

이차함수
그래프

■ input() 함수

- 키보드를 이용하여 숫자나 문자열 등의 데이터를 입력 받을 때 사용
- 입력된 정보는 문자열

```
n = input('type a number : ')
type a number : 3
print(n)          #문자열로 입력된다
'3'

print(int(n))      #문자열을 정수로 변환
3

print(float(n))    #문자열을 실수형으로 변환
3.0

n = str(3)         #정수를 문자열로 변환
print(n)
'3'
```

기초 입출력 함수

- 입출력 함수의 사용 예

- 문자열을 입력 받아 다른 문자열과 결합하여 출력

```
name = input('Name: ') ◀ 문자열 입력  
print("Hello," + name)
```

- 두 정수를 입력 받아 곱셈 결과 출력

```
a = int(input('Number 1: ')) ◀ 문자열을 입력한 후 정수로 변환  
b = int(input('Number 2: '))  
print(a * b)
```

- 섭씨온도(C)를 입력 받아 화씨 온도(F)로 바꾸어 출력

```
C = float(input("Type 섭씨 온도 : ")) ◀ 입력 문자열을 실수로 변환  
F = C*9/5 + 32    #온도 변환  
print("화씨 온도는", F)
```

기본 자료형

자료형과 크기

- 부울형(Boolean)
 - 참(True) 또는 거짓(False) 중 하나의 값을 갖는 자료형(True, False)
 - 비교 결과를 저장할 때 사용

```
aVar = True
bVar = False
print(type(aVar))
<class 'bool'>
```

```
a = 1
b = 2
c = a > b
print(c)
False
```

```
d = a < b
print(d)
True
```

자료형과 크기

■ 정수(integer)형

- 소수점 없는 숫자 표현

```
intVar = 5  
print(type(intVar))  
<class 'int'>
```

- 여러 가지 정수 표현

```
1, 23, 3493      # 10진수  
0b010101, 0b110010  # 2진수  
0o1, 0o27, 0o6645   # 8진수  
0x1, 0x17, 0xDA5    # 16진수
```

자료형과 크기

- 실수(float)형

```
fVar = 1.0  
print(type(fVar))  
<class 'float'>
```

```
# 여러 가지 실수 표현  
0., 0.0, .0, 1., 1.0, 1e0, 1.e0, 1.0e2
```

- 쉽게 식별할 수 있도록 _를 사용한 숫자의 표현(파이썬 3.6 부터 적용)

```
>>> 100_000.000_0001, 0xFF_FF, 0o7_777, 0b_1010_1010  
(100000.0000001, 65535, 4095, 170)  
>>> 0.00_10_27  
0.001027
```

자료형과 크기

- 문자열(string)
 - 나열된 문자들
 - ' ' 또는 " "로 둘러 씌움

```
strVar = "test"  
print(type(strVar))  
<class 'str'>
```

```
'This is a literal string'  
"This is another string"
```

```
"I'm a Python fanatic"      #""" 속에 작은 따옴표를 사용할 경우  
'I'm a Python fanatic'     #' ' 속에 큰 따옴표를 사용할 경우
```

```
'A not very long string \   #문장을 두 줄 이상으로 표시할 경우  
that spans two lines'
```


역사
자

연산자의 종류

- 지정 연산자(assignment operator); 할당 연산자

```
i = 3  
i = i + 2
```

- 변수의 값을 지정하기 위해 사용(=)
- 산술 연산자(arithmetic operator)
 - 사칙 연산자: $+$, $-$, $*$, $/$
 - 정수 나눗셈: $//$
 - 나머지 연산: $%$

연산자	설명	사용 예
$x + y$	x와 y를 더함	$c = a + b$
$x - y$	x에서 y를 뺌	$s = a - b$
$x * y$	x와 y를 곱함	$m = a * b$
x / y	x를 y로 나눔	$d = a / b$
$x // y$	x를 y로 나눔 (정수 나눗셈)	$d = a // b$
$x \% y$	x를 y로 나눈 나머지	$r = a \% b$
$x ** y$	x의 y 제곱	$r = 2 ** 5$

```
a = 10  
b = 3  
c = a % b ⇨ 1  
d = a // b ⇨ 3  
e = a / b ⇨ 3.3333333333333335
```

연산자의 종류

- 관계 연산자(relation operator)
 - 항의 대소 관계 또는 동등 관계 판정
 - 연산 결과는 True(참) 또는 False(거짓)

```
score = 95
print(score >= 90)   ⇒ True
print(score < 80)    ⇒ False
print(score == 90)   ⇒ False
print(score != 80)   ⇒ True
```

연산자	설명	사용 예
<code>x < y</code>	x가 y보다 작으면 True	<code>x=1; y=2; print(x<y)</code>
<code>x <= y</code>	x가 y보다 작거나 같으면 True	<code>x=1; y=2; print(x<=y)</code>
<code>x > y</code>	x가 y보다 크면 True	<code>x=1; y=2; print(x>y)</code>
<code>x >= y</code>	x가 y보다 크거나 같으면 True	<code>x=1; y=2; print(x>=y)</code>
<code>x == y</code>	x와 y가 같으면 True	<code>x=1; y=2; print(x==y)</code>
<code>x != y</code>	x와 y가 다르면 True	<code>x=1; y=2; print(x!=y)</code>

연산자의 종류

- 논리 연산자(logical operator)
 - 논리곱(and), 논리합(or), 논리 부정(not)
 - 논리 연산의 대상은 True 또는 False. 결과도 True 또는 False

```
exam = 85
report = 85
print(exam >= 80 and report >= 80)
True

print(exam >= 90 or report >= 90)
False

print(not(exam < 90))
False
```

연산자	설명	사용 예
x and y	x와 y가 모두 참일때만 연산결과가 참	a=5; print(a > 0 and a < 10)
x or y	x와 y 둘 중 하나라도 참이면 연산 결과가 참	a=5; print(a < -3 or a > 3)
not x	x의 부정. x가 참(거짓)이면 결과는 거짓(참).	a=False; print(not a)

연산자의 종류

■ 멤버 연산자(membership operator)

- **in을 사용**

- 특정 문자열 또는 값이 문자열 또는 리스트에 속해 있는지 판별하는 연산자

```
'py' in 'python'  
True  
'ty' not in 'python'  
True
```

■ 식별 연산자(identity operator)

- is : 양쪽 변수의 주소가 같은지 판단. if id(a) == id(b)
- is not : 양쪽 변수의 주소가 다른지 판단. if id(a) != id(b)

```
a = 2  
b = 2  
print(id(a))  
1921220752  
  
print(id(b))  
1921220752
```

```
print(id(a) == id(b))  
True  
  
print(a is b)  
True  
  
print(a is not b)  
False
```

■ 타입 에러(type error)

- 연산할 수 없는 자료형을 연산하면 타입 에러 발생

```
age = 23
message = "Happy " + age + "rd Birthday!"
print(message)
```

Traceback (most recent call last):

File "birthday.py", line 2, in <module> ❶

message = "Happy " + age + "rd Birthday!"

TypeError: Can't convert 'int' object to str implicitly ❷

❶ 에러 발생 위치

❷ 에러 내용; int 형을 암시적으로 문자열로 변환할 수 없음