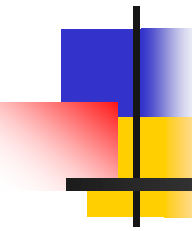
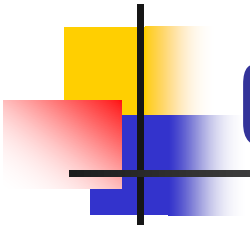


04장 프로그램의 입력과 출력은 어떻게 해야 할까?

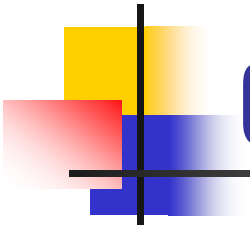
함수, 입력과 출력, 파일 처리 방법



04-1 함수

파이썬 함수의 구조

```
def 함수명(입력 인수):
    <수행할 문장1>
    <수행할 문장2>
    ...
```

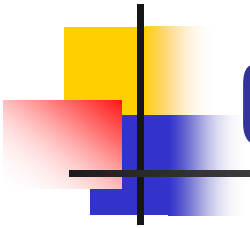


04-1 함수

일반적인 함수

```
def sum(a, b):
    result = a + b
    return result
```

```
>>> a = sum(3, 4)
>>> print(a)
7
```

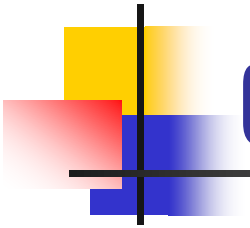


04-1 함수

입력값이 없는 함수

```
>>> def say():
...     return 'Hi'
...
>>>
```

```
>>> a = say()
>>> print(a)
Hi
```

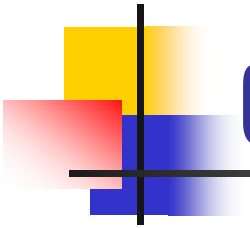


04-1 함수

결과값이 없는 함수

```
>>> def sum(a, b):
...     print("%d, %d의 합은 %d입니다." % (a, b, a+b))
...
>>>
```

```
>>> sum(3, 4)
3, 4의 합은 7입니다.
```

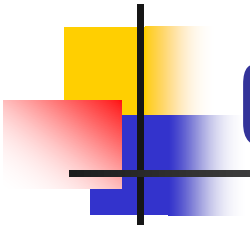


04-1 함수

입력값도 결과값도 없는 함수

```
>>> def say():
...     print('Hi')
...
>>>
```

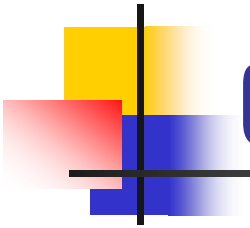
```
>>> say()
Hi
```



04-1 함수

여러 개의 입력값

```
>>> def sum_many(*args):
...     sum = 0
...     for i in args:
...         sum = sum + i
...     return sum
...
>>>
```



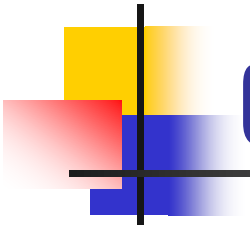
04-1 함수

함수의 결과값은 언제나 하나이다

```
>>> def sum_and_mul(a,b):
...     return a+b, a*b
```

```
>>> result = sum_and_mul(3,4)
```

```
result = (7, 12)
```

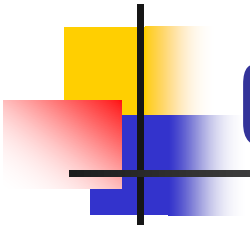


04-1 함수

매개 변수에 초깃값 미리 설정하기

```
def say_myself(name, old, man=True):
    print("나의 이름은 %s 입니다." % name)
    print("나이는 %d살입니다." % old)
    if man:
        print("남자입니다.")
    else:
        print("여자입니다.")
```

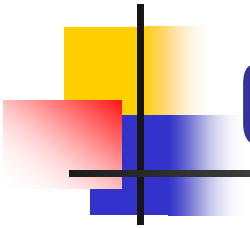
```
say_myself("박응용", 27)
say_myself("박응용", 27, True)
```



04-1 함수

함수 매개 변수에 초깃값을 설정할 때 주의할 사항

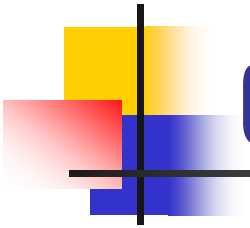
```
def say_myself(name, man=True, old):
    print("나의 이름은 %s 입니다." % name)
    print("나이는 %d살입니다." % old)
    if man:
        print("남자입니다.")
    else:
        print("여자입니다.")
```



04-1 함수

함수 안에서 선언된 변수의 효력 범위

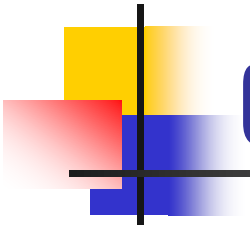
```
a = 1
def vartest(a):
    a = a +1

vartest(a)
print(a)
```



04-1 함수

함수 안에서 함수 밖의 변수를 변경하는 방법 1

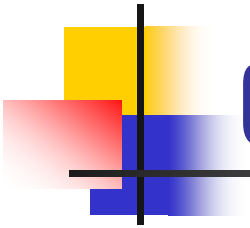
```
a = 1
def vartest(a):
    a = a +1
    return a

a = vartest(a)
print(a)
```



04-1 함수

함수 안에서 함수 밖의 변수를 변경하는 방법 2

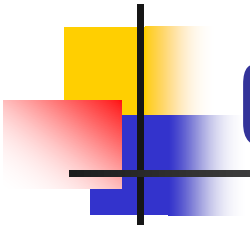
```
a = 1
def vartest():
    global a
    a = a+1

vartest()
print(a)
```



04-2 사용자 입력과 출력

input의 사용

```
>>> a = input()
Life is too short, you need python
>>> a
'Life is too short, you need python'
>>>
```



04-2 사용자 입력과 출력

프롬프트를 띄워서 사용자 입력 받기

```
>>> number = input("숫자를 입력하세요: ")
숫자를 입력하세요: 3
>>> print(number)
3
>>>
```

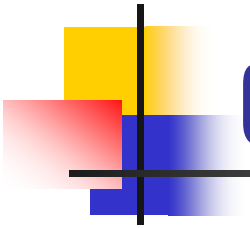
04-2 사용자 입력과 출력

print 문

```
>>> print("life" "is" "too short")
lifeistoo short
>>> print("life"+"is"+"too short")
lifeistoo short
```

```
>>> print("life", "is", "too short")
life is too short
```

```
>>> for i in range(10):
...     print(i, end=' ')
...
0 1 2 3 4 5 6 7 8 9
```

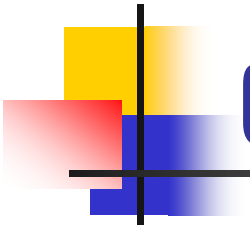


04-3 파일 읽고 쓰기

파일 생성하기

```
f = open("새파일.txt", 'w')
f.close()
```

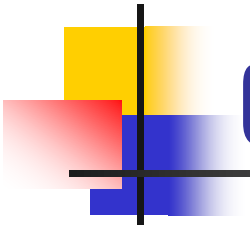
파일열기모드	설명
r	읽기모드 - 파일을 읽기만 할 때 사용
w	쓰기모드 - 파일에 내용을 쓸 때 사용
a	추가모드 - 파일의 마지막에 새로운 내용을 추가시킬 때 사용



04-3 파일 읽고 쓰기

파일을 쓰기 모드로 열어 출력값 적기

```
f = open("C:/Python/새파일.txt", 'w')
for i in range(1, 11):
    data = "%d번째 줄입니다.\n" % i
    f.write(data)
f.close()
```

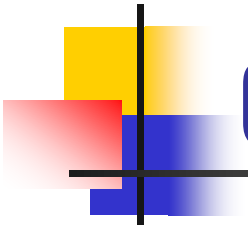


04-3 파일 읽고 쓰기

readline() 함수

```
f = open("C:/Python/새파일.txt", 'r')
line = f.readline()
print(line)
f.close()
```

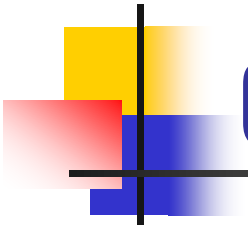
```
f = open("C:/Python/새파일.txt", 'r')
while True:
    line = f.readline()
    if not line: break
    print(line)
f.close()
```



04-3 파일 읽고 쓰기

readlines() 함수 이용하기

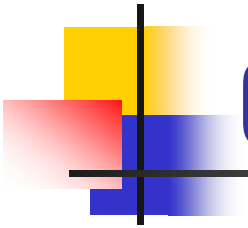
```
f = open("C:/Python/새파일.txt", 'r')
lines = f.readlines()
for line in lines:
    print(line)
f.close()
```



04-3 파일 읽고 쓰기

read() 함수 이용하기

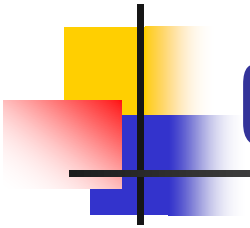
```
f = open("C:/Python/새파일.txt", 'r')
data = f.read()
print(data)
f.close()
```



04-3 파일 읽고 쓰기

파일에 새로운 내용 추가하기

```
f = open("C:/Python/새파일.txt",'a')
for i in range(11, 20):
    data = "%d번째 줄입니다.\n" % i
    f.write(data)
f.close()
```



04-3 파일 읽고 쓰기

with문과 함께 사용하기

```
f = open("foo.txt", 'w')
f.write("Life is too short, you need python")
f.close()
```

```
with open("foo.txt", "w") as f:
    f.write("Life is too short, you need python")
```