



K-Digital Training 스마트 팩토리 3기



# 반복문

## 반복문



- 컴퓨터의 계산 속도는 사람보다 빠르므로 반복시키는 일을 하기 좋음
- 반복문을 사용하면 코드를 줄이고 똑같은 일을 반복하지 않아도 됨
  - ex) 1부터 100까지 출력, print(1), print(2),,,...
- for문 지정한 횟수만큼 반복
- while 특정 조건이 만족되는 동안 반복

## for문



#### • 리스트에 대한 반복

```
for 변수 in 리스트:
실행할 문장1
실행할 문장2
실행할 문장3
```

```
shop = ["반팔", "바지", "이어폰", "키보드"]

for i in shop:
    print(i)

...

반팔
    바지
    이어폰
    키보드
```

# range 함수



- for문에서 자주 쓰임, 정수 리스트 반환
- range(): 지정된 범위의 값을 반환
- range(시작값, 끝값+1, 증가값)
  - 시작값 생략: O부터
  - 증가값 생략: 1씩 증가

```
>>> a = range(10)
>>> a # range(0, 10): 0 이상부터 10 미만까지
>>> list(a) # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
\Rightarrow b = range(1, 11)
>>> b # 1 이상부터 11 미만까지
>>> list(b) # [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> c = range(1, 10, 2)
>>> c # 1 이상부터 11미만까지 2씩 증가
>>> list(c) # [1, 3, 5, 7, 9]
```

# 이중 for문



- 반복문 안에 또 다른 반복문
- 중첩 반복문 (외부/내부 반복문)
  - 내부 반복문은 외부 반복문이 한번 반복할 때마다 새롭게 실행

```
for y in range(3): # 외부 반복문
for x in range(2): # 내부 반복문
    print(f"y: {y}, x: {x}")
    print() # 내부 반복문이 종료될 때마다 실행
```

# while문



- while: ~하는 동안, ~하는 동안에
- 조건식이 참이면 반복, 문장이 끝나는 곳에서 조건식으로 돌아와 반복함
- 반복 구간이 없으므로 끝을 의미하는 문장을 넣어서 사용

```
while 조건문:
수행할 문장1
수행할 문장2
수행할 문장3
```

```
count = 0
while (count <= 5):
    print(count)
    count = count + 1 # Line2 이동
print("끝")
```

# while문 — 무한 루프



- while문에서 무한 루프 (무한 반복)를 사용할 수 있는데 조건식을 항상 참이 되게 하면 됨
- 반복을 중지하고 싶다면?? crtl + c

```
while True:
launch = input("오늘 점심 후보? ")
print(launch)
print("done")
```

## break문



- 프로그램 종료 없이 반복문 탈출하는 법
- break: 깨다, 부수다 -> 반복문 흐름을 깨다, 부수다

```
while True:
lunch = input("오늘 점심 리스트")
if lunch == "그만":
break # 반복문 흐름을 깨다 = 탈출
print(lunch+"!!!")
print("done")
```

# continue문



- continue : 계속하다 -> 다음 조건으로 계속 가다
- continue문을 만나면 무조건 블록의 남은 부분을 건너뛰고 반복문의 처음으로 돌아감 -> 처음부터 반복문을 수행한다고 생각하면 됨

```
      num = 0

      while num < 10:</td>
      4

      num = num + 1
      6

      if num % 2 != 0:
      8

      continue # 아래 print문 실행x
      8

      print(num)
      10
```