

# 파이썬 프로그래밍

12차시

조건에 따른 선택  
if ... else



## ⚠ 학습개요

... 조건문

- if
- 콜론과 인덴트 블록

... if else

... if elif

## ⚠ 학습목표

- … 조건문을 이해하고 if 문을 사용할 수 있다.
- … if else 문을 사용할 수 있다.
- … if elif 문을 사용할 수 있다.

## Chapter 1.

# 조건문

P Y T H O N   P R O G R A M M I N G

## ⚠ 조건에 따른 선택을 결정하는 if문

### + 조건문

```
weather = '화창'
if weather == '화창'
    print('어제 산 신발을 신고 가야지!')
```

```
PM = 90 # particulate matter: 미세먼지
if 81 < PM:
    print('마스크를 착용합시다!')
```

[그림12-1] 일상생활에서 날씨나 미세먼지 조건에 따른 선택

## ⚠ 조건에 따른 선택을 결정하는 if문

### + 조건문

- if문에서 논리 표현식 이후에는 반드시 콜론이 있어야 함
- 콜론 이후 다음 줄부터 시작되는 블록은 반드시 들여쓰기(indentation)를 해야 한다. 그렇지 않으면 오류가 발생하니 주의

들여쓰기는 보통 4개의 빈 공백을 사용한다. 통합 개발 환경에서는 대부분의 자동으로 들여쓰기가 수행되는데, 대부분 직업 **Tab**을 사용하면 4개의 빈 공백이 삽입된다.

**if 논리 표현식:**

문장 1  
문장 2

블록의 시작의 표시인 콜론(:)은 반드시 필요하다. :앞의 빈 공간은 선택이다.

진리 표현식의 결과가 True이면 블록인 문장 1과 문장 2가 실행되며, 필요하다면 더 많은 문장을 쓰면 된다.



## ⚠ 일상 코딩: 놀이공원에서 키를 제한하는 놀이 기구

[코딩실습] 키가 140 이상이면 놀이 기구 타기

난이도 기본

```
1. height = 152 # 탑승을 체크할 키를 대입
2. if 140 <= height:
3.     print ('롤러코스터 T-Express, 즐기세요!')
```

**주의** 조건 검사 이후의 콜론을 잊지 말고 3번 줄은 블록으로 들여쓰기하자.

**결과** 롤러코스터 T-Express, 즐기세요!

### Upgrade coding

변수 height에 표준 입력으로 검사할 키를 표준 입력으로 받아 저장하도록 수정해보자.  
키를 139로 입력하면 아무것도 출력되지 않을 것이다.

**힌트** `int(input('키 ?'))`

## Chapter 2.

# if else

P Y T H O N   P R O G R A M M I N G



## ⚠ if ... else 문

+ 논리 표현식 결과인 True와 False에 따라 나뉨

if 논리 표현식:

블록의 시작의 표시인 콜론(:)은 반드시 필요하다. :앞의 빈 공간은 선택이다.

[ ][ ][ ] 문장 1

[ ][ ][ ] 문장 2

문장 2 이후에 **Enter**를 누르면  
자동으로 문장 2와  
열을 맞추지만, 앞으로 이동시켜  
if와 열을 반드시 맞춰  
else:를 입력해야 한다.

else:

else 이후의 콜론도 반드시 필요하며,  
이후 들여쓰기의 블록이 필요하다.

→ [ ][ ][ ] 문장 3

[ ][ ][ ] 문장 4

## ⚠ 일상 코딩: 영화 조조 할인 판정

[코딩실습] 영화 조조 할인 판정

난이도 응용

```
1. from time import localtime
2. hour = localtime().tm_hour
3. mnt = localtime().tm_min

4. if hour < 10:
5.     print('지금 시각: %d시 %d분, 조조 할인 된다.' % (hour, mnt))
6. else:
7.     print('지금 시각: %d시 %d분, 조조 할인 안 된다.' % (hour, mnt))
```

**주의** 1, 2, 3번 줄에 유의해 작성하자. 여러분의 실행 시간에 따라 조조 할인이 결정된다.

**결과** 지금 시각: 15시 59분, 조조 할인 안 된다.    ...    지금 시각: 9시 58분, 조조 할인 된다.

## Chapter 3.

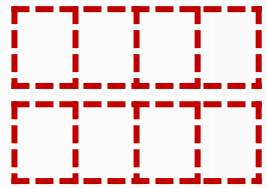
# if elif

P Y T H O N   P R O G R A M M I N G

## ⚠ 다중 택일 결정 구조인 if ... elif

키워드 elif는 else if를 의미하며,  
논리 표현식 1이 False여야  
논리 표현식 2를 검사한다.

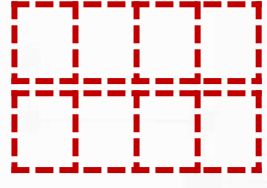
if 논리 표현식:



문장 1

문장 2

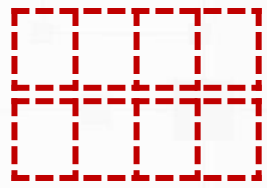
elif 표현식 2:



문장 3

문장 4

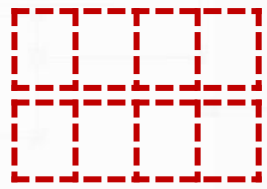
elif 표현식 3:



문장 5

문장 6

else:



문장 7

문장 8

else 이후의 콜론도 반드시 필요하며,  
이후 들여쓰기 블록이 필요하다.

## ⚠ 다중 택일 결정 구조인 if ... elif

```
>>> point = 82 # 성적
>>> if 90 <= point:
...     print('점수 {}, 성적 {}'.format(point, 'A'))
... elif 80 <= point:
...     print('점수 {}, 성적 {}'.format(point, 'B'))
... elif 70 <= point:
...     print('점수 {}, 성적 {}'.format(point, 'C'))
... elif 60 <= point:
...     print('점수 {}, 성적 {}'.format(point, 'D'))
... else:
...     print('점수 {}, 성적 {}'.format(point, 'F'))
...
점수 82, 성적 B
```

## ⚠ 일상 코딩: 미세먼지 농도에 따른 미세먼지 경고 예보

### + [표] 미세먼지(10마이크로그램) 판정 구간

미세먼지 농도 구간	if...elif문(변수 PM에 미세먼지 농도 저장)	등급
151이상	if 151 <= PM:	매우 나쁨
81 이상, 151 미만	elif 81 <= PM:	나쁨
31 이상, 81 미만	elif 31 <= PM:	보통
31 미만	else:	좋음



## ⚠ 일상 코딩: 미세먼지 농도에 따른 미세먼지 경고 예보

[코딩실습] 미세먼지 예보

난이도 기본

```
1. PM = float('미세먼지(10마이크로그램)의 농도는? '))
2. if 151 <= PM:
3.     print('미세먼지 농도: {:.2f}, 등급: {}'.format(PM, '매우 나쁨'))
4. elif 81 <= PM:
5.     print('미세먼지 농도: {:.2f}, 등급: {}'.format(PM, '나쁨'))
6. elif 31 <= PM:
7.     print('미세먼지 농도: {:.2f}, 등급: {}'.format(PM, '보통'))
8. else:
9.     print('미세먼지 농도: {:.2f}, 등급: {}'.format(PM, '좋음'))
```

**주의** 2, 4, 6, 8번 줄에는 콜론, 3, 5, 7, 9번 줄에는 들여쓰기가 반드시 필요하다.

**결과**

미세먼지(10마이크로그램)의 농도는? 140  
미세먼지 농도: 140.00, 등급: 나쁨

미세먼지(10마이크로그램)의 농도는? 30  
미세먼지 농도: 30.00, 등급: 좋음

## ⚠ f 문자열

### + 포맷 문자열(간단히 f-문자열이라고도 한다)

- 문자열 내부에 표현식 값을 쉽게 삽입한 문자열
- 시작 인용 부호(' 또는 ") 또는 삼중 인용 부호('' 또는 ''') 앞에 **f 또는 F** 를 붙여 문자열을 시작
  - 포맷 문자열 안에 표현식을 삽입하려면 {표현식} 처럼 **중괄호 내부에** 표현식(변수와 리터럴 상수로 구성된 연산식) 값을 사용
- 2016년 **버전 3.6** 추가

## ⚠ f 문자열

```
>>> f'포맷문자열'
'포맷문자열'
>>> import math
>>> print(f'원주율 pi: {math.pi:.3f}')
원주율 pi: 3.142
>>> print(F'오일러 수(자연수) e: {math.e:.3f}')
오일러 수(자연수) e: 2.718
>>>
```



## 좀 더 알아봅시다!

### + 범위라는 단어를 쓰는 함수 range()

- 내장 함수 **range()**는 일정 간격의 정숫값의 나열인 **시퀀스**를 생성해주는 함수다.

함수의 인자는 총 **3개**로, **적어도 1개**의 인자를 지정해 줘야 한다.  
만약 인자 중 하나만 지정하면 자동으로 **stop** 값이 지정되는데,  
stop의 이전 정숫값까지 포함된다. 시퀀스의 시작인 **start**와  
인접한 숫자의 간격인 **step**은 자동으로 각각 0, 1로 지정된다.  
인자를 2개 지정하면 자동으로 start와 stop 값을 설정하게 되며,  
증가값은 자동으로 1로 지정된다.

- 모든 인자는 문자열 또는 부동소수점 숫자나 다른 유형이 올 수 없으며,  
**반드시 정수여야 한다**

### Python range(6)





## 좀 더 알아봅시다!

### + 범위라는 단어를 쓰는 함수 range()

- 다음 코드에서 볼 수 있듯이 list() 함수 내부에 range() 함수를 사용하면 바로 시퀀스 결과를 알 수 있다.

```
>>> list(range(4))
[0, 1, 2, 3]
>>> list(range(6, 10))
[6, 7, 8, 9]
>>> list(range(10, 20, 2))
[10, 12, 14, 16, 18]
```

함수 호출	인자	결과
range(4)	stop	0, 1, 2, 3
range(6, 10)	start, stop	6, 7, 8, 9
range(10, 20, 2)	start, stop, step	10, 12, 14, 16, 18

# SUMMARY



## ⚠ 조건문

... if

... 콜론과 인덴트 블록

## ⚠ if else



# SUMMARY



## ⚠ if elif