

컴퓨터정보과 파이선 프로그래밍

02주차 – 자료형

Interpreter

- 인터프리터 : 코드를 한 줄씩 읽어 내려가면서 해석하고 실행하는 방식
 - 비교: 컴파일러 - 모든 코드를 읽어서 문제가 없으면, 실행할 수 있도록 하는 방식
- Shell 실행방식
 - 프롬프트(>>>))에서 한 줄씩 입력하여 실행하고 결과 보여주는 방식
 - 대화형 인터프리터라고 함
- Editor를 이용한 실행방식
 - 여러 번 사용하거나 다량의 코드를 한꺼번에 작성하고 실행하기 위할 때 필요
 - 코드가 여러 줄이어도, 한 줄씩 읽으면서 해석하고 실행하는 방식임
 - 파일(.py)로 저장하여 처리

python 설치

- Python Interpreter
- Python Shell
- Python IDE
 - IDLE
 - Shell
 - Edit

개발환경

- IDE나 Web 환경 모두 가능

파이썬 개발 전 사항 (1장)

- 목차 보기
- 파이썬이란 p.19
- 파이썬의 특징 p.20
- 할 수 있는 것 p.23
- 파이썬 인터프리터 설치 p.27
- 파이썬과 에디터 p.36

사용자 입출력 (4장)

- 키보드 입력 (p.171)
 - 함수 : `input()`
 - 입력 값은 모두 문자열(`string`)
 - 예제
 - `input()`
 - `input("점수:")`
 - `score = input("점수:")`
 - `score = int(input("점수:"))`
 - `score = float(input("점수:"))`

사용자 입출력 (4장)

- 모니터 출력 (p.172)

- 함수 : `print()`

- 예제

- `print()`

- `print(1)`

- `print(1.1)`

- `print([1,2,3])`

- `print("123")`

- `print("1" "2" "3")`

- `print("1", "2", "3")`

- `print("1" + "2" + "3")`

- 특징 : 자동줄바꿈

- `print(1, end="")`

- `print(2, end="")`

- `print(3, end="")`

- `print()` 함수는 어떤 타입이든 출력 가능 → 출력 타입은 ' 문자열 '

자료형

- 파이선에서 다루는 기본 객체(데이터, 자료)의 종류
 - 단일형태
 - 숫자
 - 불
 - 문자열(실제 내용은 복합)
 - 복합형태
 - 리스트
 - 튜플
 - 딕셔너리
 - 집합
 - 사용자 정의 자료형
 - class

자료형과 기본 연산 (2장)

그룹	자료형	생성자	리터럴 표현	내용
수치형 (Number)	Integer	int()	1	정수
	Floating point number	float()	1.1	실수
	Complex Number	complex()	2 + 3j	복소수
논리형 (Boolean)	Boolean	bool()	True	참(True)과 거짓(False)만 값을 가짐
군집형 (Collection)	String	str()	"1" 혹은 '1'	문자열 (순서 0, 수정 X, 중복 0, 구성요소한정 0)
	List	list()	[1,2]	리스트 (순서 0, 수정 0, 중복 0, 구성요소한정 X)
	Tuple	tuple()	(1,2)	튜플 (순서 0, 수정 X, 중복 0, 구성요소한정 X)
	Set	set()	{1,2}	집합, 세트 (순서 X, 수정 0, 중복 X, 구성요소한정 X)
	Dictionary	dict()	{1: "1", 2: "2"}	사전, 딕셔너리 (순서 X, 수정 0, 중복 X, 구성요소한정 X) {key:value, ..., key _n :value _n }

숫자형

- 숫자(수치)형 Number p.46
 - 정수(integer)
 - 생성자 : `int()`
 - 상수 표기법 :
 - 2진수 : `0b10`
 - 8진수 : `0o10`
 - 10진수: `10`
 - 16진수 : `0x10`
 - 정해진 크기 제한 없음
 - 내부적으로는 2진수로 표기
 - 실수(floating-point)
 - 생성자 : `float()`
 - IEEE754(를 따르는 8바이트 크기의 실수형태)
 - 표기법
 - 소수점 표현방식 : `420.0`
 - 컴퓨터식 지수 표현방식 : `4.2e2`

산술 연산자

- `+`, `-`, `*`, `/`, `//`, `**`

- 예제

```
a = int(3)
```

```
b = int(2)
```

```
print( a + b )
```

```
print( a - b )
```

```
print( a * b )
```

```
print( a / b )
```

```
print( a // b )
```

```
print( a % b )
```

```
print( a ** b )
```

논리형

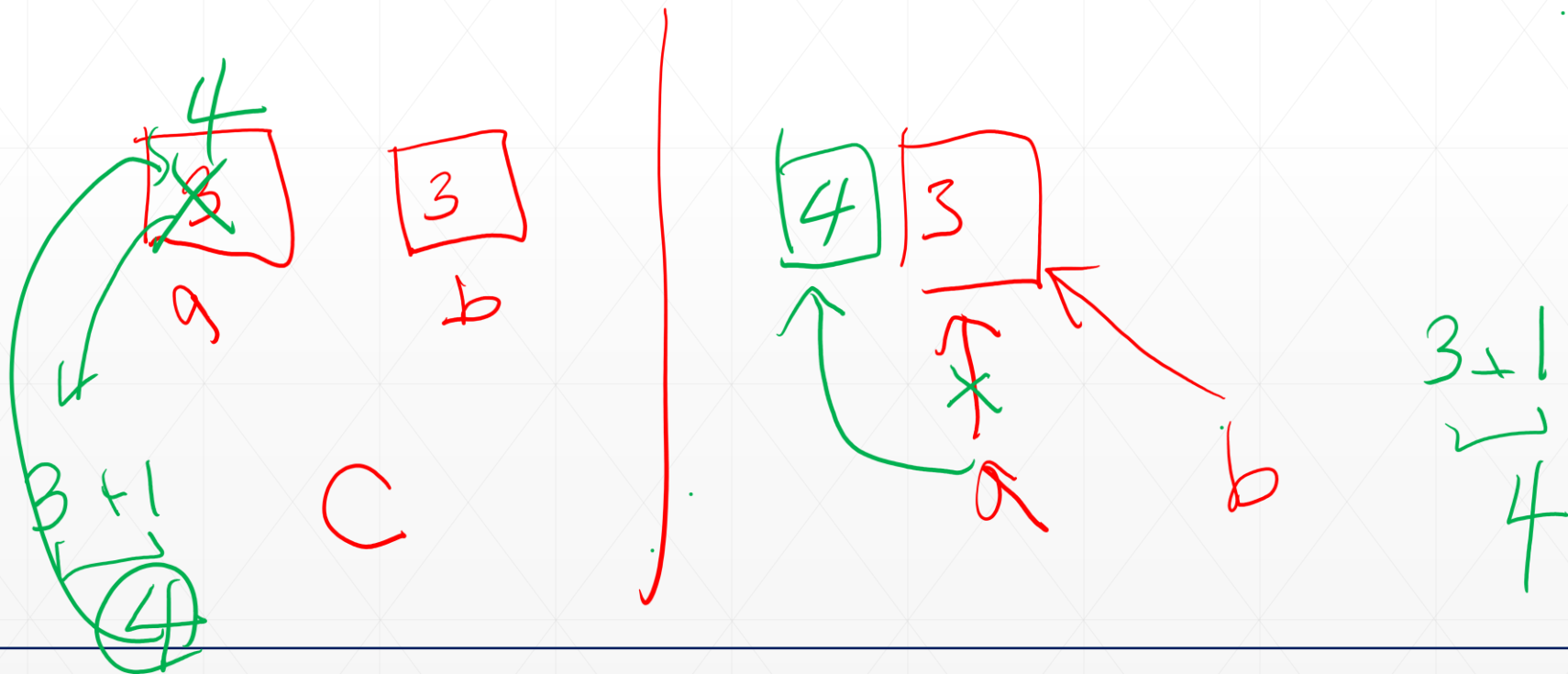
- 논리형 (boolean) p.106
 - 생성자 : `bool()`
 - 단 2개의 값만 갖는 자료형 : `True`(참), `False`(거짓)
 - 예제
 - `a = True`
 - `b = False`
 - `c = (3 <= 7)`
 - 자료형의 참과 거짓
 - `a = bool("")`
 - `b = bool("abc")`
 - `c = bool(0)`
 - `d = bool(1)`
 - `e = bool([])`
 - `f = bool([1,2,3])`
 - 비교(관계)연산자 : `==` , `!=` , `>` , `>=` , `<` , `<=`
 - 추가 함수
 - `type()` : 자료형 종류 확인

변수

malloc → free

■ 변수 p.111

- 변수 : (파이썬에서는) 객체(자료, 데이터, 값)를 가리키는 것
- 연산자 = assignment 기호 이용해 생성
 - 예제) $a = 1$
메모리에 1을 저장하고, 저장한 메모리의 위치를 가리킨다.
- 내장함수 `id()` : 객체의 주소를 반환



변수

- 임의정밀도 : Arbitrary Precision Integers
 - 파이썬3의 모든 정수는 임의 정밀도 (메모리 크기 허용에 따라 크기 제한 없음)
- Interning
 - 동일한 값을 가진 불변 객체들이 메모리에서 같은 위치를 참조하도록 하는 것
 - Integer Interning : -5~256
 - 미리 캐시하여 재사용
 - 나머지는 interning 사용하지 않음
 - String Interining
 - 특정 문자열들을 자동으로 인터닝하여 메모리 절약
 - 특수 문자가 포함된 경우는 인터닝이 적용되지 않을 수 있음.
 - 수동 인터닝 `sys.intern()`
 - 연산자 `is` , `==` 주의 할 것
 - `is` : 변수가 가리키고 있는 주소 비교
 - `==` : 변수가 가리키고 있는 실제 값 비교