

## 4. 파이썬 반복문

---

1. "~하는 동안에" 라는 뜻으로 조건식이 참인 경우에만 반복 처리하는 반복문은 ?

1. for
2. during
3. whale
4. while +++++

2. 주로 리스트의 모든 요소들을 하나씩 꺼내서 처리하는 방식의 반복문은?

1. for +++++
2. during
3. whale
4. while

3. 다음 코드의 실행 결과는?

```
while True:
    print("apple")
```

1. 에러가 난다.
2. 아무 것도 실행되지 않는 것처럼 보인다.
3. "apple"
4. "apple"이 무한히 출력된다. +++++

4. 다음 코드의 실행 결과는?

```
a = 3

while a < 5 :
    print(a)
```

1. 에러가 난다.
2. 아무 것도 실행되지 않는 것처럼 보인다.
3. "3 4 5"
4. "3"이 무한히 출력된다. +++++

## 5. 다음 코드의 실행 결과는?

```
a = 3

while a < 5 :
    print (a)
    a = a + 1
```

1. 에러가 난다.
2. "3"이 무한히 출력된다.
3. 3 ++++
4. 3
4. 4
5. 5

## 6. 다음 코드의 실행 결과는?

```
a = 3

while a < 5 :
    print (a)
a = a + 1
```

1. 에러가 난다.
2. "3"이 무한히 출력된다. ++++
3. 3
4. 4
4. 3
4. 4
5. 5

## 7. 다음 코드의 실행 결과는?

```
a = [1, 2, 3]

for item in a :
    print(a)
```

1. [1, 2, 3]

- 2. 1  
2  
3
- 3. [1, 2, 3] ++++  
[1, 2, 3]  
[1, 2, 3]
- 4. 1 2 3

## 8. 다음 코드의 실행 결과는?

```
a = [1, 2, 3]

for item in a :
    print(item)
```

- 1. [1, 2, 3]
- 2. 1 ++++  
2  
3
- 3. [1, 2, 3]  
[1, 2, 3]  
[1, 2, 3]
- 4. 1 2 3

## 9. 다음 코드의 실행 결과는?

```
a = [1, 2, 3]

for item in a :
    if item = 2 :
        break

    print(item)
```

- 1. 에러가 발생한다. ++++
- 2. 1  
2  
3
- 3. [1, 2, 3]  
[1, 2, 3]  
[1, 2, 3]
- 4. 1

## 10. 다음 코드의 실행 결과는?

```
a = [1, 2, 3]

for item in a :
    if item == 2 :
        break

    print(item)
```

1. 에러가 발생한다.
2. 1  
2  
3
3. [1, 2, 3]  
[1, 2, 3]  
[1, 2, 3]
4. 1 +++++

## 11. 다음 코드의 실행 결과는?

```
a = [1, 2, 3]

for item in a :
    if item == 2 :
        continue

    print(item)
```

1. 1  
2
2. 1  
2  
3
3. 2
4. 1 +++++  
3

## 12. 다음 코드의 실행 결과는?

```
for i in range(3):  
    print (i)
```

1. 1
- 2
- 3
2. 3
3. 2
4. 0 +++++  
1  
2

### 13. 다음 코드의 실행 결과는?

```
a = [i*3 for i in range(3)]  
print (a)
```

1. [1, 2, 3]
2. [1, 3, 6]
3. 6
4. [0, 3, 6] +++++