements ("햄버거", "치킨", "피자") and the loop variable (food). Red arrows show the iteration process. Red circles with numbers 1 through 7 indicate the sequence of operations: 1. List creation, 2. Loop start, 3. First iteration (print), 4. Second iteration (print), 5. Third iteration (print), 6. Loop end, 7. Final print statement.

for문 예시



for문 예시

```
hi = "안녕하세요"  
for s in hi:  
    print(s)
```

```
tuple_food = ("햄버거", "치킨", "피자")  
for food in tuple_food:  
    print(food)
```


1. 5명에 대한 정보처리기사 자격증 시험 점수가 리스트에 담겨있습니다.
이때 각 점수가 합격 점수인지 불합격 점수인지 판별하여 출력하시오.
(60점 이상 합격)

```
score_list = [90, 45, 70, 60, 55]  
?
```

```
1번 학생은 합격입니다.  
2번 학생은 불합격입니다.  
3번 학생은 합격입니다.  
4번 학생은 합격입니다.  
5번 학생은 불합격입니다.
```

range() 함수 사용

- 필요한 만큼의 숫자를 만들어내는 유용한 기능
- range(시작할 숫자, 종료할 숫자, 증가량)
- range(1, 10, 1) → 1부터 9까지 1씩 증가
- range(1, 100, 3) → 1부터 99까지 3씩 증가
- range(10, 1, -1) → 10부터 2까지 1씩 감소(-1씩 증가)

range() 함수 사용

```
for i in range(1, 10, 1):  
    print(i)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

1
2
3
4
5
6
7
8
9

print() 함수

- end 속성

```
for i in range(1, 10, 1):  
    print(i, end=" ")
```

1 2 3 4 5 6 7 8 9

```
for i in range(1, 10, 1):  
    print(i, end="\n")
```

1
2
3
4
5
6
7
8
9

range() 함수 사용

- 필요한 만큼의 숫자를 만들어내는 유용한 기능
- range(**기본값 0**, 종료할 숫자, **기본값 1**)
- range(3, 10) → 3부터 9까지 1씩 증가
- range(10) → 0부터 9까지 1씩 증가

```
for i in range(3, 10):  
    print(i, end=" ")
```

3 4 5 6 7 8 9

```
for i in range(10):  
    print(i, end=" ")
```

0 1 2 3 4 5 6 7 8 9

range(len(list)) 함수 사용

```
1 list1 = ["햄버거", "피자", "핫도그"]
```

```
1 for i in range(0, len(list1), 1):  
2     print(i, list1[i])
```

0 햄버거
1 피자
2 핫도그

```
1 for i in range(len(list1)):  
2     print(i, list1[i])
```

0 햄버거
1 피자
2 핫도그

2. 5명에 대한 **정보처리기사 자격증 시험 점수**가 리스트에 담겨있습니다.
이때 각 점수가 **합격 점수**인지 **불합격 점수**인지 판별하여 출력하시오.
(60점 이상 합격)

```
score_list = [90, 45, 70, 60, 55]  
?
```

```
1번 학생은 합격입니다.  
2번 학생은 불합격입니다.  
3번 학생은 합격입니다.  
4번 학생은 합격입니다.  
5번 학생은 불합격입니다.
```

1. for문을 이용하여 97부터 77까지 출력하시오.

?

97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77

2. for문을 이용하여 23부터 39까지 출력하시오

?

23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39


```
list1 = [[1,2],[3,4],[5,6]]  
for i,j in list1:  
    print(i, j)
```

```
1 2  
3 4  
5 6
```

```
list1 = [[1,2],[3,4],[5,6]]  
a,b,c = list1
```

```
print(a)  
a1,a2 = a  
print(a1, a2)
```

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

For문을 이용하여 다음 list에 들어있는 요소 중
가장 큰 수를 찾아 출력하세요.

```
list2 = [4, 5, 2, 1, 99, 15, 2, 7, 27]
```

```
?
```

```
99
```

For문을 이용하여 다음 list에 들어있는 요소 중
가장 작은 수를 찾아 출력하세요.

```
list2 = [4, 5, 2, 1, 99, 15, 2, 7, 27]  
?
```

1

1. 1부터 100사이의 숫자 중 3의 배수인 값들의 합을 출력하세요.

정답 : 1683

2. for문을 이용하여 구구단 2단을 출력하시오.

2	*	1	=	2
2	*	2	=	4
2	*	3	=	6
2	*	4	=	8
2	*	5	=	10
2	*	6	=	12
2	*	7	=	14
2	*	8	=	16
2	*	9	=	18

숫자를 입력 받고 입력 받은 숫자의 **약수**를 구하시오.
(약수란 어떤 수를 나누어 떨어지게 하는 수)

```
num = int(input("정수 입력 >> "))  
?
```

```
정수 입력 >> 32  
32의 약수 : 1 2 4 8 16 32
```

다음 결과화면 같이 동작하는 프로그램을 for문을 활용하여 작성하시오.

```
=====SJY Kitchen=====
요리 3가지 입력해보기
1 번째 요리는? 탕수육
2 번째 요리는? 가지튀김
3 번째 요리는? 짬뽕
주문하신 메뉴는 탕수육, 가지튀김, 짬뽕 입니다.
```

```
print("=" * 20 + "SJY Kitchen" + "=" * 20)
print("요리 3가지 입력해보기")

menus = []

# 요리 3가지만 주문할 수 있어요. 입력받기 위해서 for문 사용!
for i in range(3):
    # [Blank box for input]

# 주문 메뉴 출력도 menus 안에 있는 내용을 for문 돌리며 출력
print(f"주문하신 메뉴는", end=' ')
for i in range(3):
    # [Blank box for output]

print("입니다.")
```

요구사항정의에 따라 메뉴 주문 프로그램을 구현해보자.

1. 어서오세요~ chef's kitchen 입니다. (초기 실행 당시 1번만 띄워주기)
2. [1]메뉴선택 [2]메뉴삭제 [3]주문완료(종료) 메인 안내 화면 출력하기
3. [1]메뉴선택
 - 메뉴판 음식 리스트 띄워주기 (["Burger", "Pizza", "Pasta", "Salad", "Sushi"])
 - 선택할 메뉴 개수를 입력받고 그만큼 메뉴를 입력받아 장바구니 list에 저장하기
 - 장바구니 list 내용 띄워주기
 - 위 내용이 끝나면 다시 [1]메뉴선택 [2]주문변경 [3]주문요청(종료) 선택항목 띄워주기
 - 단, [1]메뉴선택을 다시 눌러서 메뉴를 더 담을 경우 기존의 입력된 메뉴는 그대로 유지되면서 추가되어야 함
4. [2]메뉴삭제
 - 수정은 불가 삭제만 가능
 - 삭제할 메뉴이름을 입력받아 장바구니 list에서 삭제해주면 됨
 - 삭제하고 장바구니 list내용을 띄워주기
 - 위 내용이 끝나면 다시 [1]메뉴선택 [2]주문변경 [3]주문완료-종료 선택항목 띄워주기
5. [3]주문요청(종료)
 - 주문요청을 완료할 경우 사용자로부터 'q'를 입력받으면 "주문요청 완료!" 출력한 후 반복문 종료

요구사항정의에 따라 메뉴 주문 프로그램을 구현해보자.

[실행 결과]

```
어서오세요~ chef's kitchen 입니다.

[1]메뉴선택 [2]메뉴삭제 [3]주문완료(종료)
선택하세요: 1

메뉴판
Burger
Pizza
Pasta
Salad
Sushi
선택할 메뉴 개수를 입력하세요: 2
메뉴를 입력하세요: Burger
메뉴를 입력하세요: Pasta

장바구니: ['Burger', 'Pasta']

[1]메뉴선택 [2]메뉴삭제 [3]주문완료(종료)
선택하세요: 2
삭제할 메뉴를 입력하세요: Burger
Burger이(가) 장바구니에서 삭제되었습니다.

장바구니: ['Pasta']

[1]메뉴선택 [2]메뉴삭제 [3]주문완료(종료)
선택하세요: 1

메뉴판
Burger
Pizza
Pasta
Salad
Sushi
선택할 메뉴 개수를 입력하세요: 1
메뉴를 입력하세요: Sushi

장바구니: ['Pasta', 'Sushi']

[1]메뉴선택 [2]메뉴삭제 [3]주문완료(종료)
선택하세요: 3
주문을 완료하시겠습니까? 완료하려면 'q'를 입력하세요: q
주문요청 완료!
```




다음시간에는?

딕셔너리(dictionary)
