

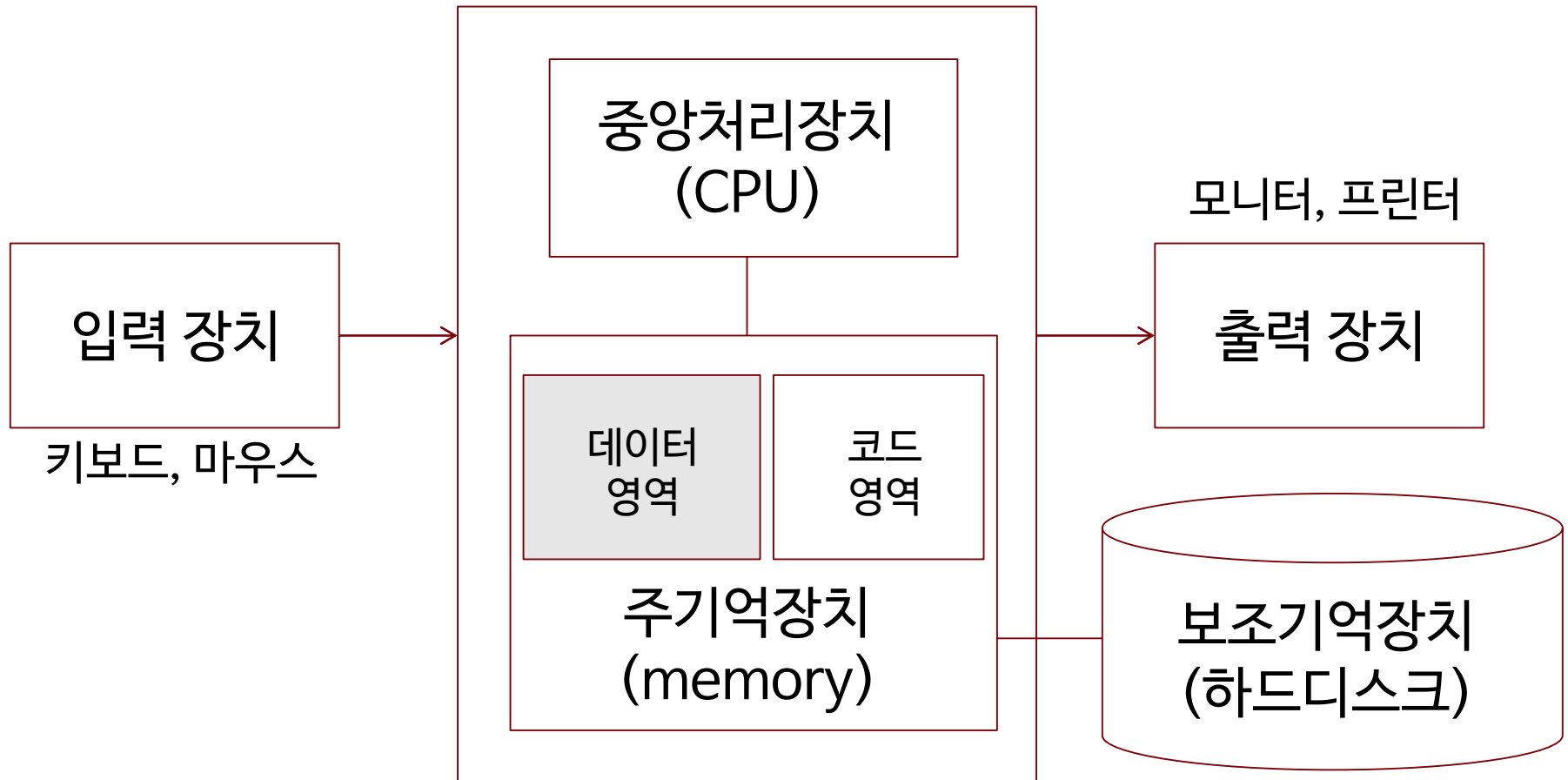
기초 PYTHON 프로그래밍

2. 객체, 변수, 자료형

1. 객체 (object)와 변수 (variable)
2. 객체 삭제하기
3. 자료형 (9가지)
4. 주석 (comment)

1. 객체 (object)와 변수 (variable)

◆ 데이터 영역과 객체 (object)



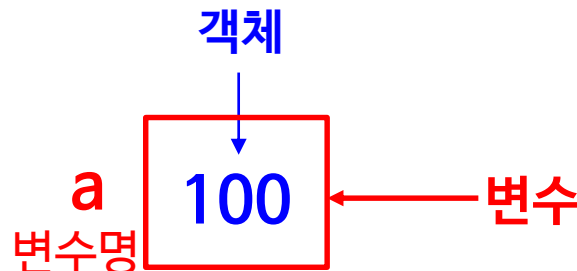
1. 객체 (object)와 변수 (variable)

◆ 객체 (object)

- 메모리에 저장된 자료를 ‘객체’라고 부른다.
- 모든 객체는 자료형 (data types)을 갖는다.
- 모든 객체는 고유 id를 갖는다.
- 객체를 저장한 공간을 변수 (variable)라고 하고 변수명 (variable name)으로 객체에 접근할 수 있다.

◆ 객체 생성하기


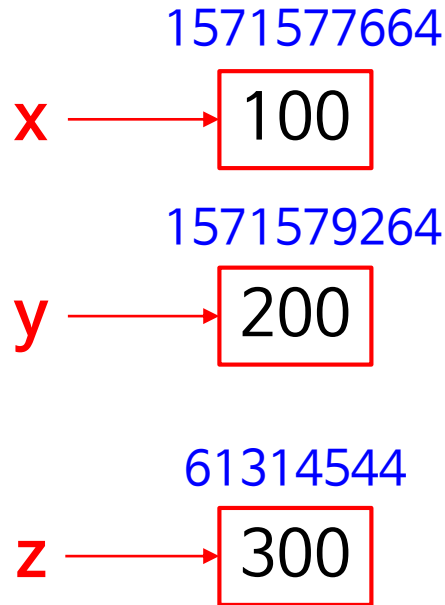
a = 100



1. 객체(object)와 변수(variable)

◆ 객체와 변수 예

```
>>> x = 100
>>> y = 200
>>> z = x + y
>>> print(x,y,z)
100 200 300
>>> id(x)
1571577664
>>> id(y)
1571579264
>>> id(z)
61314544
```



```
>>> type(x)
<class 'int'>
>>> type(y)
<class 'int'>
>>> type(z)
<class 'int'>
```

= assignment (할당연산자)
오른쪽의 값을 왼쪽 변수에 넣어라.
= 의 왼쪽에는 변수가 와야 한다.



1. 객체(object)와 변수(variable)

◆ 변수명 만들기

- 변수명은 영어 소문자, 대문자, 숫자, _ (underscore)로만 구성한다.
- 변수명은 숫자로 시작할 수 없다.
- 대소문자를 구분한다. 즉, data와 Data는 다른 변수이다.
- 변수명은 한글도 가능하다.
- 키워드(keyword)를 변수명으로 사용하면 안 된다.

〈 파이썬 키워드 (33개) 〉

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

1. 객체(object)와 변수(variable)

◆ 변수명 예



```
>>> number = 5
```

```
>>> score = 90
```

```
>>> python_score = 95
```

```
>>> _score = 100
```

```
>>> math-score = 90 # syntax 에러. 특수 기호는 _ 만 가능하다
```

```
>>> math1 = 80
```

```
>>> 1math = 80 # syntax 에러. 숫자로 시작할 수 없다
```

```
>>> 학생수 = 50 # 한글 변수명도 가능함
```

2. 객체 삭제하기

- ◆ del 을 이용하면 객체를 삭제할 수 있다.

```
>>> data = 100
```

```
>>> print(data)
```

```
100
```

```
>>> del data
```

```
>>> print(data)
```

```
Traceback (most recent call last):
```

```
File "<pyshell#13>", line 1, in <module>
```

```
    print(data)
```

```
NameError: name 'data' is not defined
```

◆ 파이썬 자료형 (data types)

- 수치 자료형 - int (정수), float (실수), complex (복소수)
- 부울 자료형 - bool (True / False)
- 군집으로 다루는 자료형
 - str (문자열)
 - list (리스트)
 - tuple (튜플)
 - set (집합)
 - dict (사전)

3.1. 수치 자료형 - 정수형

◆ 정수형 (int)

- 정수형 자료는 10진수, 2진수, 8진수, 16진수가 있다.

```
>>> score = 23
```

```
>>> print(score)
```

```
23
```

1525430384
score → 23

```
>>> type(score)
```

```
<class 'int'>
```

```
>>> id(score)
```

```
1525430384
```

type(변수명) : 변수의 자료형을 알려준다.

id(변수명) : 변수의 식별자를 알려준다.

3.1 수치 자료형 - 실수형

◆ 실수형 (float)

- 소수점을 포함하는 수이다.

```
>>> data = 1.5
```

```
>>> print(data)
```

```
1.5
```

```
>>> type(data)
```

```
<class 'float'>
```

```
>>> id(data)
```

```
51659472
```

51659472
data → 1.5

3.1 수치 자료형 - 복소수

◆ 복소수형 (complex)

- 복소수형은 실수부와 허수부로 표현한다. 허수부 뒤에는 j 또는 J를 붙인다.

```
>>> x = 3 + 5j
```

```
>>> print(x)
```

```
(3+5j)
```

```
>>> type(x)
```

```
<class 'complex'>
```

```
>>> id(x)
```

```
54137944
```

54137944

x → 3+5j

◆ 부울형 (bool)

- 파이썬의 부울형은 참 / 거짓을 나타내는 **True / False** 의 두 값만을 갖는다.

```
>>> a = True
```

```
>>> b = False
```

```
>>> type(a)
```

```
<class 'bool'>
```

```
>>> type(b)
```

```
<class 'bool'>
```

```
>>> x = 1
```

```
>>> y = x > 0
```

```
>>> print(y)
```

```
True
```

```
>>> type(y)
```

```
<class 'bool'>
```

◆ 문자열 (str)

- 따옴표를 이용하여 표현한다.

```
>>> name = 'Alice' # 홑따옴표
```

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

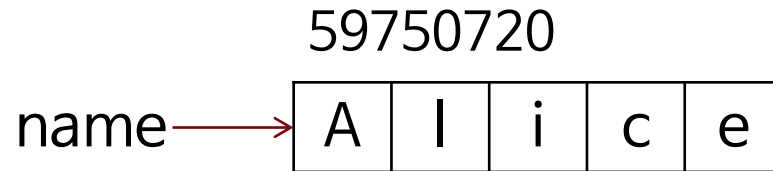
Alice

```
>>> type(name)
```

<class 'str'>

```
>>> id(name)
```

59750720



◆ 문자열 (str)

```
>>> city = "Seoul, Korea" # 쌍따옴표
```

```
>>> print(city)
```

```
Seoul, Korea
```

```
>>> language = '''python''' # 홀따옴표 세 개
```

```
>>> print(language)
```

```
Python
```

```
>>> w = """Python is widely used""" # 쌍따옴표 세 개
```

```
>>> print(w)
```

```
Python is widely used
```

3.4 군집 자료형 - 리스트

◆ 리스트 (list) - 대괄호 [] 로 표현한다.

- 여러 개의 자료들을 모아서 저장해야 하는 경우에 사용한다.

```
>>> scores = [80, 90, 85, 93, 78]
```

```
>>> print(scores)
```

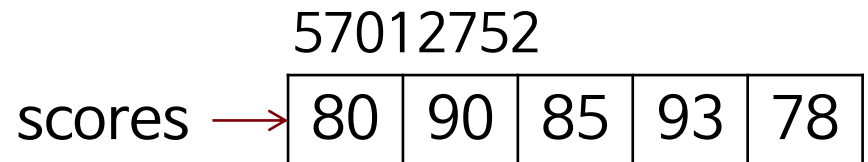
```
[80, 90, 85, 93, 78]
```

```
>>> type(scores)
```

```
<class 'list'>
```

```
>>> id(scores)
```

```
57012752
```



3.4 군집 자료형 - 튜플

◆ 튜플 (tuple) - 괄호 () 로 표현한다.

- 리스트처럼 여러 개의 자료를 저장할 수 있다.

```
>>> data = (3,6,5,9)
```

```
>>> print(data)
```

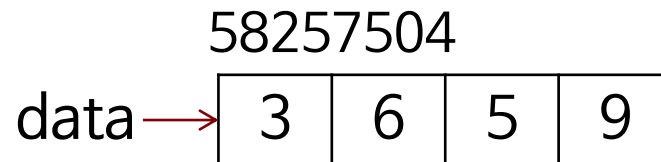
```
(3, 6, 5, 9)
```

```
>>> type(data)
```

```
<class 'tuple'>
```

```
>>> id(data)
```

```
58257504
```



3.4 군집 자료형 - 집합

◆ 집합 (set) - {} 로 표현한다.

- 중복되지 않는 여러 개의 자료들을 모아서 저장해야 하는 경우에 집합을 사용한다.

```
>>> primes = {7,17,3,5,7,19}
```

```
>>> print(primes)
```

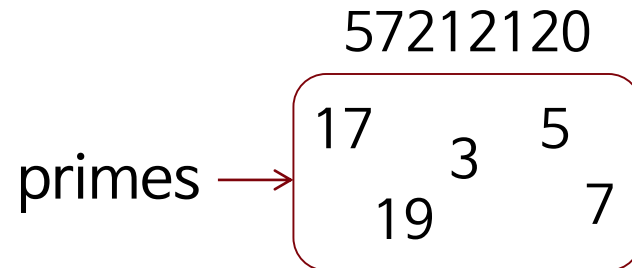
```
{17, 19, 3, 5, 7}
```

```
>>> type(primes)
```

```
<class 'set'>
```

```
>>> id(primes)
```

```
57212120
```



3.4 군집 자료형 - 사전

◆ 사전 (dict)

- **키(key)**와 **값(value)**의 쌍으로 구성되는 집합의 일종이다.
- 예) 1반 35명, 2반 32명, 3반 30명, 4반 33명의 학생들이 있다면 사전을 이용하여 다음과 같이 표현할 수 있다.

```
>>> count = {1:35, 2:32, 3:30, 4:33}
```

```
>>> print(count)
```

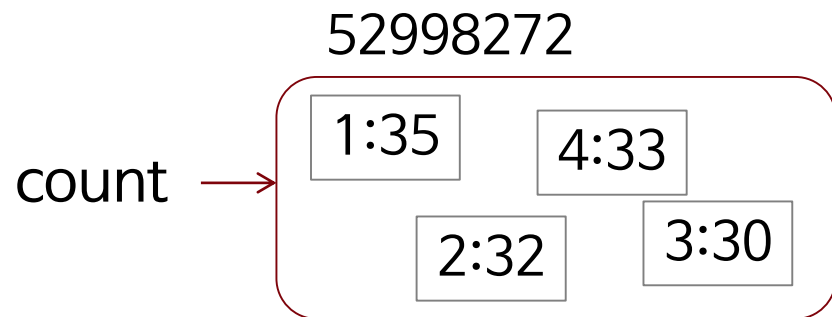
```
{1: 35, 2: 32, 3: 30, 4: 33}
```

```
>>> type(count)
```

```
<class 'dict'>
```

```
>>> id(count)
```

```
52998272
```



4. 주석 (comment)

- ◆ 프로그램에 설명을 넣을 때 주석을 사용함
- ◆ 한 줄 주석 - # 기호를 사용함

숙제 1번

```
print('hello world')  
print('python programming')
```

```
print('what is your name? ') # 이름 물어보기  
print('how old are you? ')  # 나이 물어보기
```

4. 주석 (comment)

- ◆ 여러 줄 주석 - 홀따옴표 세 개 또는 쌍따옴표 세 개를 사용함

```
""" 프로젝트 1.  
작성자 : 홍길동  
완성일 : 2016. 07. 31  
프로젝트 버전 : 0.01  
  
이 프로젝트는 파이썬 버전 3을 이용하였음 """  
  
print('start of the program')  
  
# 주요 코드 시작 부분  
  
print('.....')
```

""" ... """ - 쌍따옴표 3개로 작성한 부분을 docstring 이라고 한다.
docstring은 함수, 클래스, 모듈 등을 작성할 때 유용하다.