



## 연산자, 변수, 자료형을 배우고 "인상적인 내용"에 대한 우리들의 이야기!

변수지정 할 때 그 변수의 종류를 지정해주지 않아도 된다는 점

파이썬과 c언어의 공통점과 차이점에 대해 잘 이해할 수 있었다

정수형 실수형 문자형을 입력하지않아도 자동으로 된다는 점

글자를 합하는게 인상적이었어요

변수형태를 생각할 필요없이 변수에 값을 대입할 수 있었던 것, 너무 편해서

파이썬이 처음이여서 그런지 모든 게 다 신기했다

거북이로 그림그리기

거북이

단위환산

다른 언어와 달리 변수를 선언할 때 자료형을 입력하지 않았던 것이 인상적이었다.

프로그래밍을 본격적으로 배우게 되어서 의미있었던 수업이었던 것 같다.

등식 쓰면 뜨는 게 가사쓰면 되서 좋았다.

거북이 - 별 수고 없이 도형을 그릴 수 있다는 점이 인상적이었음

처음에는 재미있었는데 갈수록 빨라지는 듯한...

## 연산자, 변수, 자료형을 배우고 "인상적인 내용"에 대한 우리들의 이야기!

변수지정 할 때 그 변수의 종류를 지정해주지 않아도 된다는 점

파이썬과 c언어의 공통점과 차이점에 대해 잘 이해할 수 있었다

정수형 실수형 문자형을 입력하지않아도 자동으로 된다는 점

글자를 합하는게 인상적이었어요

변수형태를 생각할 필요없이 변수에 값을 대입할 수 있었던 것, 너무 편해서

파이썬이 처음이여서 그런지 모든 게 다 신기했다

거북이로 그림그리기

거북이

단위환산

다른 언어와 달리 변수를 선언할 때 자료형을 입력하지 않았던 것이 인상적이었다.

프로그래밍을 본격적으로 배우게 되어서 의미있었던 수업이었던 것 같다.

등식 쓰면 뜨는 게 가사쓰면 되서 좋았다.

거북이 - 별 수고 없이 도형을 그릴 수 있다는 점이 인상적이었음

처음에는 재미있었는데 갈수록 빨라지는 듯한...

## 연산자, 변수, 자료형을 배우고 "인상적인 내용"에 대한 우리들의 이야기!

변수지정 할 때 그 변수의 종류를 지정해주지 않아도 된다는 점

파이썬과 c언어의 공통점과 차이점에 대해 잘 이해할 수 있었다

정수형 실수형 문자형을 입력하지않아도 자동으로 된다는 점

글자를 합하는게 인상적이었어요

변수형태를 생각할 필요없이 변수에 값을 대입할 수 있었던 것, 너무 편해서

파이썬이 처음이여서 그런지 모든 게 다 신기했다

거북이로 그림그리기

거북이

단위환산

다른 언어와 달리 변수를 선언할 때 자료형을 입력하지 않았던 것이 인상적이었다.

프로그래밍을 본격적으로 배우게 되어서 의미있었던 수업이었던 것 같다.

등식 쓰면 뜨는 게 가사쓰면 되서 좋았다.

거북이 별 수고 없이 도형을 그릴 수 있다는 점이 인상적이었음

처음에는 재미있었는데 갈수록 빨라지는 듯한...

**#논리적인 거북이** 🐢



사용자로부터 문자(r,g,b 중 하나)를 입력받아서  
입력받은 문자에 따라  
색을 다르게 사각형을 그리는 거북이를 만들어보자.

**r**을 입력하면 "**빨간색**"

**g**를 입력하면 "**초록색**"

**b**를 입력하면 "**파란색**"



## [논리적인 거북이 - solution]

#문자를 입력받는다

```
color=turtle.textinput("", "문자입력 (r,g,b 중 하나) :")
```

#문자의 입력 값에 따라 다른 색깔로 표현한다.

```
if (color=='r') :
```

```
    t.color("red")
```

```
    t.pencolor("red")
```

```
elif (color=='g') :
```

```
    t.color("green")
```

```
    t.pencolor("green")
```

```
elif (color=='b') :
```

```
    t.color("blue")
```

```
    t.pencolor("blue")
```

```
else :
```

```
    t.write("잘못된 입력값입니다")
```

```
    t.write("검정색으로 그립니다")
```



## [논리적인 거북이 - solution]

#문자를 입력받는다

```
color=turtle.textinput("", "문자입력 (r,g,b 중 하나) :")
```

#문자의 입력 값에 따라 다른 색깔로 표현한다.

```
if (color=='r') :  
    t.color("red")  
    t.pencolor("red")  
elif (color=='g') :  
    t.color("green")  
    t.pencolor("green")  
elif (color=='b') :  
    t.color("blue")  
    t.pencolor("blue")  
else :  
    t.write("잘못된 입력값입니다")  
    t.write("검정색으로 그립니다")
```

#사각형 그리기

```
t.fd(100)  
t.left(90)  
t.fd(100)  
t.left(90)  
t.fd(100)  
t.left(90)  
t.fd(100)  
t.left(90)
```



## [논리적인 거북이 - solution]

### #사각형 그리기

```
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
```



뭔가 아쉽다....

## [논리적인 거북이 - solution]

### #사각형 그리기

```
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
```



뭔가 아쉽다....

우리는 **반복**하고 있다..!

**#1. while**

**#2. for**

**#3. continue, break**

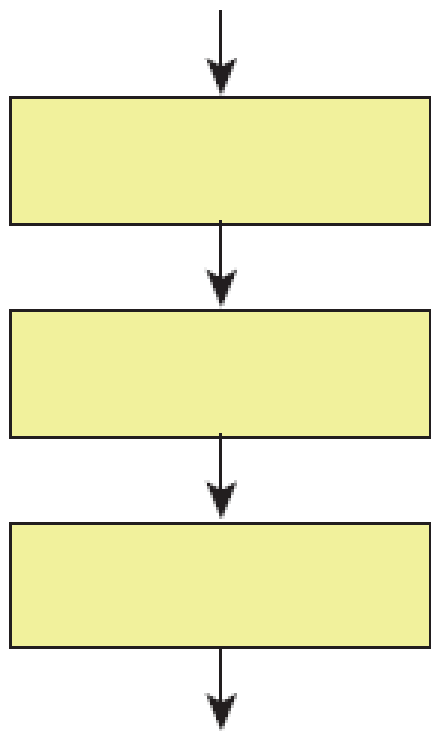
**#4. 369 프로그램 만들기**

**#5. 분식집 차림표 프로그램 만들기 Ver2**

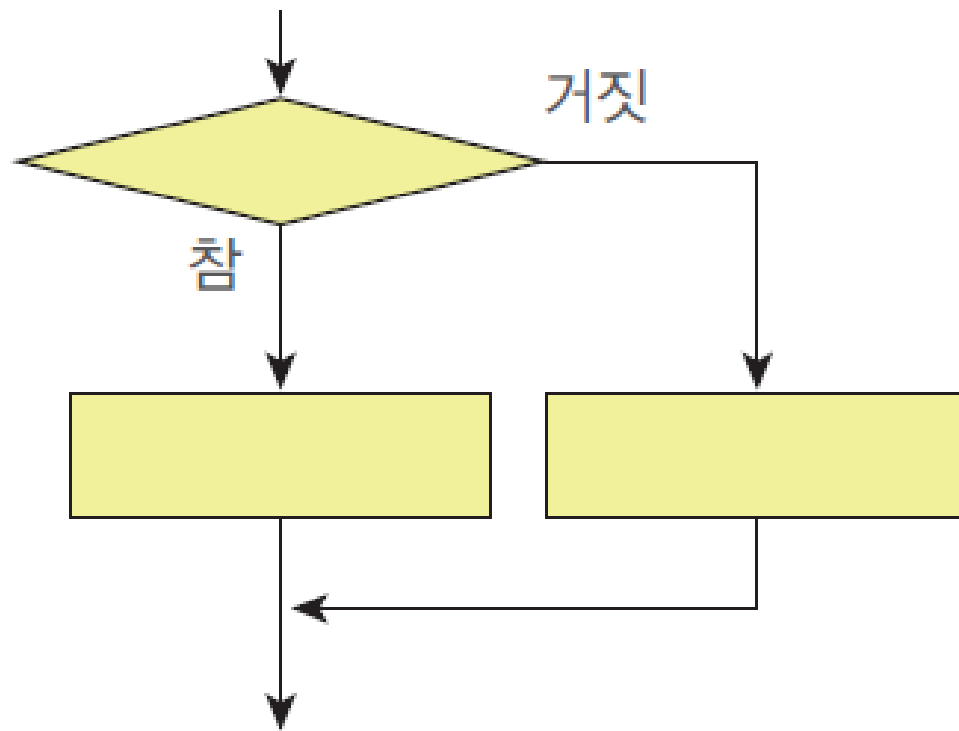
# #1. while문

# [순차, 선택, 반복 구조]

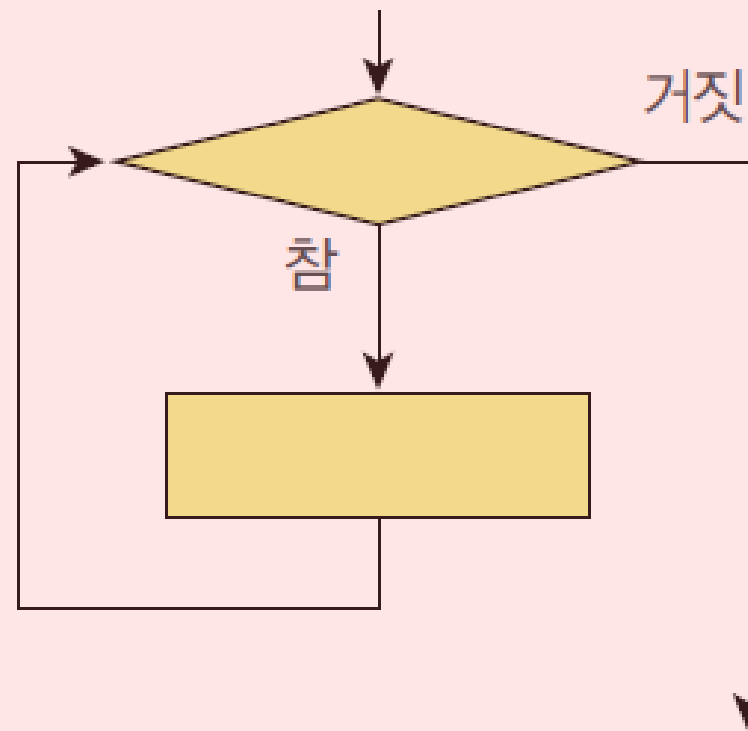
순차구조



선택구조



반복구조



# [while]

#조건식이 True인 동안, 코드를 반복해서 실행한다.

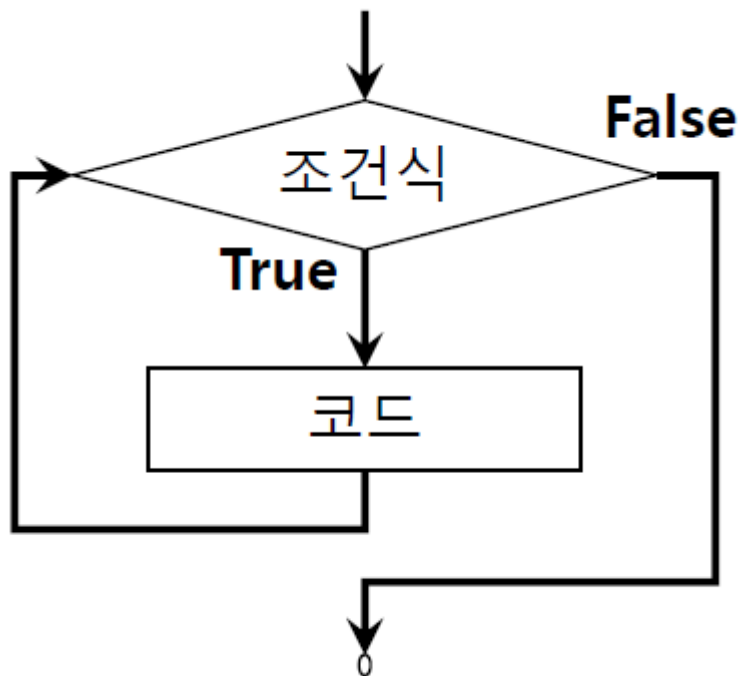
**while (조건식) :**  
**코드**



## 들여쓰기!!(Indentation)

파이썬은 들여쓰기 기준으로  
코드 블록을 구성한다.

**공백 4칸**을 기준으로 하자!



## [while]

```
In [1]: reply=input('##'y##'또는##'n##'을 입력 : ')
while(reply!='n') :
    print("repeat...")
    reply=input('##'y##'또는##'n##'을 입력 : ')
```

'y'또는'n'을 입력 : y

repeat...

'y'또는'n'을 입력 : y

repeat...

'y'또는'n'을 입력 : n

## [문제]

**'열 번 찍어 안 넘어가는 나무는 없다'를  
프로그램으로 작성해 보세요!**



# [문제 : 열 번 찍어 안 넘어가는 나무 없다-solution]

```
In [3]: treeHit=0
while(treeHit<10) :
    treeHit=treeHit+1
    print("나무를 %d번 찍었습니다"%treeHit)
    if(treeHit==10) :
        print("나무가 넘어갑니다~~~")
```

나무를 1번 찍었습니다  
나무를 2번 찍었습니다  
나무를 3번 찍었습니다  
나무를 4번 찍었습니다  
나무를 5번 찍었습니다  
나무를 6번 찍었습니다  
나무를 7번 찍었습니다  
나무를 8번 찍었습니다  
나무를 9번 찍었습니다  
나무를 10번 찍었습니다  
나무가 넘어갑니다~~~

#논리적인 거북이

## [논리적인 거북이 아쉬움 해결하기]

### #사각형 그리기

```
t.fd(100)
```

```
t.left(90)
```

```
t.fd(100)
```

```
t.left(90)
```

```
t.fd(100)
```

```
t.left(90)
```

```
t.fd(100)
```

```
t.left(90)
```



뭔가 아쉽다....

우리는 **반복**하고 있다..!

#논리적인 거북이

## [논리적인 거북이 아쉬움 해결하기-solution]

### #사각형 그리기

```
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
t.fd(100)
t.left(90)
```

### #반복문을 사용하여 사각형 그리기

```
i=1
while (i<=4):
    t.fd(100)
    t.left(90)
    i=i+1
```

## [문제 : yes인 동안 숫자 입력받기]

사용자가 입력한 숫자들을 더하는 프로그램을 작성해 보세요.

단, 사용자가 yes라고 입력한 동안에만 입력 받으세요

```
숫자를 입력하시오: 10  
계속 입력하시겠습니까?(yes/no):yes  
숫자를 입력하시오: 20  
계속 입력하시겠습니까?(yes/no):no  
합계는 : 30
```

**#2. for**

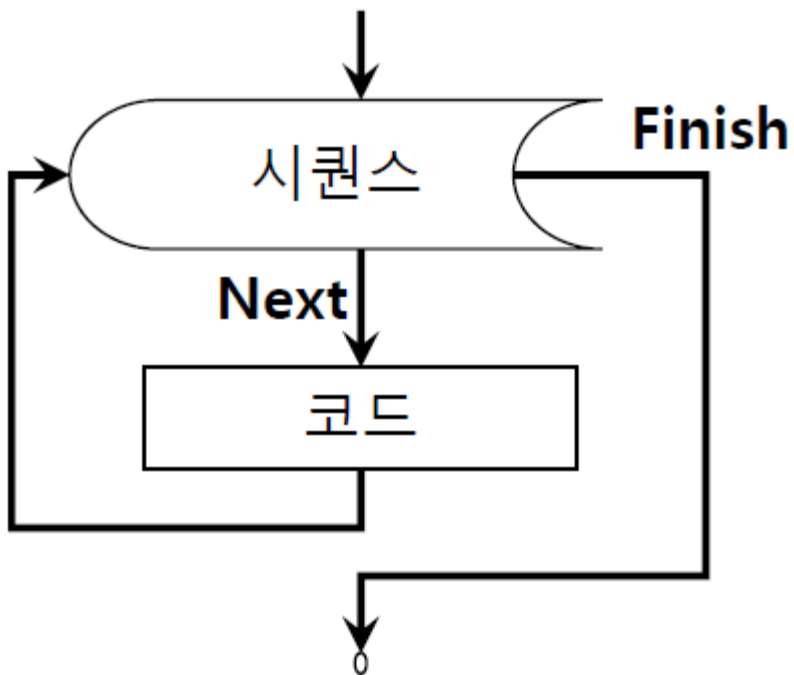
# [for]

#범위 안의 내용을 차례대로 변수에 대입되어 코드를 반복 수행한다.

**for** 변수 **in** 범위:  
코드



범위로 사용되는 것은  
**시퀀스 자료형** 또는 **반복가능한 자료**  
여야 한다!



In [3]:

```
str="KyungBoak"  
for i in str :  
    print(i,end=" ")
```

K y u n g B o a k

## [for문의 범위로 사용되는 것들]

#문자열

#리스트

#튜플

#딕셔너리

#range( )

#그 외 반복 가능한 객체

## [for문의 범위로 사용되는 것들]

```
In [1]: #문자열
str='abcdef'
for c in str :
    print(c,end=" ")
```

a b c e d f

```
In [2]: #리스트
list=[1,2,3,4,5]
for item in list :
    print(item,end=" ")
```

1 2 3 4 5

```
In [3]: #튜플
tuple=(1,2,3)
for item in tuple :
    print(item,end=" ")
```

1 2 3

```
In [4]: #딕셔너리
ascii_codes={'a':97, 'b':98, 'c':99}
for c in ascii_codes.values() :
    print(c,end=" ")
print()
for c in ascii_codes.keys():
    print(c,end=" ")
```

97 98 99

a b c



# [for를 range( )와 함께 사용하기]

# range함수는 숫자 리스트를 자동으로 생성하여 준다.

# 0부터 숫자만큼 변수에 대입되어 코드를 반복 수행한다.

**for** 변수 **in** *range*(숫자):  
    코드



**0~숫자-1까지** 값이 생성된다!

숫자를 하나씩 꺼냄      숫자 100개 생성  
0, 1, 2, 3, 4 ... 97, 98, 99

```
for i in range(100):  
    print('Hello, world!')
```

숫자를 꺼낼 때마다  
코드 실행

출처 <https://dojang.io/mod/page/view.php?id=919>

## [for를 range( )와 함께 사용하기]

**for** 변수 **in** *range(start, stop-1)*:  
코드

**for** 변수 **in** *range(start, stop-1, step)*:  
코드

```
In [1]: for i in range(10) :  
        print(i,end=" ")
```

0 1 2 3 4 5 6 7 8 9

```
In [2]: for i in range(2,10) :  
        print(i, end=" ")
```

2 3 4 5 6 7 8 9

```
In [3]: for i in range(1,11,2) :  
        print(i, end=" ")
```

1 3 5 7 9

**[문제 : 홀수 출력하기]**

**500~1000까지의 수 중에서 홀수만 출력 해 주세요!**

**[문제 : 3의 배수 출력하기]**

**13~39까지의 수 중에서 3의 배수만 출력 해 주세요!**

**[문제 : 3의 배수 또는 5의 배수 출력하기]**

**13~39까지의 수 중에서**

**3의 배수이거나 5의 배수면 출력 해 주세요!**

**[문제 : 5의 배수의 합 구하기]**

**1~20까지의 수 중에서 5의 배수의 합을 출력 해 주세요!**

**[문제 : 학급의 중간고사 평균 구하기]**

**우리 반의 프로그래밍 중간고사 성적이 아래와 같네요.**

**[70, 60, 55, 75, 95, 90, 80, 80, 85, 100]**

**우리 반의 평균 점수를 구해 주세요~**

**[문제 : 짝수 출력하기]**

**2~100까지의 수 중에서 짝수만 출력 해 주세요!**



**[문제 : 1부터 사용자가 입력한 수까지 출력]**

**사용자로부터 숫자를 입력받아  
1부터 입력받은 수 까지 출력해 보세요.**

**[문제 : 구구단 출력]**

**사용자로부터 단을 입력받아  
입력받은 단의 구구단을 출력해 보세요**

### **#3. break, continue**

## [for~continue~break]

**for** 변수 **in** 범위:

.....

**continue**    *#다음 반복문 수행*

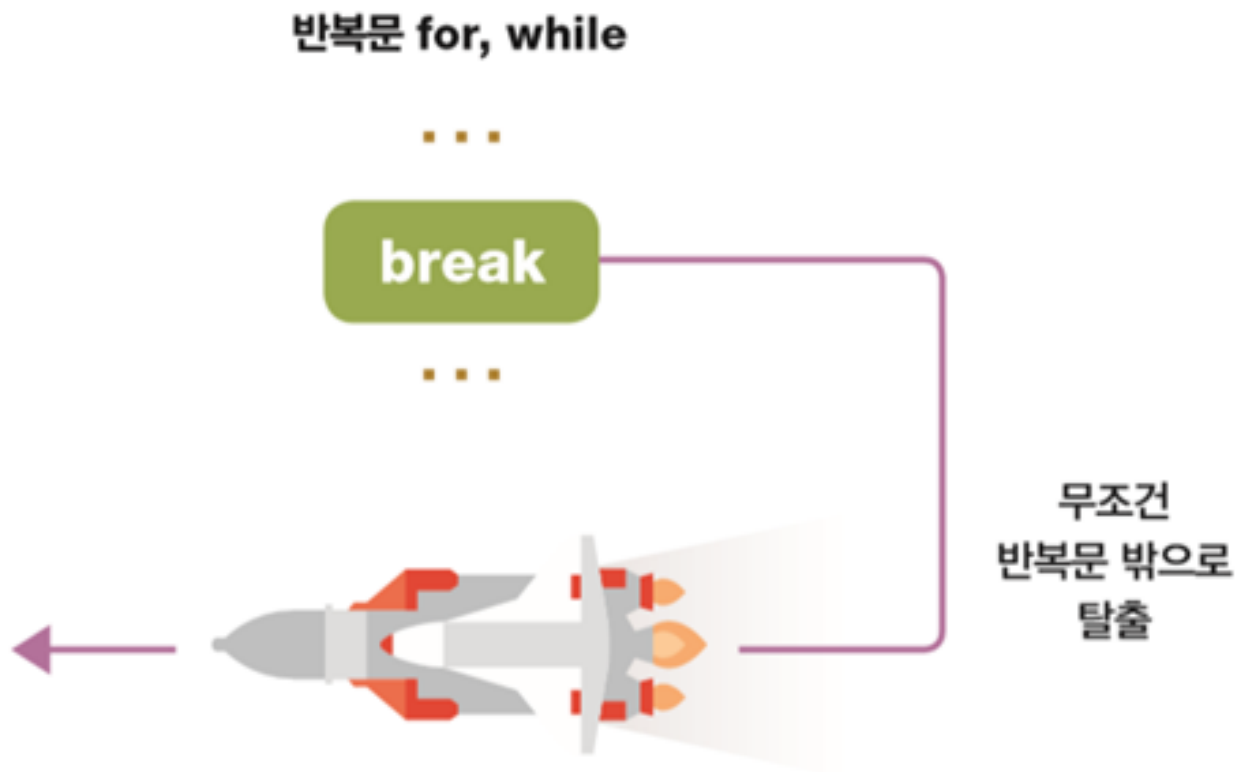
.....

**break**        *#for 반복문 탈출*

```
In [ ]: scope=[1,2,3,4,5]
        for x in scope :
            print(x)
            if x<3 :
                continue
            else :
                break
```

# [break]

#제어의 흐름을 중단하고 반복문을 빠져나온다.



```
In [20]: i = 0
while True:
    i += 1
    print(i)
    if i == 10:
        break
```

```
In [22]: for i in range(1, 100):
print(i)
    if i == 10:
        break
```

# [continue]

#제어의 흐름을 유지한 상태에서 코드의 실행을 건너뛴다

```
In [23]: for i in range(1, 11):  
          if i % 2 != 0:  
              continue  
          print(i)
```



```
In [24]: i = 1  
while i <= 10:  
    i += 1  
    if i % 2 != 0:  
        continue  
    print(i)
```

**[문제 : 1~입력 수 까지 4의 배수를 제외하고 더하기]**

**사용자로부터 숫자를 입력받아**

**1부터 입력받은 숫자까지 더하는 프로그램을 작성해 보세요.**

**단, 4의 배수는 제외하고 더하세요.**

## [for~else]

**for** 변수 **in** 범위:  
반복 실행 코드

**else :**  
for 구문이 모두 실행되었을 때 실행할 코드

```
scope=[1,2,3,4,5]
for x in scope :
    print(x)
    #if (x > 3) : break
else :
    print("Perfect!")
```

```
scope=[1,2,3,4,5]
for x in scope :
    print(x)
    if (x > 3) : break
else :
    print("Perfect!")
```



## **#4. 369프로그램 만들기**

## #4. 369프로그램 만들기

[알고리즘 생각하기]

## #4. 369프로그램 만들기

1. 사용자로부터 숫자를 입력받는다(0을 입력하면 종료)
2. 숫자가 3의 배수면 "짝"을 출력한다
3. 숫자가 6의 배수면 "짝짝"을 출력한다
4. 숫자가 9의 배수면 "짝짝짝"을 출력한다
5. 숫자가 3, 6, 9의 배수가 모두 아니면 숫자를 출력한다.

## **#5. 분식집 차림표 만들기 Ver.2**

## #5. 분식집 차림표 만들기-Ver02

1. 분식집 메뉴와 가격을 결정한다.
2. 메뉴와 가격표 출력
3. 사용자로부터 선택한 메뉴를 입력받는다.
4. 사용자가 "0"을 누르기 전까지 메뉴를 반복 입력 받는다.
5. 사용자가 "0"을 누르면 메뉴 선택 종료
6. 선택한 메뉴와 개수, 총금액을 출력한다.

# [분식집 프로그램의 아쉬움 해결하기]

*a.k.a 처음이라 불리운 사나이, 마수빈*

## [수업 4차시-정리하기]

<https://goo.gl/forms/Pfu5CzpF9F7f5OjL2>