

Chapter 03. 변수

목차

1. 정보의 표현
2. 변수의 개념과 활용

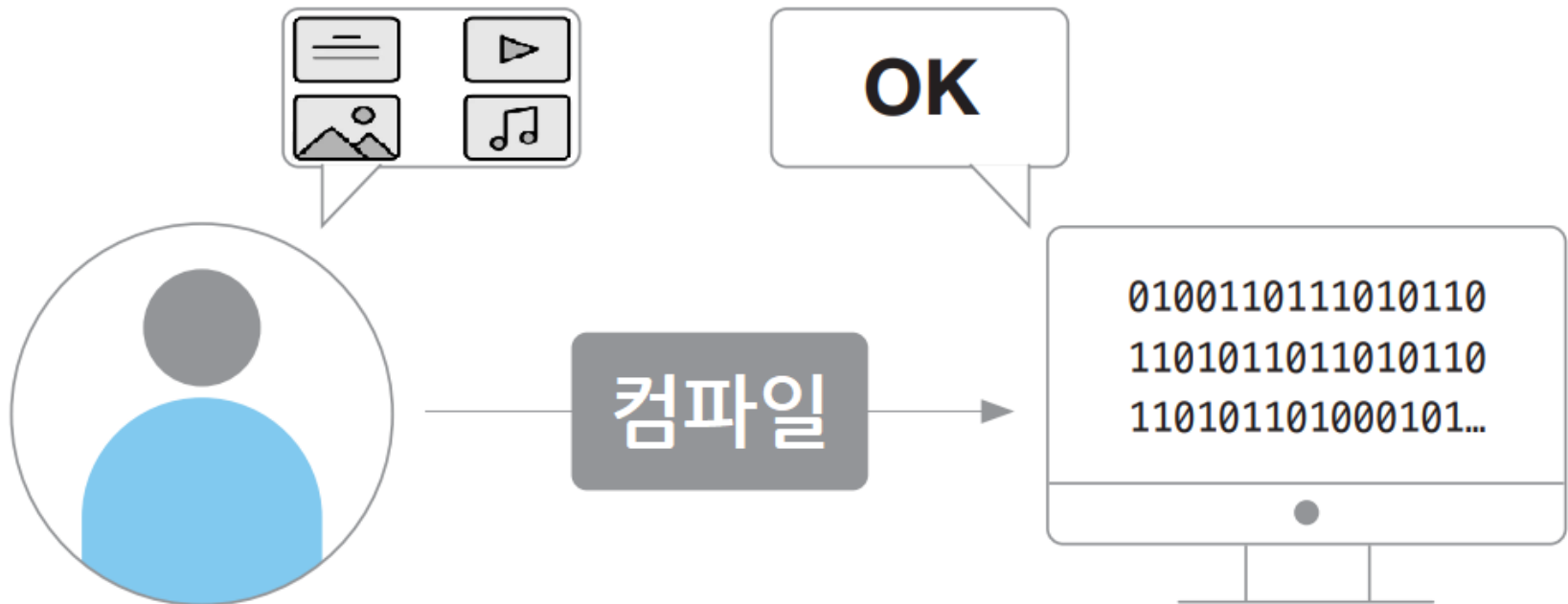
01

정보의 표현

01. 정보의 표현

[컴퓨터의 데이터 처리 과정]

- 컴퓨터는 다양한 종류의 데이터를 처리할 수 있다.
- 하지만 프로그래밍 언어와 컴파일 과정이 필요하다.



01. 정보의 표현

I. 컴퓨터의 데이터 처리

■ 데이터와 정보

- 데이터란 어떤 의미나 목적을 포함하지 않은 수집되거나 측정된 값 혹은 자료를 의미
- 정보는 이런 데이터를 의도나 목적에 맞게 분석 혹은 가공하여 그 의미를 표현
- 컴퓨터는 정보처리장치이며, 데이터를 사람이 이용할 수 있는 정보로 제공하는 장치

01. 정보의 표현

I. 컴퓨터의 데이터 처리

■ 이진 데이터의 표현

- 컴퓨터에서 데이터를 처리하는 기본 단위는 비트이며, 0과 1의 이진값을 표현
- 컴퓨터는 필요한 모든 데이터를 이진수로 처리하며, 비트는 스위치나 전구의 off 나 on처럼 0이나 1이라는 2개의 값만 저장
- 여러 개의 비트들을 비트열이라고 부르며 비트열이 모이면 아주 많은 값을 표현

할 수 있음 표 3-1 비트열의 개수에 따라 표현 가능한 값의 범주

비트 개수	표현 가능한 값의 개수	비트와 값의 패턴
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
8	256	2^8
16	65,536	2^{16}
n	2^n	2^n

01. 정보의 표현

I. 컴퓨터의 데이터 처리

■ 이진 데이터의 표현

여기서 잠깐 데이터 단위

표 3-2 데이터 단위

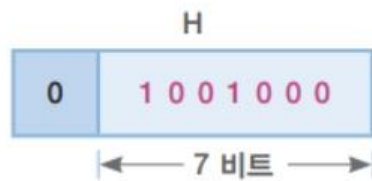
이름	비트 개수	값의 개수	저장 항목의 예
비트(bit)	1	$2^1 = 2$	예/아니오
바이트(byte)	8	$2^8 = 256$	알파벳 하나
킬로(kilo)	10	$2^{10} = \text{약 } 10^3$	문단 몇 개
메가(mega)	20	$2^{20} = \text{약 } 10^6$	1분 길이 노래
기가(giga)	30	$2^{30} = \text{약 } 10^9$	256개 노래
테라(tera)	40	$2^{40} = \text{약 } 10^{12}$	영화 1300편
페타(peta)	50	$2^{50} = \text{약 } 10^{15}$	국내 인구 1/2의 주민정보
엑사(exa)	60	$2^{60} = \text{약 } 10^{18}$	2018년 드롭박스 데이터 용량
제타(zetta)	70	$2^{70} = \text{약 } 10^{21}$	세계 인구 1인당 36t 분량의 책

01. 정보의 표현

II. 다양한 정보의 표현

■ 텍스트

- 문자 코드표는 모든 컴퓨터에서 동일하게 유지되도록 해야 통신이나 파일 공유 작업이 가능하므로 아스키라고 하는 코드 표준을 사용
- 아스키는 전체 128개 문자에 대해 각 문자당 8개 비트(첫 번째 비트 0은 다른 용도로 사용)를 사용해 다양한 값을 표현



아스키 코드표 일부 →

Binary	Char	Binary	Char
00100000	space	01000001	A
00100001	!	01000010	B
00100010	"	01000011	C
00100011	#	01000100	D
00100100	\$	01000101	E
00100101	%	01000110	F
00100110	&	01000111	G
00100111	'	01001000	H
00101000	(01001001	I

그림 3-4 아스키 코드표

01. 정보의 표현

II. 다양한 정보의 표현

■ 이미지

- 컴퓨터에서 이미지를 처리하는 방식은 연속적인 점을 찍어 그림을 그리듯 아주 작은 격자를 사용해 표현, 이것을 픽셀 또는 화소라고 함
- 해상도는 동일한 면적에 표시할 수 있는 픽셀 수를 의미
- 해상도가 높을수록 더 또렷하게 볼 수 있고 이러한 해상도를 측정하는 단위로 PPI를 사용

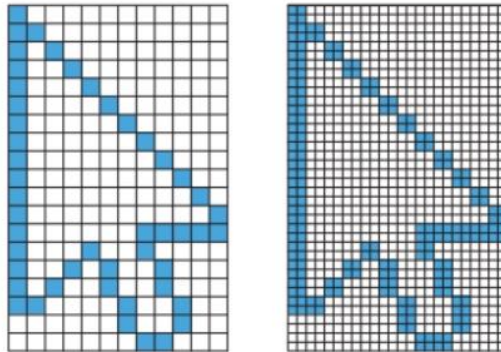


그림 3-6 픽셀 수가 다른 동일 이미지

01. 정보의 표현

II. 다양한 정보의 표현

■ 사운드

- 컴퓨터의 입력장치인 마이크로 수집되는 사운드는 디지털 신호로 변환하는 과정인 ADC를 진행
- 컴퓨터에 저장된 사운드를 재생하려면 역으로 변환하는 과정인 DAC와 스피커 등의 출력장치를 통해 원래의 아날로그 신호 형태로 전달

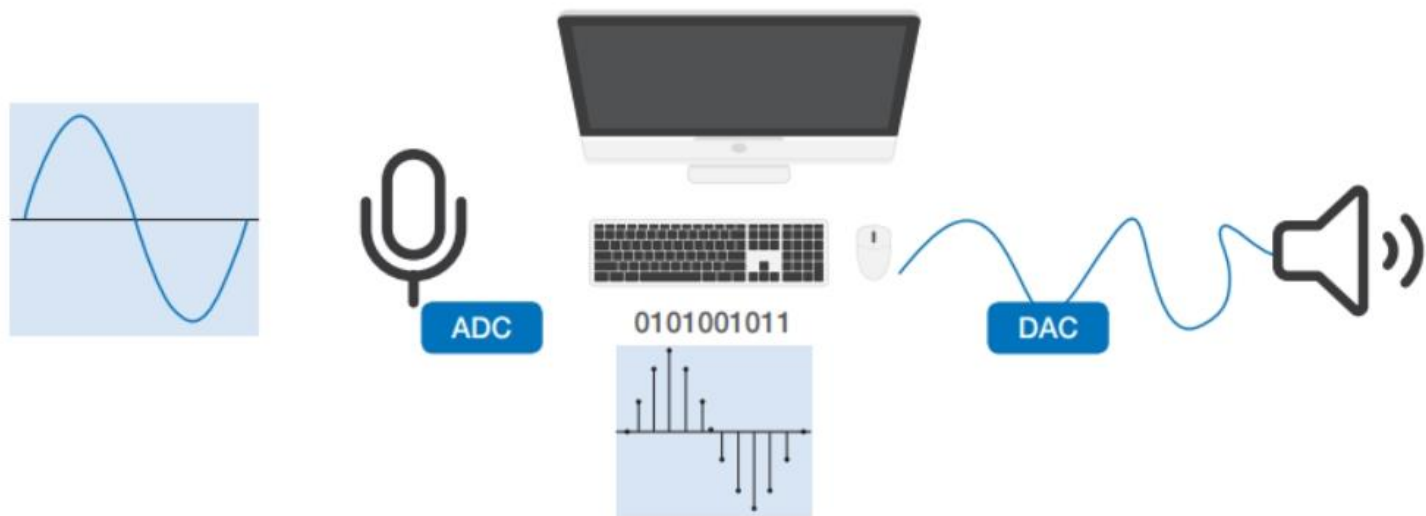


그림 3-9 사운드의 변환 과정

02

변수의 개념과 활용

02. 변수의 개념과 활용

I. 변수의 개념과 사용

- 메모리 주소와 변수명
 - 프로그램이 실행되는 동안 필요한 데이터를 저장하는 공간이 변수
 - 변수의 이름(변수명)은 결국 메모리의 특정 위치에 이름을 붙이는 것

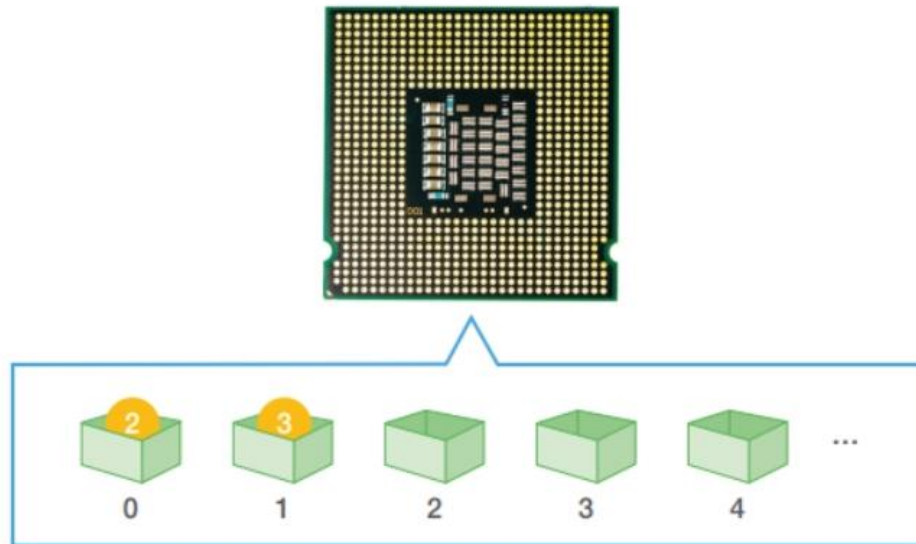


그림 3-11 물건을 담은 상자와 같이 값을 저장하는 변수

02. 변수의 개념과 활용

I. 변수의 개념과 사용

- 변수 생성과 값 저장
 - 파이썬에서는 변수명과 대입 연산자(=)를 사용하여 변수를 만들고 값을 저장
 - 대입 연산자 우측에는 숫자나 문자 등 메모리에 저장하려는 값을 직접 적거나 수식을 사용

```
>>> 변수명 = 저장할 값
```

```
>>> sum = 3
```

오른쪽 값을 왼쪽 변수에 저장(대입)

02. 변수의 개념과 활용

I. 변수의 개념과 사용

- 변수 생성과 값 저장

실습 3-1

변수를 만들고 값 저장하기

①

```
>>> x = 3
>>> y = 5
>>> z = "안녕"
```



그림 3-13 변수 생성과 값 저장

②

```
>>> x
3
>>> y
5
>>> z
'안녕'
```

```
>>> sum = x + y
>>> print(sum)
8
```

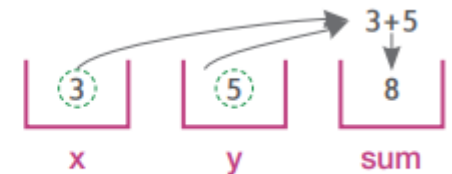


그림 3-14 변수 값 저장 시, 수식 사용

02. 변수의 개념과 활용

I. 변수의 개념과 사용

- 변수명 생성 시 주의 사항
 - 문자(A-z)와 숫자(0-9), _(underscore)를 사용할 수 있음
 - '2x'처럼 변수명이 숫자로 시작하면 안 됨
 - 대소문자를 구분하므로, 'data'와 'Data'는 다른 변수로 취급
 - '합계'와 같은 한글 변수명도 사용할 수 있음
 - 파이썬의 키워드(keyword)인 True, if, for 등은 사용할 수 없음

02. 변수의 개념과 활용

I. 변수의 개념과 사용

- 변수 생성과 값 저장

실습 3-2

파이썬의 키워드 확인하기

- ① 예약어를 가지고 있고 이를 변수 이름으로 사용할 수 없음

```
>>> import keyword
>>> print(keyword.kwlist)
['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async', 'await',
'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass',
'raise', 'return', 'try', 'while', 'with', 'yield']
```


02. 변수의 개념과 활용

II. 변수 값의 변경과 입력

실습 3-3

변수 값을 변경하고 입력받기

- ② 변수 number와 name에 각각 숫자와 문자열을 저장하고 출력

```
>>> number = 5
>>> name = "Kim"
>>> print(number, name)
5 Kim
```



그림 3-15 변수 생성과 값 저장

- ③ 대입 연산자 우측에 바꾸려는 값이나 수식을 입력하면 변수의 값이 변경

```
>>> number = 3
>>> name = "Kim" * number
>>> print(number, name)
3 KimKimKim
```



그림 3-16 변수 값 변경

02. 변수의 개념과 활용

II. 변수 값의 변경과 입력

실습 3-3

변수 값을 변경하고 입력받기

- ③ `input()` 함수를 이용해 `name` 변수 값을 키보드로 직접 입력받아 바꾸고 출력

```
>>> name = input("이름 입력 : ")
```

```
이름 입력 : 홍길동
```

```
>>> print(name)
```

```
홍길동
```

`input()` 함수의 괄호() 안에 적은 메시지가 입력할 때 표시됨

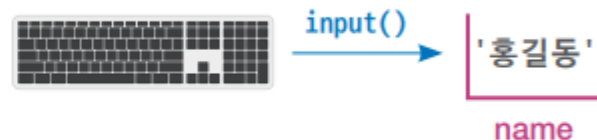


그림 3-17 변수 값 입력받기

- ④ `number` 변수에는 `input()` 함수로 입력한 정수를 저장하고, 수식을 만들어 출력

```
>>> number = int(input("번호 입력 : "))
```

```
번호 입력 : 5
```

```
>>> print(name * number)
```

```
홍길동홍길동홍길동홍길동홍길동
```

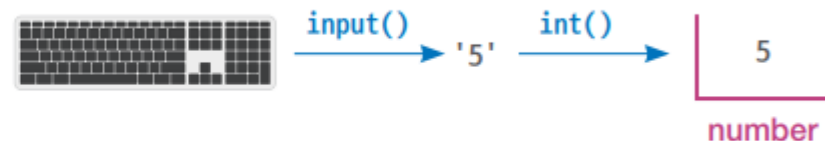


그림 3-18 정수 입력과 수식의 결과 출력

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 정수와 실수

표 3-3 데이터형의 종류

데이터형	설명	예시
int	정수(integer)	5 -1 100 -13123
float	실수(floating-point)	5.23 -0.024 2349012.412
str	문자열(string)	"python" '문자' '123'
bool	불린(boolean)	True False

실습 3-4

점수의 합계와 평균을 계산해서 출력하기

- ① 3개의 변수에 각각 점수를 저장하고 합계를 계산

```
>>> x = 85
>>> y = 92
>>> z = 76.5
>>> sum = x+y+z
```

02. 변수의 개념과 활용

III. 데이터형과 형 변환

- 정수와 실수

실습 3-4

점수의 합계와 평균을 계산해서 출력하기

- ② `type()` 함수로 각 변수의 데이터형을 확인

```
>>> type(x)
<class 'int'>
>>> type(sum)
<class 'float'>
```

- ③ 합계를 출력하고, 평균도 계산해서 출력

```
>>> print("합계 = ", sum)
합계 = 253.5
>>> print("평균 = ", sum/3)
평균 = 84.5
```

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 문자열

- 하나 이상의 문자를 저장하는 데이터형이 문자열(텍스트)
- 파이썬에서는 큰따옴표(" ") 나 작은따옴표(' ')를 문자열 기호로 사용

```
>>> name = "Kim"
```

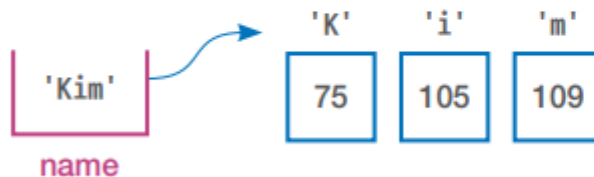


그림 3-19 문자열 처리

02. 변수의 개념과 활용

III. 데이터형과 형 변환

- 문자열

실습 3-5

입력한 문자의 실제 숫자 값 확인하기

- ① 키보드로 하나의 문자를 입력받아 변수 ch에 저장하고 값을 확인

```
>>> ch = input("문자 하나만 입력하세요 : ")
문자 하나만 입력하세요 : a
>>> ch
'a'
```

- ② ord() 함수를 이용해 입력한 문자 'a'가 실제로 처리되는 값이 97임을 확인

```
>>> ord(ch)
97
>>> bin(ord(ch))
'0b1100001'
```

02. 변수의 개념과 활용

III. 데이터형과 형 변환

▪ 정수와 실수

실습 3-6

문자열의 다양한 형태 알아보기

- ① 문자열 데이터형은 덧셈과 곱셈 연산이 가능

```
>>> a = "Hello," + "students~"
>>> type(a)
<class 'str'>
>>> b = 3
>>> print(a * b)
Hello,students~Hello,students~Hello,students~
```

- ② 긴 문자열의 경우 여러 줄로 나눈 상태(줄바꿈)로 변수에 저장

```
>>> a = """Hello,
students~"""
>>> print(a)
Hello,
students~
>>> a
'Hello,\nstudents~'
```

두 줄로 나뉜 문자열이 변수에 저장됨
3개의 따옴표를 사용하면 입력한 모양(Enter나 Tab 등이 적용) 그대로 저장

문자열 변수 a를 출력한 모양

문자열 변수 a에 실제 저장된 값

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 정수와 실수

실습 3-6

문자열의 다양한 형태 알아보기

- ③ 문자열 사이에 줄바꿈 문자나 탭 문자('wt')를 넣어 변수에 저장하고 출력

```
>>> b = "안녕하세요,\t학생 여러분~\n반갑습니다!"
>>> print(b)
안녕하세요,      학생 여러분~
반갑습니다!
```

실습 3-7

문자열 서식(formatting) 설정하기

- ① 좋아하는 과일을 변수에 저장해 두고, 하나의 문장으로 표현

```
>>> a = "사과"
>>> "I think " + a + " is the best fruit"
'I think 사과 is the best fruit'
>>> "I think %s is the best fruit" % a ..... %s 자리가 변수 a의 값으로 교체
'I think 사과 is the best fruit'
```


02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 문자열

실습 3-7

문자열 서식(formatting) 설정하기

- ② 여러 개의 값이 필요할 때는 % 기호 뒤에 괄호를 이용하고, 개수가 일치하도록 주의

```
>>> b = "배"
>>> "I think %s and %s are the best fruits" % (a, b)
'I think 사과 and 배 are the best fruits'
>>> c = 3
>>> "You ate %d %s today" % (c, a) ..... 변수 c는 정수형이라 %d를 사용
'You ate 3 사과 today'
```

- ③ 실수형에는 %f 포맷 코드를 사용하면 되는데, 마침표('.') 뒤에 숫자를 사용해 소수점 이하 자릿수를 정할 수 있음

```
>>> "반지름은 %d이고, 원주율은 %.2f입니다." % (10, 3.141592) ..... 소수점 이하 2자리까지만 표시하도록 설정
'반지름은 10이고, 원주율은 3.14입니다.'
```

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 문자열

여기서 잠깐 포맷 코드의 종류

표 3-4 포맷 코드의 종류와 사용 예시

포맷 코드	설명	예시	실행 결과
%s	문자열	"I like %s" %("meet")	'I like meet'
%d	정수	"키는 %d(cm)" %(180)	'키는 180(cm)'
%f	실수	"%f와 %5.1f" %(3.14, 3.14)	'3.140000와 3.1'
%c	문자	"90 이상은 %c 등급" %("A")	'90 이상은 A 등급'
%x	16진수	"100은 16진수로 %x" %(100)	'100은 16진수로 64'
%e	지수	"100은 %e" %(100)	'100은 1.000000e+02'

02. 변수의 개념과 활용

III. 데이터형과 형 변환

- 문자열

실습 3-8

숫자의 크기 비교하기

- 정수형 변수를 만들고 값을 저장한 다음, '>' 연산자로 식을 만들어 결과를 확인

```
>>> x = 3
>>> y = 5
>>> x > y
False
```

- 비교 연산자를 사용한 수식의 결과를 변수 answer에 저장하고 데이터형과 저장된 값을 출력

```
>>> answer = x > y
>>> type(answer)
<class 'bool'>
>>> print(answer)
False
```

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 형 변환

- 변수의 데이터형을 바꾸는 것이 형 변환
- `int()`, `float()`, `str()` 함수를 사용

표 3-5 데이터형에 따른 형 변환

함수	설명	예시	실행 결과
<code>int()</code>	다른 데이터형을 정수로 변환	<code>int("100")</code>	100
<code>float()</code>	다른 데이터형을 실수로 변환	<code>float(3)</code>	3.0
<code>str()</code>	다른 데이터형을 문자열로 변환	<code>str(12345)</code>	'12345'

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 형 변환

실습 3-9

값을 입력받아 실수형으로 합계 출력하기

- ① 두 변수 x와 y에 입력한 값을 형 변환하여 실수형으로 저장

```
>>> x = float(input("첫번째 수 : "))
```

```
첫 번째 수 : 123.456
```

```
>>> y = float(input("두번째 수 : "))
```

```
두 번째 수 : 1000
```

정수로 입력해도 실수형으로 변환되어 저장

- ② 포맷 기호를 사용하는 경우와 문자열을 연결하는 경우, 쉼표(,)를 이용해 순서대로 출력하는 경우의 3가지 결과를 비교

```
>>> print("%f와 %f의 합은 %f" % (x, y, x+y))
```

```
123.456000와 1000.000000의 합은 1123.456000
```

포맷 기호 사용

```
>>> print(str(x) + "와 " + str(y) + "의 합은 " + str(x+y))
```

```
123.456와 1000.0의 합은 1123.456
```

숫자는 문자열로 변환한 후 연결(+)

```
>>> print(x, "와 ", y, "의 합은 ", x+y)
```

```
123.456 와 1000.0 의 합은 1123.456
```

순서대로 출력(,)

02. 변수의 개념과 활용

IV. 변수의 값 복사와 교환

- 변수 x 에 10이 저장되어 있고, 다른 변수 y 에 x 의 값을 복사하려면 $y = x$ 문장을 사용
- 대입 연산자(=)의 우 측에 특정 값이나 수식이 아닌 다른 변수명이 오면 그 변수에 저장되어 있던 값이 그대로 복사

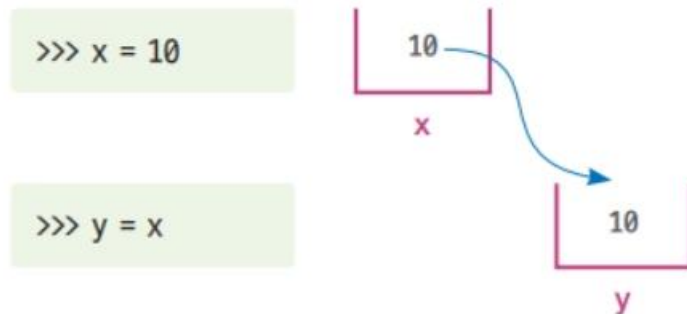


그림 3-20 변수의 값 복사

02. 변수의 개념과 활용

III. 변수의 값 복사와 교환

- x에 10, y에 20이 저장되어 있을 때 두 값을 교환하여 x는 20, y는 10을 갖도록 하는 경우라면?
- 파이썬에서는 `x, y = y, x` 라는 문장으로 간단하게 값을 교환할 수 있음

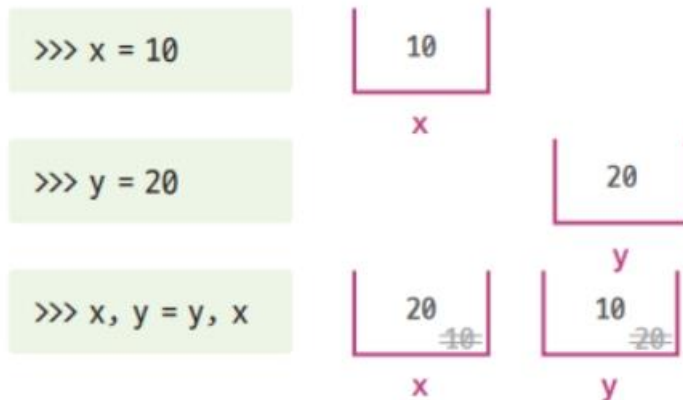


그림 3-21 두 변수의 값 교환

02. 변수의 개념과 활용

III. 데이터형과 형 변환

■ 형 변환

실습 3-10

변수의 값을 복사하고 교환하기

- ① 변수 x에 저장된 값을 y에 복사해서 출력

```
>>> x = 10
>>> y = x
>>> print("x = %d, y = %d" %(x, y))
x = 10, y = 10
```

- ② 변수의 값을 서로 교환한 후 출력

```
>>> x = "콜라"
>>> y = "주스"
>>> x, y = y, x
>>> print("x = %s, y = %s" %(x, y))
x = 주스, y = 콜라
```