



주문식교육의 산실
영진전문대학교

구조적 프로그래밍 구성 요소 (흐름제어문)

일본IT과

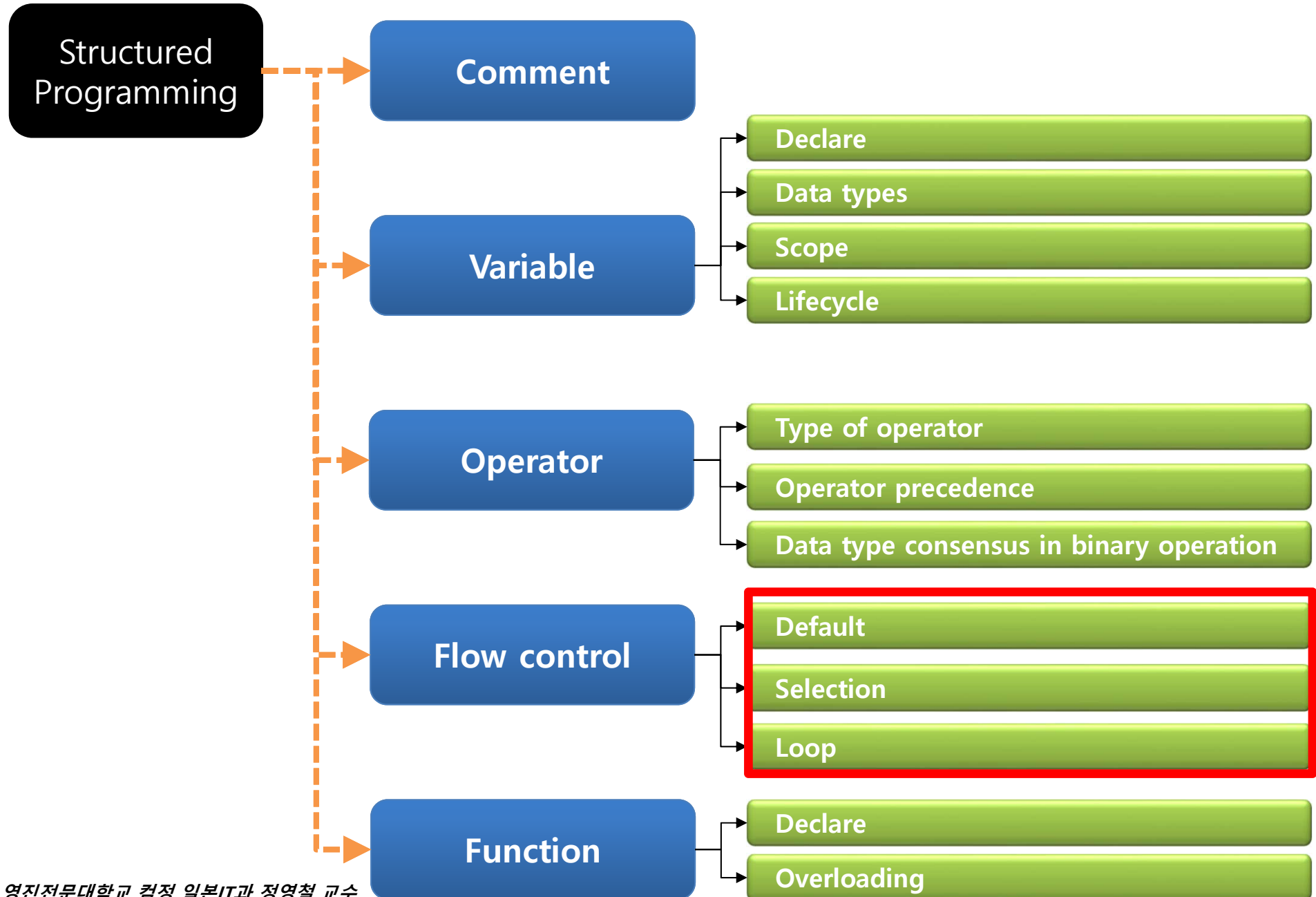
정영철 교수



영진전문대학교 컴퓨터정보계열


SCHOOL OF COMPUTER INFORMATION

구조적 프로그래밍 구성 요소



흐름제어 (Flow Control) 이란?

- 프로그램 코드가 실행되는 순서(흐름: Flow)를 제어하기 위해 사용



```
1 print("hello")
2
3 print("안녕하세요")
4
5 value = True
6
7 # value 가 "참" 이면 2단 구구단 출력
8 if value :
9     for value in range(1, 10):
10         print("2 X ", value, " = ", 2*value)
11 # value 가 "거짓" 이면 3단 구구단 출력
12 else:
13     iCount = 1
14     while iCount <= 9:
15         print("3 X ", iCount, " = ", 3*iCount)
16         iCount += 1
```

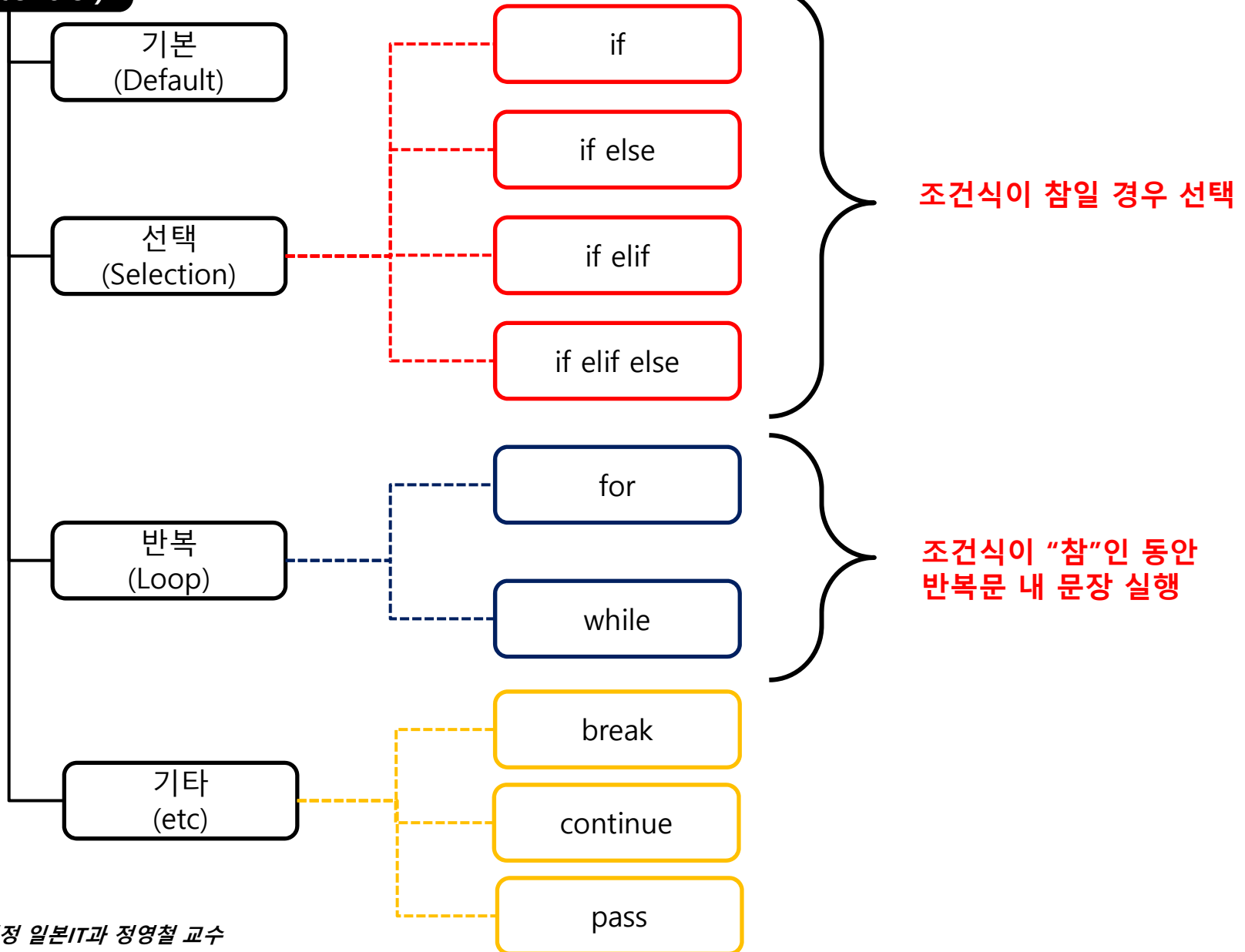
1. 코드는 위에서 아래로
순차적으로 실행 된다 (기본)

2. 조건식에 따라 선택과 반복

기본

흐름제어문 종류

흐름제어문 (Flow control)

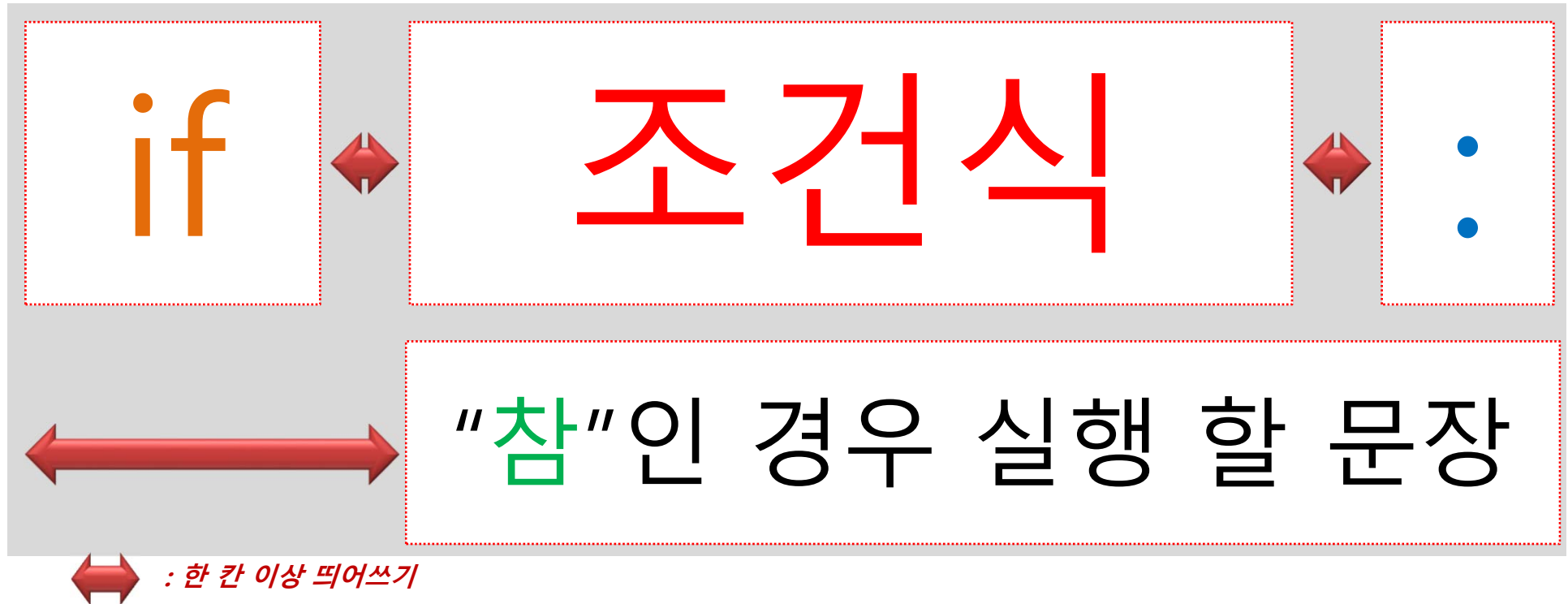




Selection (선택) 문

선택 (Selection) 흐름 제어문 : if 문 (1)

✓ if 문 원형



```
1  if 3 > 2 :  
2      print("3은 2보다 큼니다.")
```

사용 예) 선택 경우의 수 : 1개
 선택 결과 : 0개 또는 1개

선택 (Selection) 흐름 제어문 : if 문 (2)

```
1 print("1")
2
3 # 조건식이 "참"인 경우만 실행 될 경우
4 # 단일 if 문 사용
5 ~ if True :
6     print("T 1-1") # "참" 인 경우 실행될 문장이 2라인 이상인 경우
7     print("T 1-2") # "참" 인 경우 실행될 문장들의 띄어쓰기를 반드시 일치
8
9 print("3")
```

출력 결과는? 이유는?

선택 (Selection) 흐름 제어문 : if 문 (3)

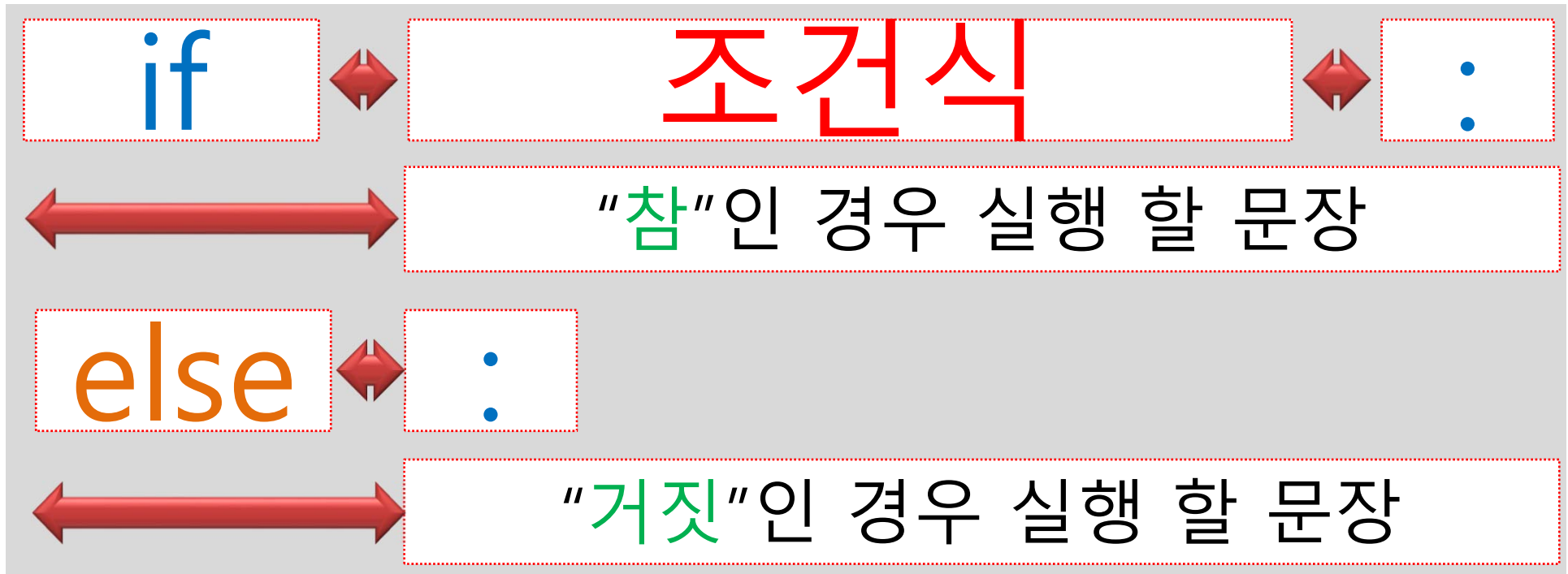
- 키보드로부터 정수를 입력 받아 입력 값의 홀수, 짝수를 구분하여, 짝수일 경우만 "짝수 입니다" 문자열을 화면에 출력하는 프로그램 작성하라

✓ 실행 결과

- 첫 번째 실행 결과 : 정수를 입력하세요1
- 두 번째 실행 결과 : 정수를 입력하세요2
짝수 입니다

선택 (Selection) 흐름 제어문 : if – else 문 (1)

✓ if – else 문 원형



↔ : 한 칸 이상 띄어쓰기

```
1 ~ if 0.3 // 3 != 0 :  
2 |     print("몫이 0이 아닙니다.")  
3 ~ else :  
4 |     print("몫이 0입니다.")
```

사용 예) 선택 경우의 수 : 2개
 선택 결과 : 1개

선택 (Selection) 흐름 제어문 : if – else 문 (2)

```
1  # 키보드로부터 문자열 입력
2  inputValue = input("정수를 입력하세요")
3  inputValue = int(inputValue) # 문자형 -> 정수형 형 변환
4
5  # inputValue 나머지 값이 0일 경우 "짝수"
6  # 아닐 경우 홀수
7  if inputValue % 2 == 0 :
8      print("짝수 입니다") # 짝수일 경우 화면에 문자열 출력
9  else:
10     print("홀수 입니다") # 짝수가 아니라면 "홀수"!!
```

선택 (Selection) 흐름 제어문 : if-else 문 (3)

- 키보드로부터 정수를 입력 받고
- 0 또는 양수이면 "0 또는 양수"를 문자열로 출력
- 음수 이면 "음수" 문자열을 화면에 출력하는 프로그램 작성하라

✓ 실행 결과

- 첫 번째 실행 결과 : 정수를 입력 하세요 **1**
0 또는 양수 입니다.
- 두 번째 실행 결과 : 정수를 입력 하세요 **-1**
음수 입니다.
- 세 번째 실행 결과 : 정수를 입력 하세요 **0**
0 또는 양수 입니다.

선택 (Selection) 흐름 제어문 : if – elif 문 (1)

✓ if – elif 문 원형



↔ : 한 칸 이상 띄어쓰기

```
1 value = 20
2 ~ if value >= 30 :
3     print("30 이상")
4 ~ elif value >= 20 :
5     print("20 이상")
```

사용 예) 선택 경우의 수 : 2개 이상
 선택 결과 : 0개 또는 1개

선택 (Selection) 흐름 제어문 : if – elif 문 (2)

```
1  # 키보드로부터 정수를 입력 받아
2  # 아래 성적으로 환산하는 프로그램을 작성하라
3  # 90 <= 점수      이면 "A"
4  # 80 <= 점수 < 90 이면 "B"
5  # 70 <= 점수 < 80 이면 "C"
6  # 60 <= 점수 < 70 이면 "D"
7
8  inputValue = input("정수를 입력하세요")
9  inputValue = int(inputValue) # 문자형 -> 정수형 형 변환
10
11  if inputValue >= 90 :
12      print("A")
13  elif inputValue >= 80 :
14      print("B")
15  elif inputValue >= 70 :
16      print("C")
17  elif inputValue >= 60 :
18      print("D")
19
20
영진전문 21  print("프로그램 종료")
```

선택 (Selection) 흐름 제어문 : if-elif 문 (3)

- 키보드로부터 정수를 입력 받고
 - 0 이면 "0" 를 문자열로 출력
 - 양수이면 "양수"를 문자열로 출력
 - 음수 이면 "음수" 문자열을 화면에 출력하는 프로그램 작성하라

✓ 실행 결과

- 첫 번째 실행 결과 : 정수를 입력 하세요 1
양수 입니다.
- 두 번째 실행 결과 : 정수를 입력 하세요 -1
음수 입니다.
- 세 번째 실행 결과 : 정수를 입력 하세요 0
0 입니다.

선택 (Selection) 흐름 제어문 : if – elif - else 문 (1)

✓ if – elif - else 문 원형

↔ : 한 칸 이상 띄어쓰기



```
1 value = 5
2 if value >= 30 :
3     print("30 이상")
4 elif value >= 20 :
5     print("20 이상")
6 else:
7     print("20 미만")
```

사용 예) 선택 경우의 수 : 2개 이상
 선택 결과 : 1개

선택 (Selection) 흐름 제어문 : if – elif - else 문 (2)

```
1  # 키보드로부터 정수를 입력 받아
2  # 아래 성적으로 환산하는 프로그램을 작성하라
3  # 90 <= 점수      이면 "A"
4  # 80 <= 점수 < 90 이면 "B"
5  # 70 <= 점수 < 80 이면 "C"
6  # 60 <= 점수 < 70 이면 "D"
7  # 아니면 "F"
8
9  inputValue = input("정수를 입력하세요")
10 inputValue = int(inputValue) # 문자형 -> 정수형 형 변환
11
12 ~ if inputValue >= 90 :
13     print("A")
14 ~ elif inputValue >= 80 :
15     print("B")
16 ~ elif inputValue >= 70 :
17     print("C")
18 ~ elif inputValue >= 60 :
19     print("D")
20 ~ else:
21     print("F")
22
23
영진전문 24 print("프로그램 종료")
```


선택 (Selection) 흐름 제어문 : if-elif-else 문 (3)

- 키보드로부터 영문 문자열을 입력 받아, 아래 테이블 규칙을 따라
- 한글로 변환하는 프로그램을 작성하라.
- 단, 아래 테이블 이외의 영문 이름이 입력 될 경우 "그 외" 문자열 출력

입력	출력	입력	출력	입력	출력
SAMSUNG	삼성	LG	엘지	KAKAO	카카오
NAVER	네이버	HYUNDAI	현대	SK	에스케이

회사명을 입력하세요SK
에스케이

회사명을 입력하세요SAMSUNG
삼성

회사명을 입력하세요HYOSUNG
그 외



LOOP (반복) 문

반복 (Loop) 문

- 조건식이 "참" 인 동안 명령어를 실행하기 위해 사용
- Python에서는 아래 두 가지 종류 반복문 제공
 - **for**
 - 정해진 횟수의 반복을 실행할 경우 주로 사용
 - **while**
 - for와 동일, 하지만 주로 횟수가 정해지지 않은 반복문 작성 시 사용

```
1 ~ for value in [1, 2, 3]:  
2     |     print(value)
```



결과 값 : 1, 2, 3

```
3  
4 value = 1  
5 ~ while value <= 3:  
6     |     print(value)  
7     |     value = value + 1
```



결과 값 : 1, 2, 3

반복 (Loop) 흐름 제어문 : for 문 (1)

✓ for 문 원형



↔ : 한 칸 이상 띄어쓰기

- 동작 절차
 1. 반복 횟수 : 리스트 내 원소의 개수
 2. 특이사항
 - ✓ 반복 시 리스트 내 원소를 획득하여 변수에 저장

```
1 # 리스트 [5, 3, 1]의 원소 개수 : 3
2 # 따라서 아래 For문은 3번 반복
3 # 매 반복 시 리스트 원소값 value 변수에 저장
4 ~ for value in [5, 3, 1]:
5     print(value)
6
7 # 실행 결과 : 5, 3, 1
```

반복 (Loop) 흐름 제어문 : **range()** 함수

```
1 ~ for value in range(2):
2     |     print(value)  # 0, 1
3
4 ~ for value in range(3):
5     |     print(value)  # 0, 1, 2
6
7 ~ for value in range(1, 2):
8     |     print(value)  # 1
9
10 ~ for value in range(1, 3):
11     |     print(value)  # 1, 2
12
13 ~ for value in range(1, 11, 3):
14     |     print(value)  # 1, 4, 7, 10
15
16 ~ for value in range(10, 0, -3):
17     |     print(value)  # 10, 7, 4, 1
```

- **for문에서 지정된 반복 횟수를 설정하기 위해 range()함수 사용**

- **range (arg1)**
 - 0부터 **arg1 - 1** 까지, 1씩 증가하여 리스트 생성
 - 예) range(4) : 0, 1, 2, 3
- **range (arg1, arg2)**
 - **arg1**부터 **arg2 - 1** 까지, 1씩 증가하여 리스트 생성
 - 예) range(5, 9) : 5, 6, 7, 8
- **range (arg1, arg2, arg3)**
 - **arg1**부터 **arg2 - 1** 까지, 정수 **arg3 크기만큼** 증가하여 리스트 생성
 - 예) range(5, 10, 2) : 5, 7, 9

반복 (Loop) 흐름 제어문 : **for** 문 (2)

```
1  # 3, 6, 9 게임 구현
2  for index in range(1, 60):
3      value    = str(index)    # 현 숫자(정수)를 문자열로 변환
4      flag     = False        # 현 숫자 내 3, 6, 9 숫자가 있는지 나타내는 플래그
5      msg      = ""
6
7      # 문자열 개수 만큼 순회
8      # 예) "34" -> 2번 순회, 첫 번째 "3", 두 번째 "4"
9      for index_char in value :
10         # 현 문자가 3, 6, 9 중에 하나일 경우 "박수" 출력
11         if index_char == "3" or index_char == "6" or index_char == "9":
12             msg += "박수 "
13             flag = True      # 현 숫자 내 3, 6, 9가 존재 함으로 플래그 ON
14
15         if flag: # 3, 6, 9 중에 하나
16             print(msg)    # 박수 출력
17         else:    # 3, 6, 9 가 아닐 경우
18             print(index)  # 숫자 출력
```

1, 2, "박수", 4, 5, "박수", 7, 8, "박수", 10 ... 28, "박수", "박수", "박수", "박수", "박수 박수"

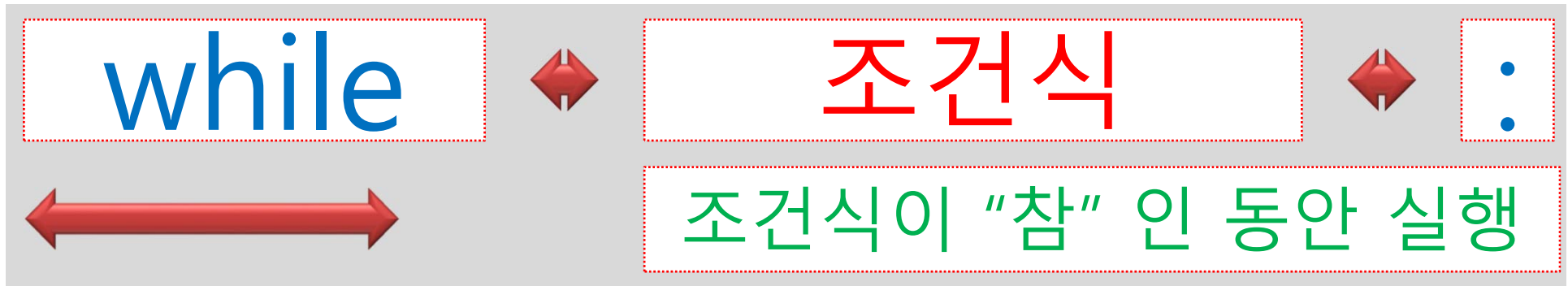
반복 (Loop) 흐름 제어문 : **for** 문 (3)

- 아래와 같이 출력되는 프로그램을 for 문을 이용해 작성하라.

2 X 1 = 2	2 X 2 = 4	2 X 3 = 6
2 X 4 = 8	2 X 5 = 10	2 X 6 = 12
2 X 7 = 14	2 X 8 = 16	2 X 9 = 18
4 X 1 = 4	4 X 2 = 8	4 X 3 = 12
4 X 4 = 16	4 X 5 = 20	4 X 6 = 24
4 X 7 = 28	4 X 8 = 32	4 X 9 = 36
6 X 1 = 6	6 X 2 = 12	6 X 3 = 18
6 X 4 = 24	6 X 5 = 30	6 X 6 = 36
6 X 7 = 42	6 X 8 = 48	6 X 9 = 54
8 X 1 = 8	8 X 2 = 16	8 X 3 = 24
8 X 4 = 32	8 X 5 = 40	8 X 6 = 48
8 X 7 = 56	8 X 8 = 64	8 X 9 = 72

반복 (Loop) 흐름 제어문 : **while** 문 (1)

✓ while 문 원형



 : 한 칸 이상 띄어쓰기

■ 동작 절차

1. while 의 조건식이 "참"인 동안 while 블록 내 코드 실행
2. 일반적으로 반복 횟수가 일정하게 정해지지 않을 경우 사용

```
1 flag = True
2 ~ while flag :
3     value = int(input("양의 정수를 입력하세요"))
4
5 ~     if value > 0:
6         print("입력 값 : ", value)
7 ~     else:
8         flag = False
```


반복 (Loop) 흐름 제어문 : **while** 문 (2)

```
1 ~ def menu_print():
2     print("-----")
3     print("1. 2단 구구단 출력")
4     print("2. 4단 구구단 출력")
5     print("3. 프로그램 종료")
6     print("-----")
7
8     flag = True
9 ~ while flag :
10     menu_print();
11
12     inputValue = int(input("메뉴를 선택 해주세요"))
13
14 ~     if inputValue == 1:
15 ~         for num in range(1, 10):
16             print("2 X ", num, " = ", 2*num)
17 ~     elif inputValue == 2:
18         num = 1
19 ~         while num <= 9:
20             print("4 X ", num, " = ", 4*num)
21             num = num + 1
22 ~     elif inputValue == 3:
23         flag = False
24 ~     else:
25         print("1~3 사이의 값을 입력 해주세요")
```



Break 문

Break 문 : (1)

- 반복문(Loop) 내 사용
- 현 반복문을 탈출하는 용도로 사용

```
3 # Loop From 0 To 9
4 for value in range(10):
5     print(value)
6
7     # value 값이 5일 경우
8     if value == 5:
9         break # Loop문 탈출
10
11 print("End")
```

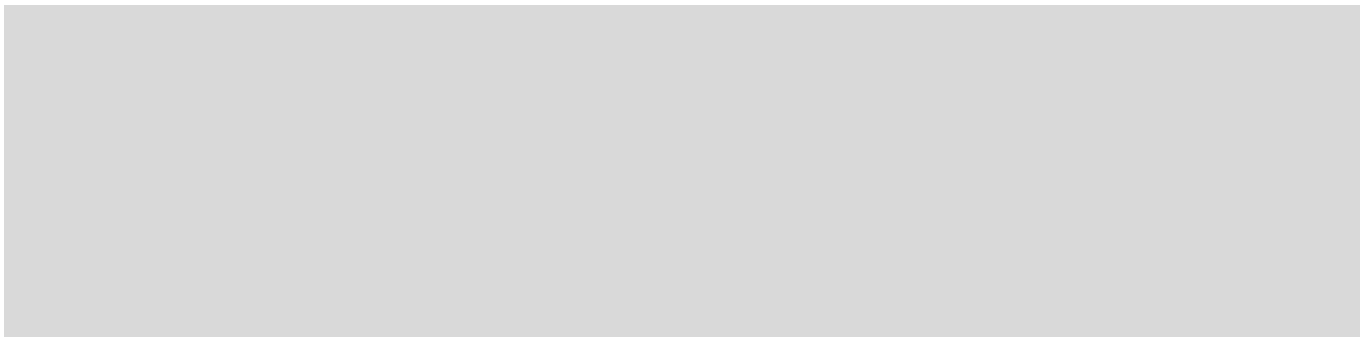
0, 1, 2, 3, 4, 5, End

```
3 while True:
4     value = int(input("숫자를 입력하세요"))
5
6     print("입력 값 : ", value)
7
8     if value < 0:
9         break
10
11 print("End")
```

```
숫자를 입력하세요2
입력 값 : 2
숫자를 입력하세요-1
입력 값 : -1
End
```

Break 문 (2)

- 1~100 사이 정수 중 7의 배수이면서, 11의 배수인 수 중 제일 작은 값을 출력하라
 - 7과 11의 최소공배수



7과 11의 최소공배수는 : 77

Break 문 (3)

- 다음 프로그램의 실행 결과는? 디버거를 활용하여 확인 할 것!

```
2  for value_1 in range(2):
3      for value_2 in range(2):
4          for value_3 in range(2):
5              if value_2 == 1:
6                  break
7
8              print("value 1 : ", value_1,
9                  "value 2 : ", value_2,
10                 "value 3 : ", value_3)
```

중첩 반복문이 있을 경우
Break가 선언 된 블록(Block)의
반복문만 탈출 한다.



continue 문

continue 문 : (1)

- 반복문(Loop) 내 사용
- continue 하위 문장을 실행하지 않고, 반복문 조건식으로 이동

```
1  for value in range(1, 6):
2      print(value)
3
4      if value == 2:
5          continue # 현 시점에서 실행을 중지하고
6                  # 반복문의 조건식으로 이동
7
8      print("-----")
9
10 print("End")
```

1

2
3

4

5

End

continue 문 (2)

- 구구단 중 3의 배수단만 출력하는 프로그램을 continue문을 이용하여 작성하라

3 X 1 = 3	3 X 2 = 6	3 X 3 = 9	3 X 4 = 12	3 X 5 = 15	3 X 6 = 18	3 X 7 = 21	3 X 8 = 24	3 X 9 = 27
6 X 1 = 6	6 X 2 = 12	6 X 3 = 18	6 X 4 = 24	6 X 5 = 30	6 X 6 = 36	6 X 7 = 42	6 X 8 = 48	6 X 9 = 54
9 X 1 = 9	9 X 2 = 18	9 X 3 = 27	9 X 4 = 36	9 X 5 = 45	9 X 6 = 54	9 X 7 = 63	9 X 8 = 72	9 X 9 = 81

continue 문 (3)

- 다음 프로그램의 실행 결과는? 디버거를 활용하여 확인 할 것!

```
1 count = 0
2 while count < 3:
3     for value in range(1, 3):
4         if count == 1:
5             continue
6
7         print("count : ", count, ", value : ", value)
8
9     count += 1
```

중첩 반복문이 있을 경우,
continue가 선언 된 블록(Block)의
반복문으로 이동한다.



pass 문

pass 문

- 선택 또는 반복문(Loop) 내 사용
- 아무것도 실행 되지 않음

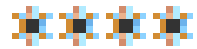
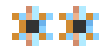
```
1 ~ for value in range(1, 4):  
2 |     pass  
3  
4  
5 ~ for value in range(1, 4):  
6 ~
```

```
1 print("1")  
2  
3 ~ if 3 > 2:  
4 |     pass  
5 |     print("2")  
6  
7 print("3")
```

실습

실습 1

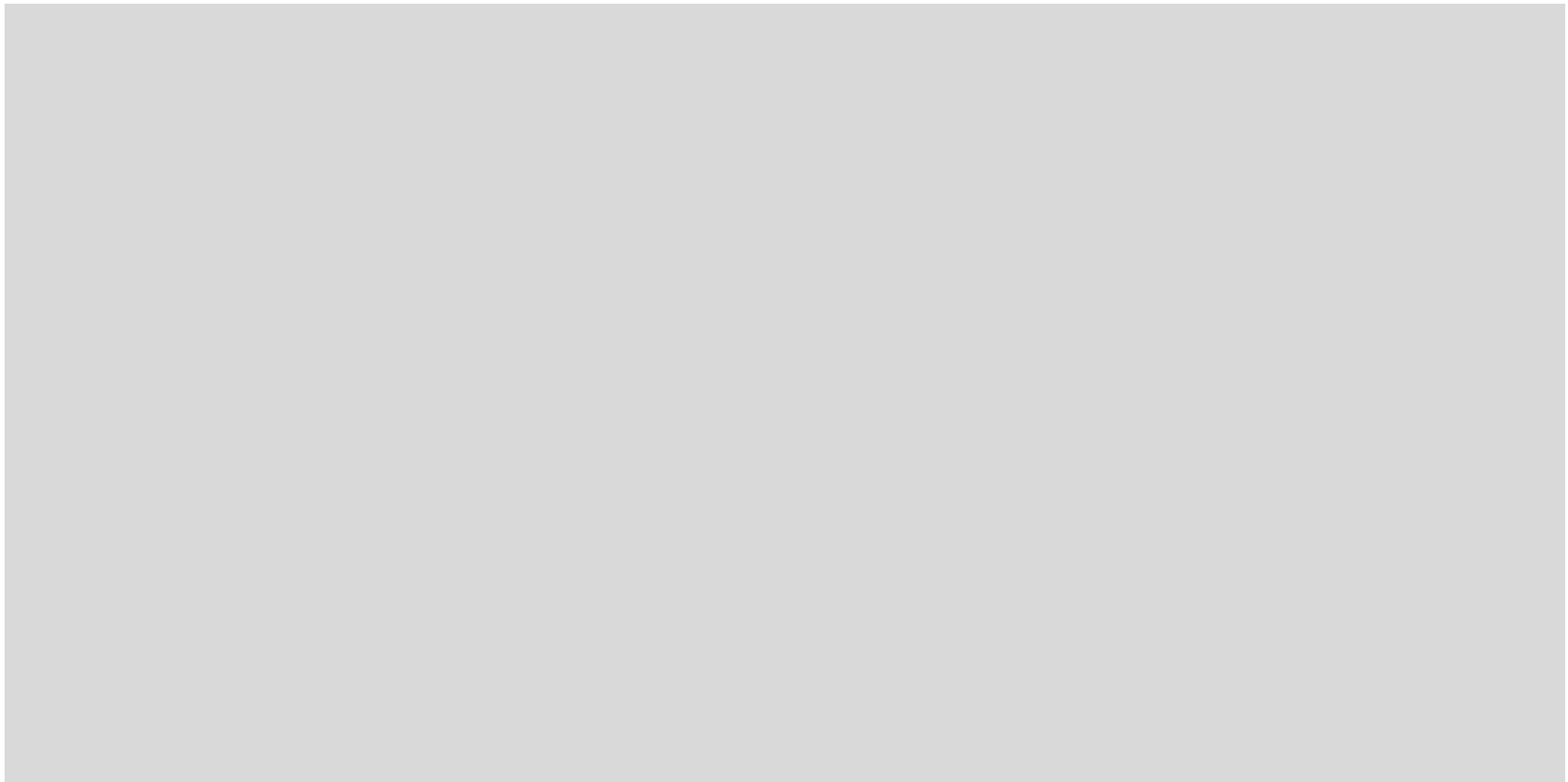
- While문을 사용하여 아래와 같이 출력하는 프로그램을 작성하라



실습 2

- While문을 사용하여 1~1000까지의 자연수 중 3의 배수의 합을 구하라

1~1000 사이 정수 중 3의 배수의 총 합은 : 166833



실습 3

- For 문을 사용하여 아래 문자열 내 'h'의 개수를 출력하는 프로그램을 작성하라

```
1 myString = "hello hyundai hoho"  
2
```

```
9 print("문자열 내 h 갯수 : ", count)
```

문자열 내 h 갯수 : 4

실습 4

- For 문을 사용하여 아래 문자열 내 단어 개수를 출력하는 프로그램을 작성하라

```
1 myString = "It is a great weather with you"
```

```
2
```

```
9 print("문자열 단어 갯수 : ", count)
```

문자열 단어 갯수 : 7

실습 5

- For 문을 사용하여 아래 학생들의 성적에 대한 총합, 평균, 학생 수를 출력하는 프로그램을 작성하라

```
1  score = [99, 29, 30, 40, 20, 60]
2
3
4
5
6
7
8
9
10
11
12
13 print("학생 수 : ", student_num, ", 총점 : ", sum, ", 평균 : ", avg)

    학생 수 : 6 , 총점 : 278 , 평균 : 46.333333333333336
```

Q/A

감사합니다



주문식교육의 산실
영진전문대학교