

데이터를
저장하여 처리하자

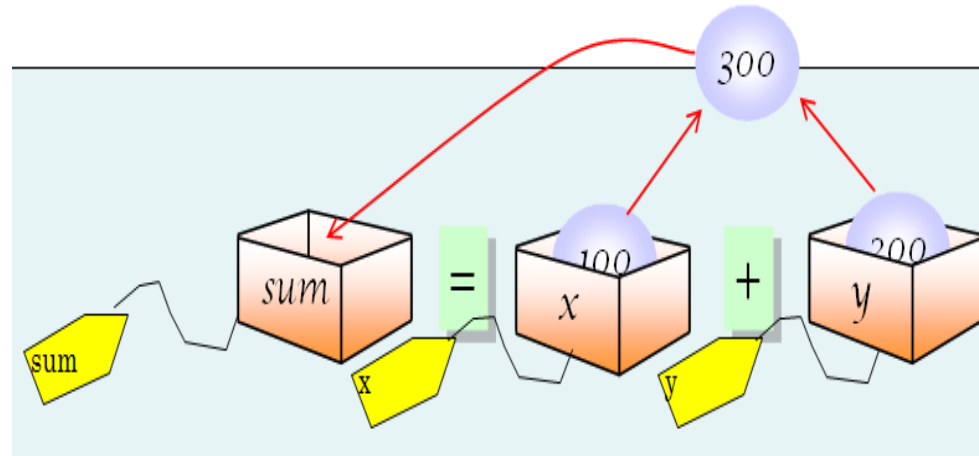
목차

1. 변수
2. 데이터 타입 (자료형)
3. input()
4. 데이터 타입 변환
5. 서식 지정자
6. 객체

변수

1. 데이터를 저장하는 공간
2. 변수는 컴퓨터의 메모리 공간에 이름을 붙이는 것
3. 변수에 저장된 값은 변경 가능

```
x = 100  
y = 200  
sum = x + y  
print(sum)
```



‘=’ 기호 : ‘오른쪽의 값을 왼쪽의 변수에 저장하라’는 의미.
이 연산자 = 를 할당연산자 또는 대입연산자라고 한다.

변수

동시 할당문 : 한 줄에 여러 개의 변수를 선언하고 이 변수에 값을 동시에 할당

```
a, b = 7, 19  
sum = a + b  
print(sum)
```

```
a, b = b, a      # a와 b의 값을 동시에 바꿈  
print(a)  
print(b)
```

변수 이름은 어떻게 짓나?

1. 변수 역할을 가장 잘 설명하는 이름을 지어야 한다. (프로그램 **가독성** 향상)
2. 영문자와 숫자, 밑줄 문자(_)로 이루어진다.
3. 식별자의 첫 글자는 숫자로 시작할 수 없다. 중간에 공백을 가질 수 없다.
4. 대문자와 소문자는 구별된다.
5. 파이썬의 예약어(키워드)는 식별자로 사용할 수 없다 (아래 표)

| | | | | | |
|--------|--------|----------|----------|---------|--------|
| False | class | return | is | finally | None |
| if | for | lambda | continue | True | def |
| from | while | nonlocal | and | del | global |
| not | with | as | elif | try | or |
| yield | assert | else | import | pass | break |
| except | in | raise | | | |

자료형(Data Type)

1. 정수^{integer}
2. 실수^{floating-point}
3. 문자열^{string}
4. 부울형^{bool} : 참(True) 또는 거짓(False)

| 자료형 | 예 |
|-----------|----------------------------------|
| 정수(int) | ..., -3, -2, -1, 0, 1, 2, 3, ... |
| 실수(float) | 3.14, 4.28, 0.01, 123.432 |
| 문자열(str) | 'Hello World', '123', "Hi" |
| 부울형(bool) | True, False |

`type()` : 변수의 데이터 타입 확인

```
print('='*15)
print(3.14, 56, 'hello', True)
print(type(3.14), type(56), type('hello'), type(True))
```

```
a = 3.14          # 대입 연산자(assignment)
print(365.1, a, type(a))
aa = 56
print(aa, type(aa))
```

```
↓ =====
⏮ 3.14 56 hello True
⏭ <class 'float'> <class 'int'> <class 'str'> <class 'bool'>
⏮ 365.1 3.14 <class 'float'>
⏭ 56 <class 'int'>
```

문자열과 문자열 인덱싱

문자로 이루어지 텍스트 데이터

큰 따옴표(“) 또는 작은 따옴표(‘)를 이용하여 생성

```
s="hello world"  
print(s)  
print(s[0])  
print(len(s))
```

| | | | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| h | e | l | l | o | | w | o | r | l | d |
| 0번 요소 | 1번 요소 | 2번 요소 | 3번 요소 | 4번 요소 | 5번 요소 | 6번 요소 | 7번 요소 | 8번 요소 | 9번 요소 | 10번 요소 |
| 변수명[0] | 변수명[1] | 변수명[2] | 변수명[3] | 변수명[4] | 변수명[5] | 변수명[6] | 변수명[7] | 변수명[8] | 변수명[9] | 변수명[10] |

자료형 의 변환 (데이터 형 변환)

1. 정수를 문자열로 변환하려면 `str()` 사용
2. 실수를 문자열로 변환하려면 `str()`을 사용
3. 문자열을 정수로 변환하려면 `int()`를 사용
4. 문자열을 실수로 변환하려면 `float()`를 사용

'100' → `int()` → 100
`int('100')`

100 → `str()` → '100'
정수를 문자열로 변환

'3.14' → `float()` → 3.14
문자열을 실수로 변환

3.14 → `str()` → '3.14'
실수를 문자열로 변환

input()

1. 사용자로부터(키보드로부터) 데이터 입력받기
2. 사용자의 입력을 무조건 **문자열** 형태로 반환

input() 사용법

변수

사용자가 입력한 문자열을 숫자로 변환한다.

```
x = int(input("첫 번째 정수를 입력하시오: "))
```

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.

```
name = input("이름을 입력하시오 :")
```

이름을 입력하시오 : 이순신

input() 사용 예

```
name = input("이름: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")
```

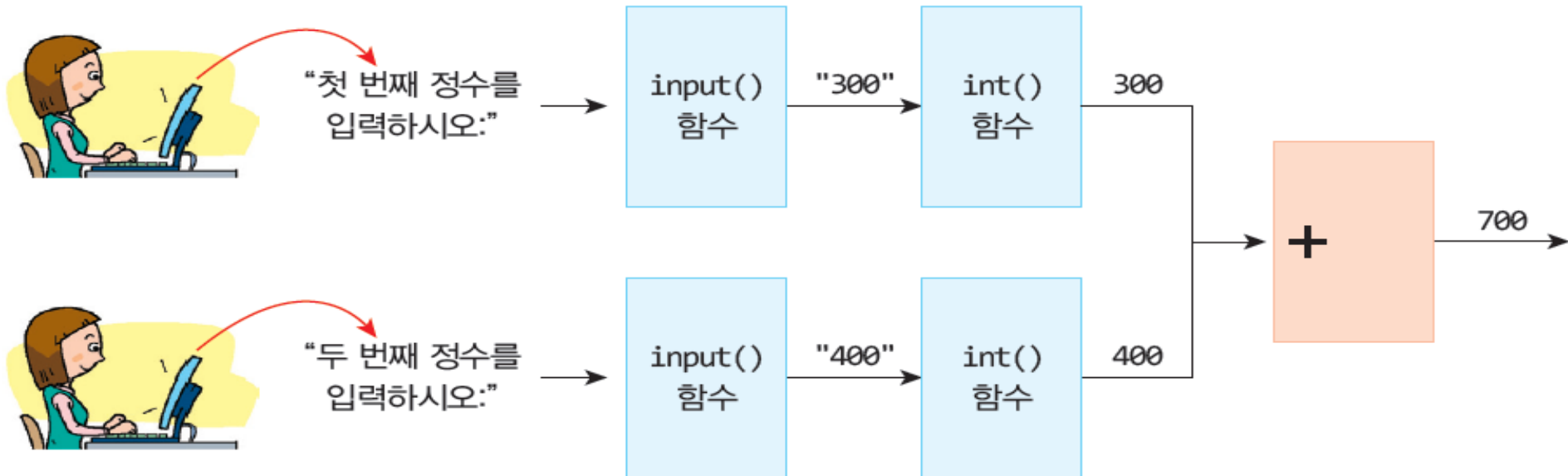
```
/////  
name = input("이름: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")  
/////
```

도전

사용자로부터 2개의 정수를 입력 받아 덧셈을 수행한 후,
그 결과를 출력하세요

도전

```
x = int(input("첫번째 정수를 입력하시오:"))  
y = int(input("두번째 정수를 입력하시오:"))  
sum = x + y  
print(x, "와", y, "의 합은", sum)
```



print()

| 함수이름 | 설명 |
|---------|---|
| print() | 변수나 데이터를 화면에 출력 print("출력할 문장") print(출력할 변수) |

| 형식 | 의미 |
|----|-----|
| %d | 정수 |
| %f | 실수 |
| %s | 문자열 |

서식 지정자 : %d (정수) %f(실수) %s(문자열)

```
print('%d' % 1)
```

```
print('%d %d' % (1,2))
```

```
print('%f' % (1.124))
```

```
print('%f %f' % (1.5,3.14))
```

```
print('%s' % ('one'))
```

```
print('%s, %s' % ('one','two'))
```

```
print("hello")
print("오늘도 굿모닝 입니다")

a = "애플파이"
b = 10
print(' 애플파이가 %d개 있네요 ' %b)
print(' %s가 %d개 있네요 ' %(a,b))

print(' 물가가 %d%%오른다고 하네요 ' %b)
```

```
오늘도 굿모닝 입니다
애플파이가 10개 있네요
애플파이가 10개 있네요
물가가 10%오른다고 하네요
```


문자열 서식화

```
name = "korea"  
age = 20
```

```
# '%' 연산자
```

```
string = "이름: %s, 나이: %d세" % (name, age)  
print(string)
```

```
# 문자열.format()
```

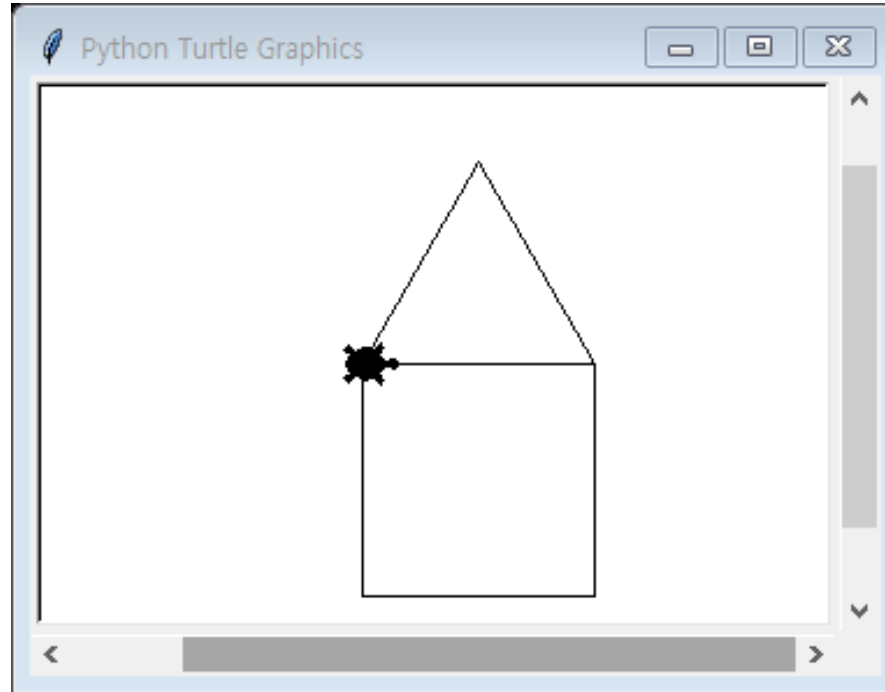
```
string1 = "이름: {}, 나이: {}세".format(name, age)  
print(string1)
```

```
# f- 문자열 (f-string)
```

```
string2 = f"이름: {name}, 나이: {age}세"  
print(string2)
```

도전

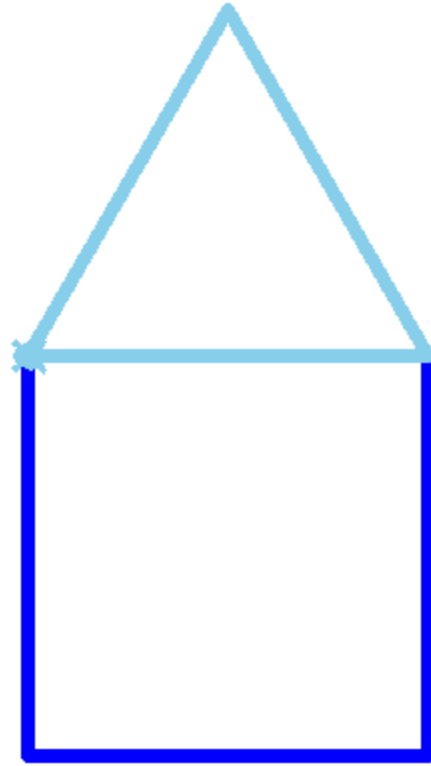
도형의 선분 길이를 입력받아 그려 보자.



```
size = int(input("집의 크기는 얼마로 할까요?"))
```

```
t.shape('turtle')  #'arrow' 'circle' 'square' 'classic' 'triangle'
```

도전



도전

1

```
import turtle
t = turtle.Turtle()

t.shape("turtle")
size = 200
t.width(7)
t.color("blue")

t.forward(size); t.right(90)
t.fd(size); t.rt(90)
t.fd(size); t.rt(90)
t.fd(size); t.rt(90)
```

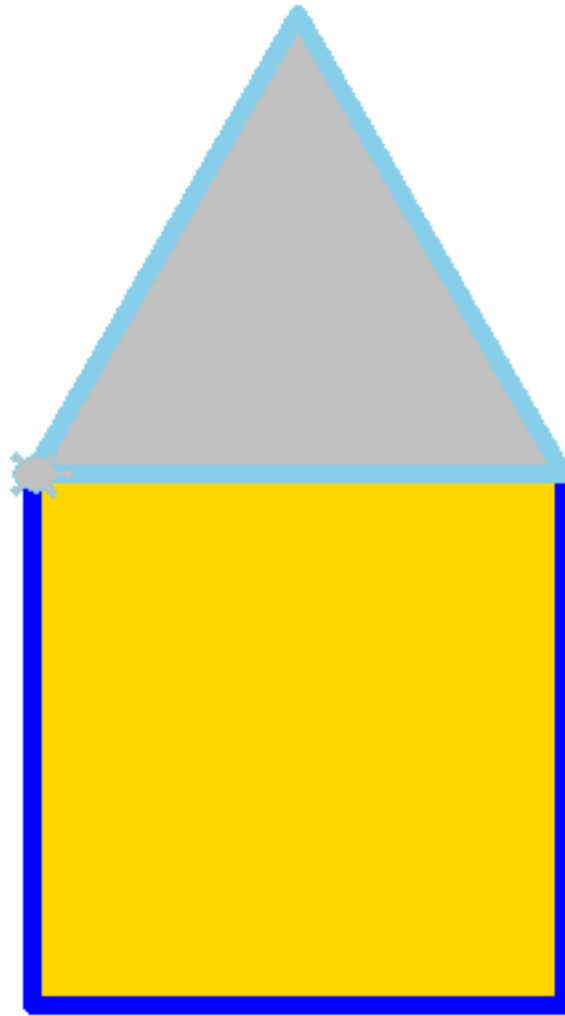
2

```
t.color("skyblue")

t.fd(size); t.lt(120)
t.fd(size); t.lt(120)
t.fd(size); t.lt(120)

turtle.done()
```

t.clear()



도전

```
t.fillcolor("gold")
```

```
t.begin_fill()
```

```
t.forward(size); t.right(90)
```

```
t.fd(size); t.rt(90)
```

```
t.fd(size); t.rt(90)
```

```
t.fd(size); t.rt(90)
```

```
t.end_fill()
```

```
t.fillcolor("silver")
```

```
t.begin_fill()
```

```
t.fd(size); t.lt(120)
```

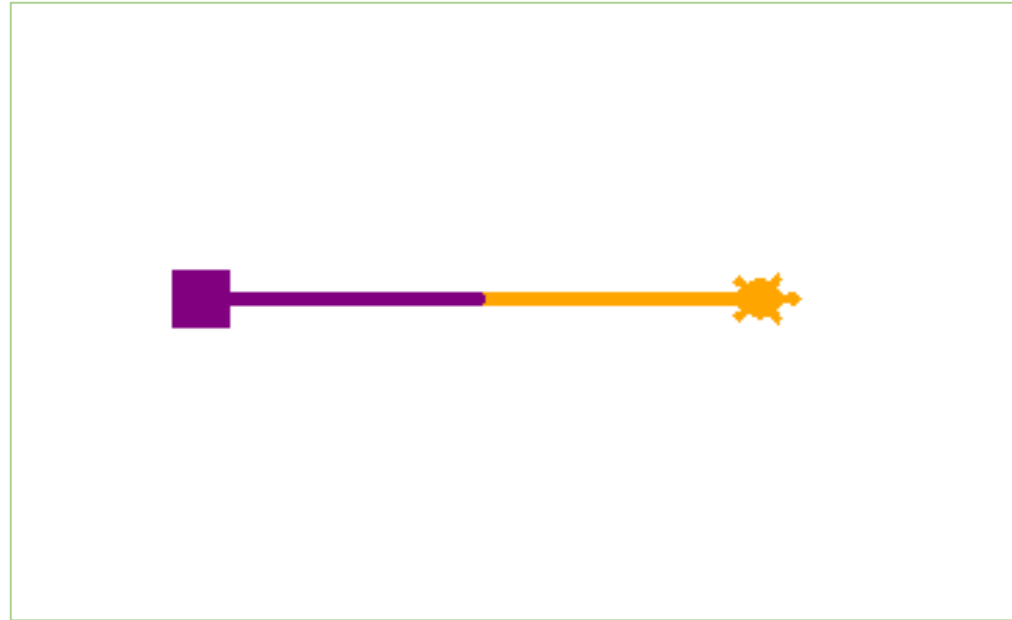
```
t.fd(size); t.lt(120)
```

```
t.fd(size); t.lt(120)
```

```
t.end_fill()
```

객체

```
import turtle          #turtle 모듈을 사용하기 위해 준비  
  
t = turtle.Turtle()    #turtle 모듈에 있는 Turtle 클래스 객체 t 생성  
s = turtle.Turtle()    # 객체 s 생성
```



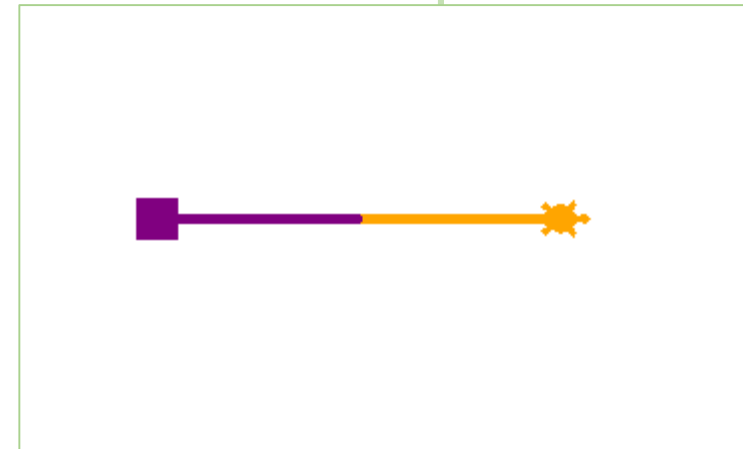
```
import turtle
t = turtle.Turtle()
s = turtle.Turtle()

t.shape("turtle")
s.shape("square")
t.color("orange")
s.color("purple")

w=5
t.width(w)
s.width(w)

t.forward(100)
s.backward(100)

turtle.done()
```



도전



```
size = 50
```

```
t.fillcolor("silver")  
t.begin_fill()
```

```
t.fd(size); t.lt(72)  
t.fd(size); t.lt(72)  
t.fd(size); t.lt(72)  
t.fd(size); t.lt(72)  
t.fd(size); t.lt(72)
```

```
t.end_fill()
```

```
s.fillcolor("gold")  
s.begin_fill()
```

```
s.fd(size); s.lt(60)  
s.fd(size); s.lt(60)  
s.fd(size); s.lt(60)  
s.fd(size); s.lt(60)  
s.fd(size); s.lt(60)
```

```
s.end_fill()
```