



D&A

Basic Session 3차시

파이썬 기초 2.

2022 / 03 / 22
D&A 운영진 이경욱



CONTENTS.

01 If문

02 while문

03 for문

04 comprehension



01. If문

If문이란? -> 조건의 참과 거짓을 판단하는 문장

```
if 조건문:  
    조건이 True일 때 수행할 문장  
  
elif 조건문:  
    if 조건이 아닌 것중 elif 조건이 True일 때 수행할 문장  
  
else:  
    위의 if, elif 조건에 해당하지 않는 모든 경우에 수행할 문장
```

- ➔ 조건문은 True와 False 둘 중 하나로 결정되도록
-> True와 False 그 자체 혹은 연산자를 통한 결과
- ➔ elif와 else는 if없이 단독으로 사용 불가
- ➔ 수행할 문장은 조건문 밑에서 들여쓰기!
- ➔ 조건문 뒤에는 콜론(:)붙이기



01. If문

예제를 통해 이해해보기

```
money = 5000
if money < 4000:
    print("학식을 먹자")
elif money < 10000:
    print("짜장면 먹자")
elif money < 20000:
    print("짜장면에 탕수육추가!")
else:
    print("요셉이형 내가 밥사줄게 ㅎㅎ")
```

짜장면 먹자

Line2 if money < 4000
-> money가 4000 미만이라면

Line4 elif money < 10000
-> money가 4000미만이 아니고 + 10000미만이라면

Line6 elif money < 20000
-> money가 4000미만과 10000미만이 아니면서 20000미만이라면

Line8 else:
-> 위의 모든 경우가 아니라면? -> money가 20000이상이라면

➡ 경우에 따라 elif는 여러 개 설정가능

➡ else는 위의 모든 조건문에 해당하지 않는 경우를 가리킴



02. while문 - 반복해서 문장을 수행하는 경우

while문의 기본구조

while 조건문:

조건문이 참일 동안 수행할 문장1
조건문이 참일 동안 수행할 문장2
조건문이 참일 동안 수행할 문장3
.
.
.

➔ 조건문이 참일 경우 **while** 조건문:
밑의 문장들이 반복해서 수행됨

➔ 조건문은 if문과 마찬가지로
True 또는 False가 들어감

예제를 통해 이해해보기

```
age = 0
while age < 20:
    print(f"{age}살이면 세빳돈 받아야지")
    age = age + 1
```

0살이면 세빳돈 받아야지
1살이면 세빳돈 받아야지
2살이면 세빳돈 받아야지
3살이면 세빳돈 받아야지
4살이면 세빳돈 받아야지
5살이면 세빳돈 받아야지
6살이면 세빳돈 받아야지
7살이면 세빳돈 받아야지
8살이면 세빳돈 받아야지
9살이면 세빳돈 받아야지
10살이면 세빳돈 받아야지
11살이면 세빳돈 받아야지
12살이면 세빳돈 받아야지
13살이면 세빳돈 받아야지
14살이면 세빳돈 받아야지
15살이면 세빳돈 받아야지
16살이면 세빳돈 받아야지
17살이면 세빳돈 받아야지
18살이면 세빳돈 받아야지
19살이면 세빳돈 받아야지

➔ age라는 변수는 while문이 한 번 반복될 때마다
1씩 더해지는 중

➔ 조건문은 age가 20 미만일 때 참이므로
Age가 20이 되었을 때 조건문이 거짓이 되어
while문의 반복이 멈춤



02. while문 - 반복해서 문장을 수행하는 경우

while문 예제

```
age = 8
while age < 20:
    if age < 14:
        print(f"{age}살이면 초등학생!")
    elif age < 17:
        print(f"{age}살이면 중학생!")
    else:
        print(f"{age}살이면 고등학생")

    age = age + 1
```

```
8살이면 초등학생!
9살이면 초등학생!
10살이면 초등학생!
11살이면 초등학생!
12살이면 초등학생!
13살이면 초등학생!
14살이면 중학생!
15살이면 중학생!
16살이면 중학생!
17살이면 고등학생
18살이면 고등학생
19살이면 고등학생
```

- ➡ While문 내에 if문을 통해 조건에 따라 다른 문장 수행 중
- ➡ 들여쓰기가 어디에서 이루어졌는지 확인!!

02. while문 – 빠져나가기

while문 강제로 빠져나가기 **break**

```
money = 10000
coffee = 2700
count = 1
while True:
    changes = money - coffee*count
    if changes < 0 :
        break
    else:
        print(f"커피 {count}잔 주문받았고 잔돈은 {changes}입니다")
        count = count + 1
```

커피 1잔 주문받았고 잔돈은 7300입니다
커피 2잔 주문받았고 잔돈은 4600입니다
커피 3잔 주문받았고 잔돈은 1900입니다

- While **True**:로 반복이 계속됨
- **Break**문을 통해서 changes<0인 경우 while문을 강제로 빠져나감!

02. while문 - 반복해서 문장을 수행하는 경우

while문 처음으로 돌아가기 `continue`

```
: a = 0
  while a < 10:
      a = a + 1

      if a % 2 == 0:
          continue

      print(a)
```

1
3
5
7
9

- ➔ 기본 조건은 a가 10 미만일 경우!
- ➔ if문을 통해 a가 짝수일 경우,
`continue`문 실행
->continue 다음의 문장을 건너뛴
- ➔ 따라서 짝수를 건너뛰고 홀수만 print

02. while문 - 반복해서 문장을 수행하는 경우

조건에 따라 건너뛰기 `pass`

```
a = 0
while a < 10:
    a = a + 1

    if a % 2 == 0:
        pass
    else:
        print(a)
```

1
3
5
7
9

➡ 해당 조건이면 그냥 지나감

`continue`는 `continue`이후의 문장 건너뛰기
`pass`는 단순히 해당 조건에 코드가 필요 없을 때
사용

03. for문 (feat. 반복가능한 객체)

반복 가능한 객체란??

- ➡ 리스트, 튜플, 세트, 딕셔너리와 같이 요소를 하나씩 꺼내 볼 수 있는 객체
- ➡ 딕셔너리와 세트는 순서가 없지만, 요소를 하나씩 꺼내볼 수 있음!

for문 기본 구조

```
for 변수명 in 반복가능한객체:
    수행할문장1
    수행할문장2
    .
    .
    .
```

- ➡ 여기서 변수명이란 반복가능한 객체에서 반복적으로 꺼내 오는 각 요소를 for문내에서 사용할 변수명을 뜻함

예제를 통해 이해해보기

```
nums = [1, 2, 3, 4]
for num in nums:
    print(num)
```

1
2
3
4

- ➡ 여기서 num은 nums의 요소를 for문 안에서 사용할 변수명

- ➡ num변수를 print()하자 객체 내의 요소들이 반복해서 print()됨

for문 풀어서 보기

```
num = 1
print(num)
# -----for문 한 번 반복-----
num = 2
print(num)
# -----for문 두 번 반복-----
num = 3
print(num)
# -----for문 세 번 반복-----
num = 4
print(num)
# -----for문 네 번 반복-----
```

1
2
3
4

num이라는 변수에 nums의 요소가 반복적으로 들어감



03. for문

for문 예제

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

3
7
11

- 리스트의 요소로 튜플
- 요소 내의 두 개 변수를 동시에 받을 수 있음

for문과 continue

```
scores = [96, 82, 64, 74, 30, 20]
for score in scores:
    if score > 60:
        continue
    else:
        print(f"{score}는 불합격입니다")
```

30는 불합격입니다
20는 불합격입니다

- while문과 마찬가지로 for문의 처음으로 돌아감
(pass와 break도 동일하게 사용 가능!)

구구단 만들기로 for문 중첩 이해하기

```
for i in range(2, 10):
    for j in range(1, 10):
        print(f"{i} * {j} = {i*j}")
```

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
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24

- range또한 for문에서 객체로 많이 쓰임
- 첫 for문의 첫 변수 i가 2일 때,
다음 for문에서j가 1부터 9까지 반복



04. Comprehension

Comprehension이란?

-> “내포”라는 뜻으로 반복되거나 특정 조건을 만족하는 값을 가지는 자료형을 쉽게 만들기 위한 기법

EX)

```
a = []  
for i in range(1, 11):  
    a.append(i*2)  
  
a  
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

→ for문을 통해 1~10의 두 배의 수를 리스트에 하나씩 저장

Comprehension을 쓰면?



```
a = [i*2 for i in range(1, 11)]  
a  
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

List comprehension 기본구조

```
a = [i*2 for i in range(1, 11)]  
a  
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

표현식

기존의 for문

→ for문을 통해 반복되는 요소들이 **표현식의 형태로** 리스트 안에 저장됨!!

중첩된 반복문으로 구구단이 담긴 리스트 만들기

```
a = [j*i for i in range(2, 10) for j in range(1, 10)]
```

표현식

상위 for문

하위 for문

→ 중첩도 가능!

04. Comprehension

List comprehension으로 조건 추가하기
EX)

```
a = [i for i in range(1, 11) if i%2==0]
a
```

표현식 for문

조건문

→ for문을 통해 반복되는 요소들이 조건문을 거쳐
짝수만 리스트에 들어가게됨

set comprehension - list와 동일!

```
a = {i*2 for i in range(1, 11)}
a
```

{2, 4, 6, 8, 10, 12, 14, 16, 18, 20}

Dictionary comprehension 기본구조

```
name = ["제니", "지수", "리사", "로제"]
birth = [96, 95, 97, 97]
{i:j for i, j in zip(name, birth)}
```

{'제니': 96, '지수': 95, '리사': 97, '로제': 97}

표현식

For문

→ List와 원리는 동일하나 key:value의 구조를
채우기 위해 표현식에 변수가 두개 필요함

Dictionary comprehension으로 key:value 위치 변경해보기

```
brand = {"휴대폰" : "갤럭시", "노트북" : "맥북", "자동차" : "테슬라"}
{j:i for i, j in brand.items()}
```

{'갤럭시': '휴대폰', '맥북': '노트북', '테슬라': '자동차'}

→ 변수의 위치 주목!

```
brand.items()
dict_items([('휴대폰', '갤럭시'), ('노트북', '맥북'), ('자동차', '테슬라')])
```

→ Dict.items() 에는(key, value)가 요소로 들어가있음



첨부자료 출처

폰트

네이버 글꼴 모음 _ 나눔 스퀘어 사용
출처 : <https://hangeul.naver.com/font>





D&A

Basic Session 3차시 파이썬 기초2

Thank You.

2022 / 03 / 22
D&A 운영진 이경욱



2022 빅데이터 분석 학회 D&A