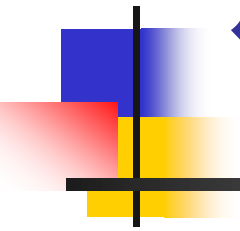
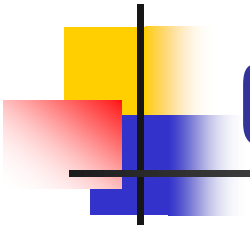


03장 프로그램의 구조를 쌓는다! 제어문

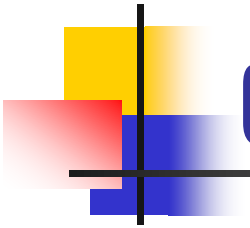
자료형을 바탕으로 제어문을 이용하여 프로그램의 구조를 만들어 보자



03-1 If 문

돈이 있으면 택시를 타고, 돈이 없으면 걸어 간다

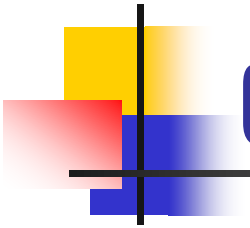
```
>>> money = 1
>>> if money:
...     print("택시를 타고 가라")
... else:
...     print("걸어 가라")
...
택시를 타고 가라
```



03-1 If 문

if문의 기본구조

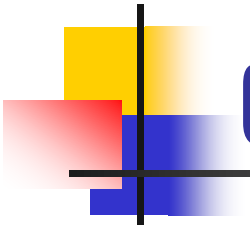
```
if 조건문:
    수행할 문장1
    수행할 문장2
    ...
else:
    수행할 문장A
    수행할 문장B
    ...
```



03-1 If 문

들여쓰기

```
if 조건문:
    수행할 문장1
    수행할 문장2
    수행할 문장3
```

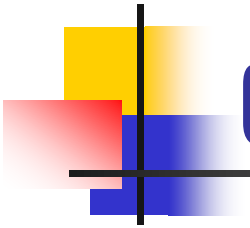


03-1 If 문

들여쓰기 오류

```
if 조건문:
    수행할 문장1
수행할 문장2
    수행할 문장3
```

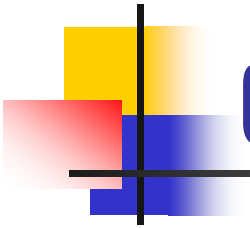
```
>>> if money:
...     print("택시를")
... print("타고")
File "<stdin>", line 4
print("가자")
^
SyntaxError: invalid syntax
```



03-1 If 문

조건문

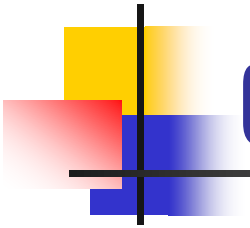
```
>>> money = 1
>>> if money:
```



03-1 If 문

비교 연산자

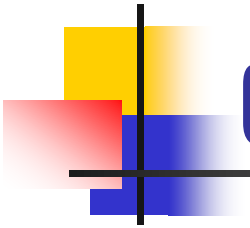
비교연산자	설명
x < y	x가 y보다 작다
x > y	x가 y보다 크다
x == y	x와 y가 같다
x != y	x와 y가 같지 않다
x >= y	x가 y보다 크거나 같다
x <= y	x가 y보다 작거나 같다



03-1 If 문

비교 연산자

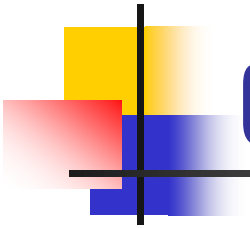
```
>>> x = 3
>>> y = 2
>>> x > y
True
>>> x < y
False
>>> x == y
False
>>> x != y
True
```



03-1 If 문

만약 3000원 이상의 돈을 가지고 있으면 택시를 타고 그렇지 않으면 걸어 가라

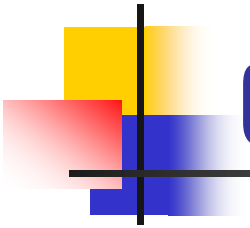
```
>>> money = 2000
>>> if money >= 3000:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
걸어가라
```



03-1 If 문

and, or, not

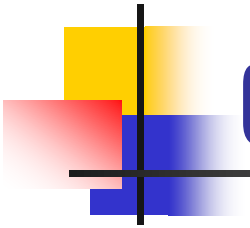
연산자	설명
x or y	x와 y 둘중에 하나만 참이면 참이다
x and y	x와 y 모두 참이어야 참이다
not x	x가 거짓이면 참이다



03-1 If 문

돈이 3000원 이상 있거나 카드가 있다면 택시를 타고 그렇지 않으면 걸어 가라

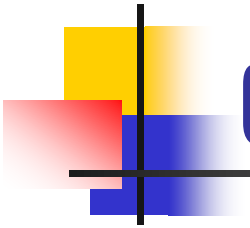
```
>>> money = 2000
>>> card = 1
>>> if money >= 3000 or card:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
```



03-1 If 문

x in s, x not in s

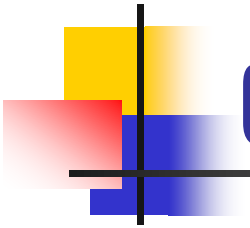
```
>>> 1 in [1, 2, 3]
True
>>> 1 not in [1, 2, 3]
False
```



03-1 If 문

만약 주머니에 돈이 있으면 택시를 타고, 없으면 걸어 가라

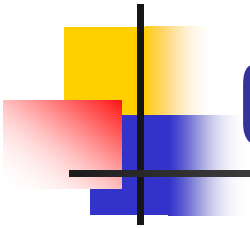
```
>>> pocket = ['paper', 'cellphone', 'money']
>>> if 'money' in pocket:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
>>>
```



03-1 If 문

조건문에서 아무 일도 하지 않게 설정하고 싶다면?

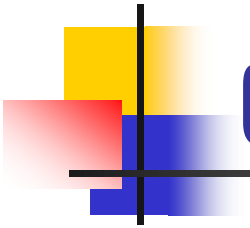
```
>>> pocket = ['paper', 'money', 'cellphone']
>>> if 'money' in pocket:
...     pass
... else:
...     print("카드를 꺼내라")
...
```



03-1 If 문

다중 조건 판단 elif

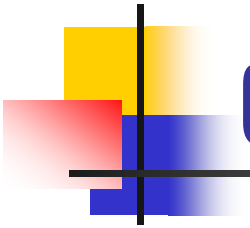
```
>>> pocket = ['paper', 'cellphone']
>>> card = 1
>>> if 'money' in pocket:
...       print("택시를 타고가라")
... elif card:
...       print("택시를 타고가라")
... else:
...       print("걸어가라")
...
택시를 타고가라
```



03-2 while 문

while문의 기본 구조

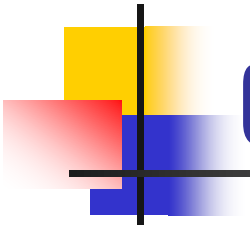
```
while <조건문>:
    <수행할 문장1>
    <수행할 문장2>
    <수행할 문장3>
    ...
```



03-2 while 문

열 번 찍어 안 넘어 가는 나무 없다

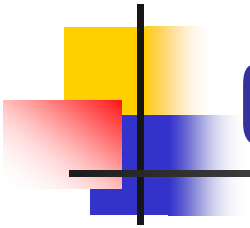
```
>>> treeHit = 0
>>> while treeHit < 10:
...     treeHit = treeHit +1
...     print("나무를 %d번 찍었습니다." % treeHit)
...     if treeHit == 10:
...         print("나무 넘어갑니다.")
...
나무를 1번 찍었습니다.
나무를 2번 찍었습니다.
나무를 3번 찍었습니다.
…
나무 넘어갑니다.
```



03-2 while 문

break

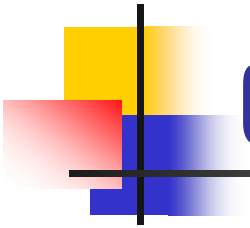
```
>>> coffee = 10
>>> money = 300
>>> while money:
...     print("돈을 받았으니 커피를 줍니다.")
...     coffee = coffee -1
...     print("남은 커피의 양은 %d개입니다." % coffee)
...     if not coffee:
...         print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
...         break
...
```



03-2 while 문

continue

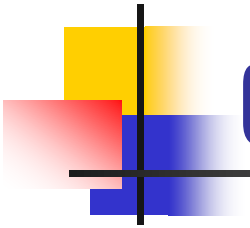
```
>>> a = 0
>>> while a < 10:
...     a = a+1
...     if a % 2 == 0: continue
...     print(a)
...
1
3
5
7
9
```



03-2 while 문

무한 루프

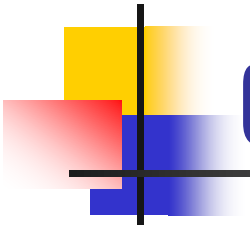
```
while True:
    수행할 문장1
    수행할 문장2
    ...
```



03-3 for 문

for문의 기본 구조

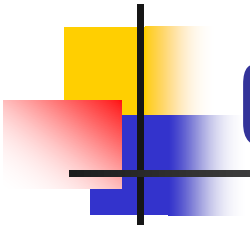
```
for 변수 in 리스트(또는 튜플, 문자열):
    수행할 문장1
    수행할 문장2
    ...
```



03-3 for 문

전형적인 for문

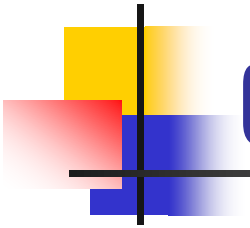
```
>>> test_list = ['one', 'two', 'three']
>>> for i in test_list:
...     print(i)
...
one
two
three
```



03-3 for 문

다양한 for문의 사용

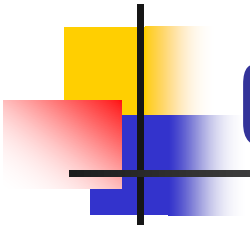
```
>>> a = [(1,2), (3,4), (5,6)]
>>> for (first, last) in a:
...     print(first + last)
...
3
7
11
```



03-3 for 문

60점이 넘으면 합격이고 그렇지 않으면 불합격

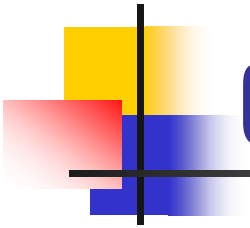
```
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number +1
    if mark >= 60:
        print("%d번 학생은 합격입니다." % number)
    else:
        print("%d번 학생은 불합격입니다." % number)
```



03-3 for 문

for문과 continue

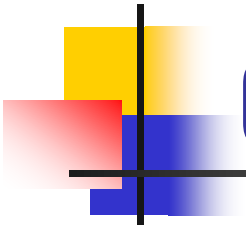
```
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number +1
    if mark < 60: continue
    print("%d번 학생 축하합니다. 합격입니다. " % number)
```



03-3 for 문

for와 함께 자주 사용하는 range함수

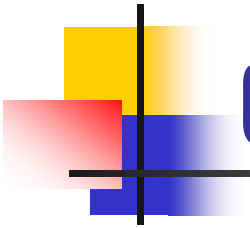
```
>>> sum = 0
>>> for i in range(1, 11):
...     sum = sum + i
...
>>> print(sum)
55
```



03-3 for 문

구구단

```
>>> for i in range(2,10):
...     for j in range(1, 10):
...         print(i*j, end=" ")
...     print('')
...
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
```



03-3 for 문

리스트 내포(List comprehension)

```
>>> result = [num * 3 for num in a]
>>> print(result)
[3, 6, 9, 12]
```

```
>>> result = [num * 3 for num in a if num % 2 == 0]
>>> print(result)
[6, 12]
```