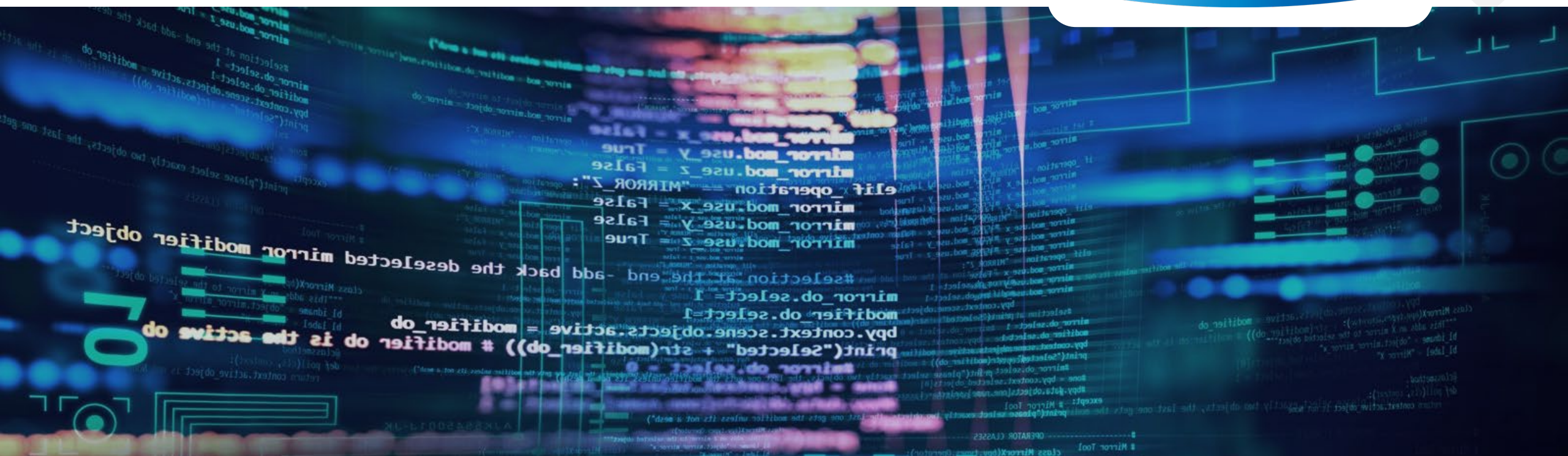


# Lecture 4

조건문

KAIST





- if문에 있는 조건의 진리값에 따라 코드의 실행여부가 결정된다
- 참이면 if문 밑에 있는 코드가 실행되고 거짓이면 코드를 실행하지 않고 탈출한다

```
1  if 조건_a:  
2      #코드_a
```





➤ 여러 조건을 확인해야 될 수도 있다

➤ **elif** (else if의 약자) 문을 쓰면 된다

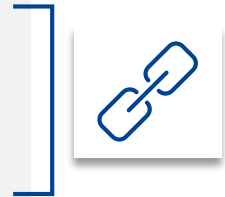
➤ 조건문의 조건이 참이면 아래의 코드가 실행이 된 후 모든 연결된 조건문에서 탈출한다

- elif문에 있는 조건\_b를 확인 하지 않고, elif 밑에 있는 코드\_b도 실행이 되지 않는다

➤ 거짓이면 연결된 다음 조건문으로 간다

➤ 들여쓰기가 중요하다

```
1  if 조건_a:
2      #코드_a
3  elif 조건_b:
4      #코드_b
```





- elif문은 **연결된 조건문 안에 여러 번** 사용할 수 있다
- 연결된 조건문들에서 **elif 이후 if를 쓰면 안 된다**

```
1  if 조건_a:
2      #코드_a
3  elif 조건_b:
4      #코드_b
5  elif 조건_c:
6      #코드_c
7  elif 조건_d:
8      #코드_d
9  if 조건_e:
10     #코드_e
```





- 모든 조건이 거짓일 때 실행하고 싶은 코드가 있을 수도 있다
- **else문**을 쓰면 된다
  - else문은 조건이 없다
- else문 이후 다른 조건문을 연결할 수 없다
  - else문이 조건문의 끝이다

```
1  if 조건_a:
2      #코드_a
3  elif 조건_b:
4      #코드_b
5  elif 조건_c:
6      #코드_c
7  elif 조건_d:
8      #코드_d
9  else:
10     #코드_e
```





- 조건문에 있는 조건의 진리값에 따라 다른 코드를 실행
- if로 무조건 시작
- elif는 0개에서 여러개까지 가능
- else는 최대한 1번까지 가능

```
1  if 조건_a:
2      #코드_a
3  elif 조건_b:
4      #코드_b
5  elif 조건_c:
6      #코드_c
7  elif 조건_d:
8      #코드_d
9  else:
10     #코드_e
```





- 한 줄로 조건문을 표현할 수 있다
- trueValue if condition else falseValue

```
1  if 조건_a:  
2      #코드_a  
3  elif 조건_b:  
4      #코드_b  
5  else:  
6      #코드_c
```

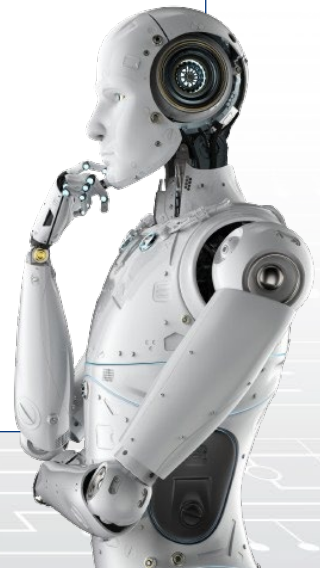
```
1  코드_a if 조건_a else 코드_b if 조건_b else 코드_c
```





- 반복문안에서 사용할 수 있는 키워드가 있다
- break 예약어는 실행되자마자 반복문에서 바로 탈출한다

```
1   for element in iterable:  
2       if condition:  
3           break
```

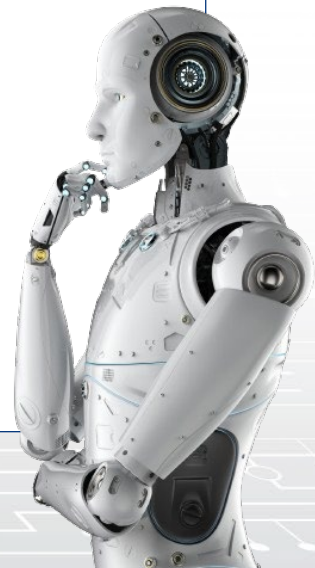






- continue 키워드는 실행되자마자 남은 코드를 실행하지 않고 실행 순서가 다시 반복문 첫 줄로 돌아간다
- for문은 객체를 새로 받고, while문은 다시 조건을 확인한다

```
1   for element in iterable:
2       if condition:
3           continue
4
5   # code used only if condition is False
```





➤ break, continue 키워드는 제일 가까이 있는 반복문에만 영향을 준다

```
1 for i in range(3):  
2     for j in range(10000):  
3         if j == 1:  
4             break  
5         print(f"j:{j}")  
6     print(f"i:{i}")
```

```
j: 0  
i: 0  
j: 0  
i: 1  
j: 0  
i: 2
```



➤ break, continue 키워드는 제일 가까이 있는 반복문에만 영향을 준다

```
1  for i in range(3):
2      for j in range(3):
3          if i%2 or j%2:
4              continue
5          print(f"j:{j}")
6          print(f"i:{i}")
```

```
j: 0
j: 2
i: 0
i: 1
j: 0
j: 2
i: 2
```