

피라미드 만들어보기(for문 사용)

In [2]: *# 여기 셀에 코드 작성하지말고 밑에 셀에서 작성해주세요*

```
*
***
*****
*****
```

In [7]: `for i in range(1, 5):`
 `print(" " * (5-i), "*" * (2*i-1))`

```
*
***
*****
*****
```

In [9]: *# 모범답안: python index의 range가 0부터 쓰는 습관*

`for i in range(0, 5):`
 `print(" " * (5-i), "*" * (2*i-1))`

```
*
***
*****
*****
```

In [27]: `n = int(input('Pyramid height: '))`
`for i in range(1, n+1):`
 `print(" " * (n+1-i), "*" * (2*i-1))`

Pyramid height: 5

```
*
***
*****
*****
*****
```

피라미드 만들어보기(for문, if문 사용)

for문 1번만 사용

In [7]: *# 여기 셀에 코드 작성하지말고 밑에 셀에서 작성해주세요*

```
*
***
*****
*****
*****
*****
***
*
```

```
In [73]: # for문 두 번 사용
# width에 맞게 n 조절 실패
n = int(input('Pyramid width/2: '))
if n % 2 == 0:
    for i in range(1, (n+1)):
        print("*" * (2*i-1))
    for i in range(n, 0, -1):
        print("*" * (2*i-1))
else:
    for i in range(1, n):
        print("*" * (2*i-1))
    for i in range(n, 0, -1):
        print("*" * (2*i-1))
```

```
Pyramid width/2: 4
*
***
*****
*****
*****
***
*
```

```
In [50]: # for문 한 번 사용
n = int(input('Pyramid width: '))
for i in range(1, n*2+1, 2):
    if i <= n:
        print("*" * i)
    else:
        print("*" * (n*2-i))
```

```
Pyramid width: 8
*
***
*****
*****
*****
***
*
```

```
In [1]: # 모범답안

for i in range(8):
    if i < 4:
        print('*' * (2*i+1))
    else:
        print('*' * (2*(7-i) + 1))
```

```
*
***
*****
*****
*****
***
*
```

1~1000 사이의

3의 배수, 4의 배수 list 자료구조 각각 만들어보기

```
In [29]: # 모범답안과 동일

mul_3 = [x for x in range(3, 1001, 3)]
mul_4 = [y for y in range(4, 1001, 4)]
```

```
In [4]: # 추가 의견

multiple_3 = [x for x in range(3, 1001) if x % 3 == 0]
multiple_4 = [x for x in range(1, 1001) if x % 4 == 0]
```

위의 2가지 list자료구조에서 공통으로 포함된 숫자 list에 담아서 정렬하기

HINT : 다른 자료구조를 통해서 공통집합을 찾고 list()함수를 이용하여 list 자료구조로 변환

```
In [105]: mul_12 = list(set(mul_3).intersection(mul_4))
mul_12.sort(reverse=False)
print(mul_12)

'''
# list item을 dictionary로 변환
set3 = dict(zip(range(len(mul_3)), mul_3))
set4 = dict(zip(range(len(mul_4)), mul_4))
# set 자료구조를 통해 중복된 값 제거 - 교집합
set3 & set4
'''
```

```
[12, 24, 36, 48, 60, 72, 84, 96]
```

```
In [ ]: # 모범답안

multiple_12 = set(multiple_3).intersection(set(multiple_4))
list(multiple_12)
# list 자료구조로 바뀐 것이 아니라 리스트로 출력만 한 것.
multiple_12.sort
```

```
In [ ]: multiple_12 = list(multiple_12)
# Out: 변수 입력된 부분을 보여주는 것
multiple_12
# 출력: enter 없이
print(multiple_12)
```

아래와 같은 student dictionary자료구조가 있을때 age의 총합 구해보기

```
In [12]: student = {1: {'name': 'a', 'age': 27},
                    2: {'name': 'b', 'age': 25},
                    3: {'name': 'c', 'age': 31},
                    4: {'name': 'd', 'age': 22},
                    5: {'name': 'e', 'age': 15}}
```

```
In [17]: sum_age = 0

for i in range(1, len(student)+1):
    sum_age += student[i]['age']
print('The total age of all students is', sum_age, '.')
```

The total age of all students is 120 .

```
In [ ]: '''
{key: value}
{key: {key: value}}
{key: {key: {key: value}}}
'''
```

```
In [19]: # 모범답안

sum_age = 0

for i in student.keys():
    sum_age += student[i]['age']
```