

파이썬 프로그래밍

연산자

수치연산자

연산자	의미	예제
+	더하기	$1 + 1 = 2$
-	빼기	$3 - 2 = 1$
*	곱하기	$5 * 2 = 10$
/	나누기	$6 / 3 = 2.0$
**	제곱	$2 ** 3 = 8$
%	나머지	$5 \% 3 = 2$
//	몫	$5 // 3 = 1$

비교연산자

연산자	의미	예제
>	~보다 크다	10 > 3
>=	~보다 크거나 같다	4 >= 7
<	~보다 작다	10 < 3
<=	~보다 작거나 같다	5 <= 5
==	좌항과 우항이 같다	3 == 3
!=	좌항과 우항이 다르다	1 != 3

```
print(10 > 3) # True
print(4 >= 7) # False
print(10 < 3) # False
print(5 <= 5) # True
```

```
# 좌항과 우항이 같은지 비교
print(3 == 3) # 같으므로 True
print(4 == 2) # 다르므로 False
print(3 + 4 == 7) # 같으므로 True
```

```
# 좌항과 우항이 다른지 비교
print(1 != 3) # 다르므로 True
```

논리연산자

연산자	의미	예제
and	두 항이 모두 참이면 참이다.	(3 > 0) and (3 > 5)
or	두 항 중 하나라도 참이면 참이다.	(3 > 0) or (3 > 5)
not	~의 반대	not(1 != 3)

```
print((3 > 0) and (3 > 5))  
# 좌항(3 > 0)은 참이지만 우항(3 > 5)는 거짓이므로 False
```

```
print((3 > 0) or (3 > 5))  
# 좌항(3 > 0)이 참이므로 우항(3 > 5)이 거짓이라도 True
```

```
print(not(1 != 3))  
# 1과 3은 다르므로 True 인데, True의 반대이므로 False
```

연산축약

```
>>> number=2+3*4
>>> number=number+2
>>> number += 2 # number = number + 2 와 동일
>>> print(number) # 18
>>> number *= 2 # number = number * 2 와 동일
>>> print(number) # 36
>>> number /= 2 # number = number / 2 와 동일
>>> print(number) # 18
>>> number -= 2 # number = number - 2 와 동일
>>> print(number) # 16
>>> number %= 2 # number = number % 2 와 동일
>>> print(number) # 0
```

숫자처리함수

함수 이름	의미	예제
abs	절대값	abs(-5)
pow	제곱	pow(4, 2)
max	가장 큰 값	max(5, 12)
min	가장 작은 값	min(5, 12)
round	반올림	round(3.14)

```
print(abs(-5)) # -5 의 절대값 = 5
print(pow(4, 2)) # 4의 2제곱 = 4 * 4 = 16
print(max(5, 12)) # 5 와 12 중 큰 값 = 12
print(min(5, 12)) # 5 와 12 중 작은 값 = 5
print(round(3.14)) # 3.14 의 반올림 = 3
print(round(4.99)) # 4.99 의 반올림 = 5
```

숫자처리함수

그리고 math 모듈을 활용하면 보다 다양한 숫자 처리도 가능합니다.

함수 이름	의미	예제
floor	내림	floor(4.99)
ceil	올림	ceil(3.14)
sqrt	제곱근	sqrt(14)

```
from math import *
```

```
print(floor(4.99)) # 4.99 의 내림 = 4
print(ceil(3.14)) # 3.14 의 올림 = 4
print(sqrt(16)) # 16 의 제곱근 = 4
```

```
import math
```

```
print(math.floor(4.99)) # math. 과 함께 사용
print(math.ceil(3.14))
print(math.sqrt(16))
```


랜덤함수

```
>>> from random import * # random 모듈에서 모든 것들을 가져다 쓰겠다는 의미
>>> print(random()) # 0.0 이상 1.0 미만의 임의의 값 생성
>>> print(random() * 10) # 0.0 이상 10.0 미만의 임의의 값 생성
>>> print(int(random() * 10)) # 0 이상 10 미만의 임의의 정수 값 생성
>>> print(int(random() * 10) + 1) # 1 이상 10 이하 (11 미만) 의 임의의 정수 값 생성
>>> print(random() * 10) # 0.0 이상 10.0 미만의 임의의 값 생성
>>> print(int(random() * 10)) # 0 이상 10 미만의 임의의 정수 값 생성
>>> print(int(random() * 10) + 1) # 1 이상 10 이하 (11 미만) 의 임의의 정수 값 생성
>>> print(randrange(1, 46)) # 1 이상 46 미만의 임의의 정수 값 생성
>>> print(randint(1, 45)) # 1 이상 45 이하(45를 포함해요!!)의 임의의 정수 값 생성
```

제어문

조건문(if): 참과 거짓을 판단하는 문장

- if문은 왜 필요할까?
- Ex) "돈이 있으면 택시를 타고, 돈이 없으면 걸어 간다."

```
>>> money = True
>>> if money:
...     print("택시를 타고 가라")
... else:
...     print("걸어 가라")
...
택시를 타고 가라.
```

조건문(if)

if 조건문:

수행할 문장1

수행할 문장2

...

else:

수행할 문장A

수행할 문장B

주의사항: 1. 조건문 다음에 콜론(:)
2. 들여쓰기

조건문(if)

```
>>> money = 2000
>>> if money >= 3000:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
```

걸어가라

- 조건문이 거짓이 되기 때문에 else문 다음 문장을 수행

조건문(if)

```
>>> money = 2000
>>> card = True
>>> if money >= 3000 or card:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
```

조건문(if)

```
>>> 1 in [1, 2, 3] # True
>>> 1 not in [1, 2, 3] # False
>>> 'a' in ('a', 'b', 'c') # True
>>> 'j' not in 'python' # True

>>> pocket = ['paper', 'cellphone', 'money']
>>> if 'money' in pocket:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
```

in	not in
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플
x in 문자열	x not in 문자열

```
if 'money' in pocket:
    print("택시를 타고 가라")
else:
    print("걸어가라")
```

조건문(if): 조건문에서 아무 일도 하지 않게 설정하고 싶다면

- "주머니에 돈이 있으면 가만히 있고 주머니에 돈이 없으면 카드를 꺼내라."

```
>>> pocket = ['paper', 'money', 'cellphone']
```

```
>>> if 'money' in pocket:
```

```
...     pass
```

```
... else:
```

```
...     print("카드를 꺼내라")
```

```
...
```


조건문(elif)

```
>>> weather = "맑아요"
```

```
>>> if weather == "비":  
    print("우산을 챙기세요")  
elif weather == "미세먼지":  
    print("마스크를 챙기세요")  
else:  
    print("준비물 필요 없어요")
```

```
if weather == "비":  
    print("우산을 챙기세요")  
elif weather == "미세먼지":  
    print("마스크를 챙기세요")  
else:  
    print("준비물 필요 없어요")
```

```
if <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>
```

```
...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>
```

```
...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>
```

```
...  
...  
else:  
    <수행할 문장1>  
    <수행할 문장2>
```

조건문(한줄쓰기)

```
>>> if 'money' in pocket:  
...     pass  
... else:  
...     print("카드를 꺼내라")
```

```
>>> if 'money' in pocket: pass  
... else: print("카드를 꺼내라")
```

```
>>> if score >= 60:  
...     message = "success"  
... else:  
...     message = "failure"...
```

```
>>> message = "success" if score >= 60 else "failure"
```

예제

다음 코드의 결과 값은 무엇일까요?

```
a = "Life is too short, you need python"
```

```
if "wife" in a: print("wife")
```

```
elif "python" in a and "you" not in a: print("python")
```

```
elif "shirt" not in a: print("shirt")
```

```
elif "need" in a: print("need")
```

```
else: print("none")
```

실습

- 기온을 입력 받아서 조건에 따른 처리하기
- 먼저 사용자로부터 정수 형태의 기온을 받는다고 가정
- 35, 20, 5, -20 을 순서대로 입력
- 30 도 이상이면 `print("너무 더워요. 나가지 마세요")`
- 10도 이상 30도 미만이면 `print("괜찮은 날씨예요")`
- 0도 이상 10도 미만이면 `print("외투를 챙기세요")`
- 0도 미만이면 `print("너무 추워요. 나가지 마세요")`

for문의 기본 구조

for 변수 in 반복대상:

실행 명령문1

실행 명령문2 ...

반복대상: 리스트 튜플, 문자등

```
>>> test_list = ['one', 'two', 'three']
```

```
>>> for i in test_list:
```

```
...     print(i)
```

```
...
```

```
one
```

```
two
```

```
three
```

```
>>> a = [(1,2), (3,4), (5,6)]
```

```
>>> for (first, last) in a:
```

```
...     print(first + last)
```

```
...
```

```
3
```

```
7
```

```
11
```

간단한 for 문

```
for waiting_no in [0, 1, 2, 3, 4]:
```

```
    print("대기번호 : {0}".format(waiting_no))
```

```
for waiting_no in range(5): # 0부터 5직전까지 (0~4)
```

```
    print("대기번호 : {0}".format(waiting_no))
```

```
for waiting_no in range(1, 6): # 1부터 6직전까지 (1~5)
```

```
    print("대기번호 : {0}".format(waiting_no))
```

```
starbucks = ["아이언맨", "토르", "아이엠 그루트"] # 손님 리스트
```

```
for customer in starbucks: print("{0}, 커피가 준비되었습니다".format(customer))
```

실습

- for와 range를 이용하여 구구단을 출력하시오

실습

- 총 5명의 학생이 시험을 보았는데 시험 점수가 60점이 넘으면 합격이고 그렇지 않으면 불합격이다. 합격인지 불합격인지 결과를 보여 주시오.
- marks = [90, 25, 67, 45, 80]
- 즉 marks 점수를 차례로 검사해서 합격했는지 불합격했는지 통보해 주는 프로그램을 만들어 봅시다.
- 결과:
 - 1번 학생은 합격입니다.
 - 2번 학생은 불합격입니다.
 - 3번 학생은 합격입니다.
 - 4번 학생은 불합격입니다.
 - 5번 학생은 합격입니다.

for문과 continue

for문 실습을 그대로 사용해서 60점 이상인 사람에게는 축하 메시지를 보내고 나머지 사람에게는 아무 메시지도 전하지 않는 프로그램을 에디터를 사용해 작성

```
marks = [90, 25, 67, 45, 80]
```

```
number = 0
```

```
for mark in marks:
```

```
    number = number + 1
```

```
    if mark < 60:
```

```
        continue
```

```
    print("%d번 학생 축하합니다. 합격입니다. " % number)
```

for문과 enumerate

enumerate: list의 멤버와 인덱스를 함께 호출

```
marks = [90, 25, 67, 45, 80]
```

```
number = 0
```

```
For i, mark in enumerate(marks):
```

```
if mark >= 60:
```

```
    print("%d번 학생은 합격입니다." %i)
```

```
else:
```

```
    print("%d번 학생은 불합격입니다." %i)
```

While문

while <조건문>:

<수행할 문장1>

<수행할 문장2>

<수행할 문장3>

while문은 조건문이 참인 동안에 while문 아래의 문장이 반복해서 수행된다.

While문 예제

어느 카페에서는 직원이 손님을 5번이나 불렀는데 나타나지 않는다고 하면 그 커피는 바로 폐기처분을 합니다.

while 을 사용해서 이 동작을 코드로 작성해 봅시다.

```
customer = "고객님" # 손님
```

```
index = 5 # 부르는 횟수, 총 5회
```

```
while index >= 1: # 부르는 횟수가 1 이상인 경우에만 반복 실행
```

```
    print("{0}, 커피가 준비 되었습니다. {1} 번 남았어요.".format(customer, index))
```

```
    index -= 1 # 부르는 횟수 감소
```

```
if index == 0: # 5번 모두 불렀다면
```

```
    print("커피는 폐기 처분되었습니다.")
```

While문 예제:비교해 봅시다.

```
customer = "아이언맨"
```

```
index = 1
```

```
while True:
```

```
    print("{0}, 커피가 준비 되었습니다. 호출 {1} 회".format(customer, index))
```

```
    index += 1
```

While문 예제: 비교해 봅시다.

```
customer = "토르"
```

```
person = "Unknown"
```

```
while person != customer:
```

```
    print("{0}, 커피가 준비 되었습니다.".format(customer))
```

```
    person = input("이름이 어떻게 되세요? ")
```

continue and break

한 교과목에서 프로젝트 발표를 한다고 합니다. 학번 1번부터 10번까지 총 10명의 학생들이 있고 순서대로 발표를 진행하는데 2번과 5번 학생들이 결석을 했다고 합니다. 그럼 1번 학생이 발표 후 2번을 건너뛰고 3, 4번 학생이 발표를 진행하고 5번을 건너뛰고 6번 학생이 발표를 진행해야 합니다.

```
absent = [2, 5] # 결석한 학생 출석번호
```

```
for student in range(1, 11): # 출석번호 1~10번
    if student in absent: # 결석했으면 다음 학생으로 넘어가기
        continue
    print("{0}, 발표하세요".format(student))
```

continue and break

```
absent = [2, 5] # 결석한 학생 출석번호
no_project = [7] # 프로젝트 준비 안 한 학생
for student in range(1, 11): # 출석번호 1~10번
    if student in absent: # 결석했으면 다음 학생으로 넘어가기
        continue
    elif student in no_project: # 프로젝트 준비 안 했으면 수업 종료 (반복문 탈출)
        print("오늘 수업 여기까지. {0}는 점수 없습니다.".format(student))
        break
    print("{0}, 발표하세요 ".format(student))
```


실습

10진수의 숫자를 다른 진법의 수로 변환하려고 합니다.

제어문을 이용하여 함수를 완성하세요.

```
def convert_from_decimal(number, base):
```

```
    multiplier, result = 1, 0
```

```
    ...
```

```
    ...
```

```
    return result
```

```
print(convert_from_decimal(9,2))
```

```
1001
```

실습

python coffee.py 를 실행하면 "돈을 넣어 주세요: " 가 뜨고

500원 300원 100원을 입력할 때마다 다음과 같은 결과를 출력합니다. 커피를 10잔까지 생산하는 자판기를 실행하는 python coffee.py 파일을 작성하세요.

결과:

돈을 넣어 주세요: 500

거스름돈 200를 주고 커피를 줍니다.

돈을 넣어 주세요: 300

커피를 줍니다.

돈을 넣어 주세요: 100

돈을 다시 돌려주고 커피를 주지 않습니다.

남은 커피의 양은 8개입니다.

돈을 넣어 주세요:

실습

- 어떤 단어가 주어지면 그 단어의 길이만큼 빈 칸(밑줄) 이 주어지고 플레이어는 알파벳을 하나씩 얘기하면서 단어를 추측합니다. 10번 시도 후 맞히지 못하면 게임이 끝납니다.
- 커피(COFFEE)라는 단어가 주어진다면, 밑줄은 총 6개가 생깁니다.
- 처음으로 E 를 해볼까요?
만약 E 가 단어에 포함돼있다면, 해당하는 위치에 E 를 표시해줍니다. (_ _ _ E E)
- 이번에는 A 를 해볼까요? A 는 포함되지 않으므로 기회가 줄어들고 지금까지 맞힌 내용 그대로 출력합니다.
- 이번에는 C 를 해봅시다. (C _ _ _ E E)
- 다음에는 F 를 (C _ F F E E)
- 마지막으로 O 를 해보면 (C O F F E E) 로 게임은 성공으로 끝나게 됩니다.

실습

- 리스트에 3개 이상의 단어 추가
- 리스트에서 랜덤으로 1개의 단어를 선택
- 단어의 길이에 맞게 밑줄 출력
- 사용자로부터 1글자씩 입력 받되 단어에 입력 값이 포함되면 'correct' , 아니면 'wrong' 출력
- 매번 입력 받을 때 마다 현재까지 맞힌 글자 표시
- 정답을 맞히면 success 출력 후 종료(횟수 제한 없음)

실습

- 당신은 음식점의 계산을 도와주는 점원입니다. 카운터에는 거스름돈으로 사용할 500원, 100원, 50원, 10원짜리 동전이 무한히 존재한다고 가정합니다. 손님에게 거슬러 주어야 할 돈이 N 원일 때 거슬러 주어야 할 동전의 최소 개수를 구하세요. 단, 거슬러 줘야 할 돈 N 은 항상 10의 배수입니다.
- $N=1260$ 일 때, 6이 출력되도록 코드를 완성 하시오.