



# Python 기초 - 제어문

Made by Jay Hong

## 1. 제어문?



## 1. 제어문?

1. if 문
2. for 문
3. while 문

## 1) if문 뜻

- ✓ Python에서 조건에 따라 다른 코드를 실행하고 싶을 때 사용
- ✓ if문은 조건문과 실행문으로 구성됨
- ✓ if 문의 조건이 참 (True)이면 실행문이 돌아가는 구조
- ✓ 즉, if True : 실행

## 2) if elif else

✓ if 문에서 여러가지 조건이 들어갈 수 있음

✓ if 조건 1 :  
    실행할 코드

elif 조건 2 :  
    실행할 코드

elif 조건 3 :  
    실행할 코드

...

else :  
    실행할 코드

의 형태로 구성되어 있다.

if : 조건문에서 반드시 하나 이상은 들어가야 함  
해당 조건을 만족하면, 실행할 코드를 실행

elif : 위에 적혀진 조건들 이외의 새로운 조건을 추가함  
해당 조건을 만족하면, 코드를 실행하고 아니면 빠져 나감

else : 가장 마지막에 적히는 if문의 끝  
위의 모든 조건에 해당이 안될 경우, 코드를 실행

### 3) Nested if문

- ✓ if 문 속에 if문을 적을 수 있다.



### 1) for문 뜻

- ✓ 특정 객체를 이용하여 반복하는 방법

## 2) for문 형식

### ✓ 형식

for variable in variables :  
statement

- ✓ variables는 range, 리스트, 튜플, 등이 들어갈 수 있다.
- ✓ variables의 원소들이 하나씩 처음부터 끝까지 들어간다.
- ✓ 그러면서 statement를 실행한다.
  - ✓ statement는 여러 개가 들어갈 수 있다.
  - ✓ 따라서 총 len(variables) 만큼 statement를 반복해서 실행한다.



### 3) 함께 사용되는 method

- ✓ enumerate : 번호를 메겨 줌
- ✓ List Comprehension
  - ✓ 효율적으로 리스트를 만드는 방법
  - ✓ for문을 압축한 기능을 가지고 있음
- ✓ Dict Comprehension
  - ✓ List Comprehension 마찬가지로, Dictionary를 만들어냄
- ✓ zip
  - ✓ 같은 길이를 가진 여러 리스트에서, 같은 인덱스에 해당하는 원소를 묶어서 반환

## 4) Nested For Loop

- ✓ 반복문 속에 반복문을 만들 수 있다.



#### 1) while문 뜻

- ✓ 조건이 True일 동안 반복하는 방법
- ✓ 무한 반복에 빠지지 않도록 주의해야 함.

## 2) while문 형식

- ✓ 형식  
while 조건 :  
statement
- ✓ 조건이 True인 동안, statement는 계속해서 시행된다.

### 3) 루프 제어문

- ✓ break
  - ✓ for문이나 while문에서 빠져나오기 위한 방법
  - ✓ 특정 조건일 때 아예 반복문에서 빠져나옴
- ✓ continue
  - ✓ 특정 조건일 때는 반복문 자체를 다음으로 넘어감
- ✓ pass
  - ✓ 아무일도 하지 않고 넘어감을 의미