절차지향(procedural) 프로그래밍

절차지향 프로그래밍

- C언어

물이 위에서 아래로 흐르는 것처럼 순차적인 처리가 중요시 되며
프로그램 전체가 유기적으로 연결되도록 만드는 프로그래밍 기법

절차지향의 단점

- 재사용할 수 없다.
- 확장성이 떨어진다.
- 유지보수가 어렵다.

객체지향(Object Oriented) 프로그래밍

프로그래밍 방식의 변화
초기 : 절차지향 방식
단점 : 조금만 복잡해져도 순서도로 나타내는 것이 불가능한 "스파게티 코드"

1968년 GOTO의 해로움이라는 논문에서 함수(프로시져)단위 호출방식의
구조적 프로그래밍 방식의 등장.
단점 : - 재사용할 수 없다.

이를 극복하기 위한 대안 : **객체 지향 프로그래밍**

객체지향(Object Oriented) 프로그래밍

객체지향 프로그래밍

- C++, JAVA, Python

구조적 프로그래밍과 다르게 큰 문제를 작게 쪼개는 것이 아니라,
먼저 작은 문제들을 해결할 수 있는 객체들을 만든 뒤,
이 객체들을 조합해서 큰 문제를 해결하는 상향식(Bottom-up) 해결을 도입.

이 객체란 것을 일단 한번 독립성/신뢰성이 높게 만들어 놓기만 하면
그 이후엔 그 객체를 수정 없이 재사용할 수 있으므로 개발 기간과 비용이 대폭 줄어들게 된다.

클래스와 객체

클래스는 객체의 틀이 되는 추상적인 개념이고, 객체는 클래스에 정의된 요소들의 실체입니다.

붕어빵틀 = 클래스
붕어빵 = 객체

클래스(class) 개념

```
class Sample:
        pass
```

위의 클래스는 아무런 기능도 없는 클래스이다.
껍질 뿐인 클래스도 인스턴스(instance)라는 것을 생성하는 기능을 갖는다.
(클래스에 의해서 생성된 객체를 인스턴스라고 부른다)
인스턴스는 클래스에 의해서 만들어진 객체로 한개의 클래스는 무수히 많은 인스턴스를 만들어 낼 수가 있다.
Sample 클래스의 인스턴스를 만드는 방법

```
a = Sample()
```

Sample()의 결과값을 돌려 받은 a가 인스턴스 마치 함수를 사용해서 그 결과 값을 돌려 받는 모습과 비슷하다.

객체와 인스턴스

객체와 인스턴스의 차이

클래스에 의해서 만들어진 객체를 인스턴스라고도 한다.

그렇다면 객체와 인스턴스의 차이는 무엇일까? 이렇게 생각 해 보자. cat = Animal() 이렇게 만들어진 cat은 객체이다. 그리고 cat이라는 객체는 Animal의 **인스턴스(instance)**이다. 즉 인스턴스라는 말은 특정 객체(cat)가 어떤 클래스(Animal)의 객체인지를 **관계**위주로 설명할 때 사용된다. 즉, "cat은 인스턴스" 보다는 "cat은 객체"라는 표현이 "cat은 Animal의 객체" 보다는 "cat은 Animal의 인스턴스" 라는 표현이 훨씬 잘 어울린다.

내장 함수

파이썬의 내장 함수는 import 하지 않고 즉시 사용 가능한 함수들이다. 내장 함수명은 일종의 키워드로 간주하여야 하며 사용자의 식별자로 사용하는 것은 피하여야 한다

내장 함수

입출력 관련 함수

함수명	기능
print(x)	객체를 문자열로 표시한다.
input([prompt])	사용자 입력을 문자열로 반환한다.
help([x])	x에 대한 도움말을 출력한다.
globals()	전역 변수의 리스트를 반환한다.
locals() 혹은 vars() vars(obj)	지역 변수의 리스트를 반환한다. __dict__ 어트리뷰트를 반환한다. (객체의 내부 변수가 저장된 딕셔너리)
del(x) 혹은 del x	객체를 변수 공간에서 삭제한다.
eval(expr)	값을 구한다.
exec(obj)	파이썬 명령을 실행시킨다.
open(filename[,mode]))	파일을 연다

내장 함수

기본 자료형의 생성과 변환 함수

함수명	기능
object()	새로운 object (모든 객체의 base)를 생성한다.
bool(obj)	객체의 진리값을 반환한다.
int(obj)	문자열 형태의 숫자나 실수를 정수로 변환한다.
float(obj)	문자열 형태의 숫자나 정수를 실수로 변환한다.
complex(re [, img])	문자열이나 주어진 숫자로 복소수를 생성한다.

내장 함수

기본 자료형의 정보를 얻는 함수

함수명	기능
type(obj)	객체의 형을 반환한다.
dir(obj)	객체가 가진 함수와 변수들을 리스트 형태로 반환한다.
repr(obj) ascii(obj)	evla()함수로 다시 객체를 복원할 수 있는 문자열 생성 repr()과 유사하나 non-ascii 문자는 escape한다.(?)
id(obj)	객체의 고유번호(int형)을 반환한다.
hash(obj)	객체의 해시값(int형)을 반환. (같은 값이면 해시도 같다.)
chr(num) ord(str)	ASCII 값을 문자로 반환 한 문자의 ASCII 값을 반환
isinstance(obj, className)	객체가 클래스의 인스턴스인지를 판단한다.
issubclass(class, classinfo)	class가 classinfo 의 서브클래스일때 True 반환

내장 함수

열거형의 정보를 얻는 함수

함수명	기능
len(seq)	시퀀스형을 받아서 그 길이를 반환한다.
iter(obj [,sentinel]) next(iterator)	객체의 이터레이터(iterator)를 반환한다. 이터레이터의 현재 요소를 반환하고 포인터를 하나 넘긴다.
enumerate(iterable, start=0)	이터러블에서 enumerate 형을 반환한다. 입력값으로 시퀀스자료형(리스트, 튜플, 문자열)을 입력을 받는다.
sorted(iterable[,key][,reverse])	정렬된 *리스트*를 반환
reversed(seq)	역순으로 된 *iterator*를 반환한다.
filter(func, iterable)	iterable의 각 요소 중 func()의 반환값이 참인 것만을 묶어서 이터레이터로 반환.
map(func, iterable)	iterable의 각 요소를 func()의 반환값으로 매핑해서 이터레이터로 반환.

iterable:반복가능한 자료형 (문자열, 리스트, 튜플, 딕셔너리, 집합)

내장 함수

산술/논리 연산에 관련된 함수

hex(n) oct(n) bin(n)	정수 n의 16진수 값을 구해서 '문자열'로 반환한다. 정수 n의 8진수 값을 구해서 '문자열'로 반환한다. 정수 n의 2진수 값을 구해서 '문자열'로 반환한다.
abs(n)	절대값을 구한다. 복소수의 경우 크기를 구한다.
pow(x,y[,z])	거듭제곱을 구한다. pow(x,y)은 x**y 와 같다.
divmod(a,b)	a를 b로 나눈 (몫, 나머지)를 구한다. 튜플 반환.
all(iterable) any(iterable)	iterable 의 모든 요소가 True 일 경우 True를 반환. iterable 의 하나 이상의 요소가 True 일 경우 True를 반환.
max(iterable) max(arg1, arg2, …)	최대값을 구한다.
min(iterable) min(arg1, arg2, …)	최소값을 구한다.
round()	반올림을 한다.