

Python Language 교육

1차시. 오리엔테이션, 기본 자료형 및 연산

카이스트 문화기술대학원

박사과정 김종수

1. 강사 소개
2. 강의 일정
3. 설문 조사 진행
4. 강의

My Profile

- 이름: 김종수
- KAIST 문화기술대학원 박사과정
- MAC Lab. (Music and Audio Computing Lab.)
 - 딥러닝 기반 방송 모니터링 시스템
 - 실시간 박자 추적 시스템

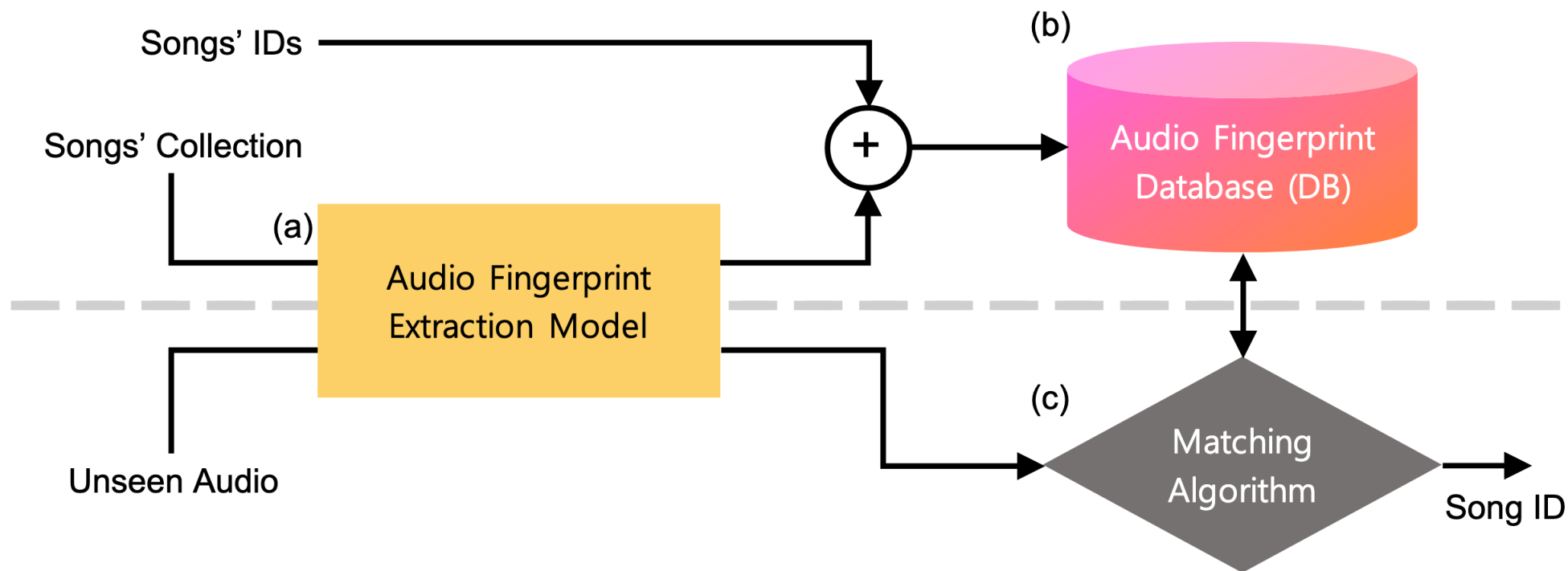
Email: jongsoo.kim@kaist.ac.kr

Web: <https://white-df.github.io/>

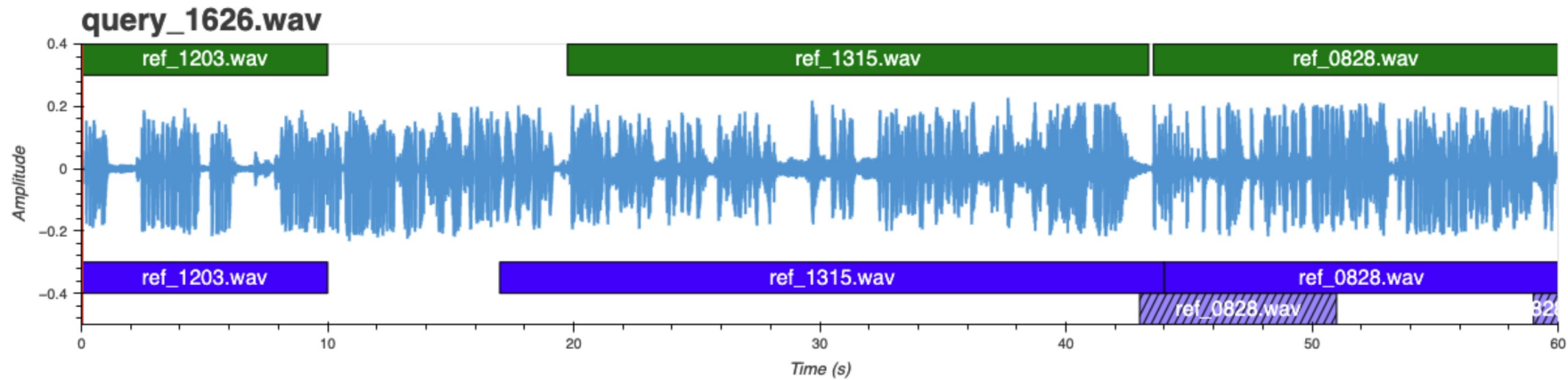
딥러닝 기반 방송 모니터링 시스템

Audio Fingerprint

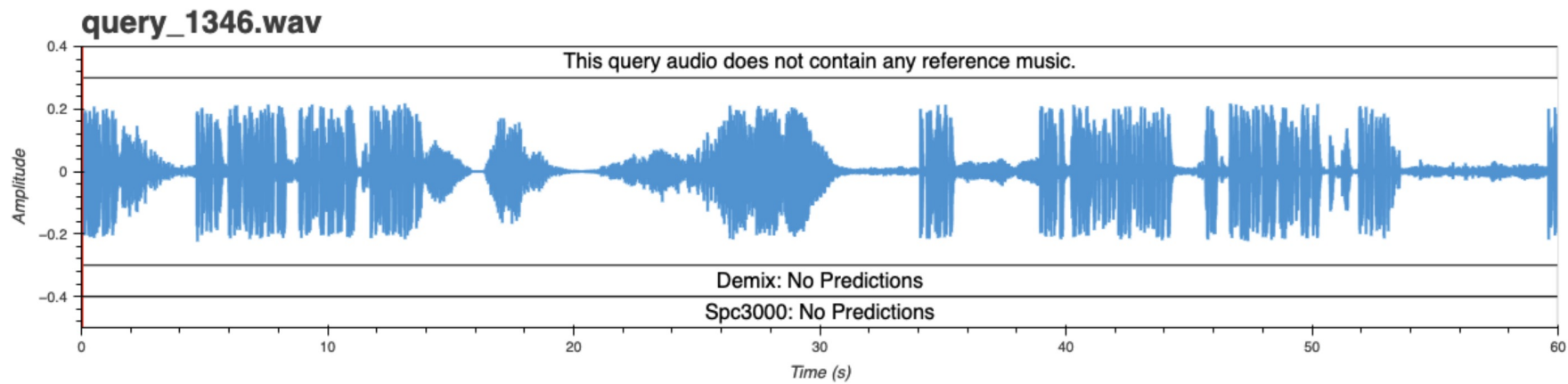
사람을 식별할 때 지문을 사용하듯 음악을 식별할 수 있는 작은 오디오 지문을 만든다면 데이터 저장 공간 절약과 정확한 검색이 동시에 가능. 딥러닝 모델을 통해 음악들로부터 오디오 지문을 추출하고 이를 데이터베이스 형태로 만들어 방송에서 사용된 음악이 무슨 음악인지 검색하는 시스템.



딥러닝 기반 방송 모니터링 시스템



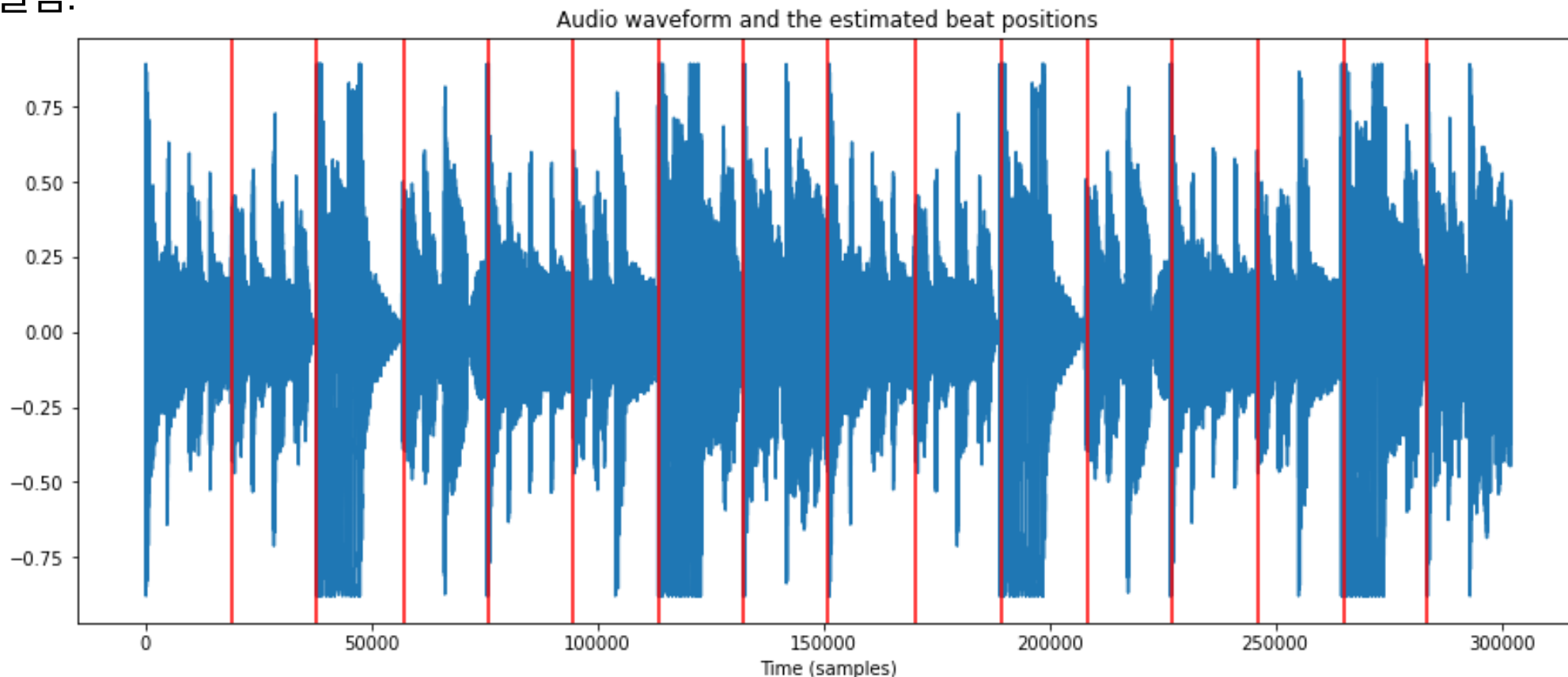
(a)



(b)

Realtime Beat Tracking

박자 추적이란 음악이 입력되었을 때 해당 음악의 박자(beat)와 아랫박자(downbeat)의 위치를 찾아주는 행위를 말함. 실시간 박자 추적 시스템은 오디오가 재생될 때 실시간으로 현재 박자(beat)인지 아랫박자(downbeat)인지 예측하는 시스템을 말함.



What We Will Do

개발 환경: Google Colab

향후 Visual Studio Code 이용 예정

해당 수업은 프로그래밍 초보자를 위한 수업이며 딥러닝 개발에 초점을 맞춘 Python 기초 강의입니다.

컴퓨터적 사고 능력을 키우는 최고의 방법은 프로그래밍을 최대한 많이 해보는 것입니다.

코딩의 재능이 없다고 느끼는 것은 컴퓨터적 사고에 익숙하지 않기 때문입니다.

문법을 빠르게 익힌 후, 다량의 실습 과정을 통해 익숙해지는 것이 목표입니다.

질문은 수업 중간 언제든지!

수업 방식

1. 오전 10:00~11:30 (1.5시간), 오후 13:00~17:00 (4시간) 수업 진행
2. 오프라인 강의 진행
 - 1) 강의자료(ppt, code 등)는 모두 notion 링크를 통해 업로드 할 예정

강의 자료 업로드 링크

<https://rb.gy/f6u72k>



추가 내용

1. 수업이 끝난 후, 질문은 카카오톡을 통해 자유롭게! (아이디: jongsu790)
2. 종강 후에도 ...
 - 1) 원하시는 분들에 한해서 지속적으로 카카오톡을 통해 관련 정보를 공유
 - 2) 언제든지 채팅을 통해 질문하세요!

Our Schedule

1일차	오리엔테이션, 컴퓨터에서의 수, 기본 자료형 및 연산
2일차	조건문과 반복문
3일차	함수, 패키지, 모듈, VS code 설치 및 가상환경
4일차	문자열 다루기, 파일 입출력
5일차	클래스 [객체 지향 프로그래밍]

Our Schedule

기본적인 프로그래밍 가능

1일차	오리엔테이션, 컴퓨터에서의 수, 기본 자료형 및 연산
2일차	조건문과 반복문
3일차	함수, 패키지, 모듈, VS code 설치 및 가상환경
4일차	문자열 다루기, 파일 입출력 및 예외처리
5일차	클래스 [객체 지향 프로그래밍]

Our Schedule

1일차	오리엔테이션, 컴퓨터에서의 수, 기본 자료형 및 연산
2일차	조건문과 반복문 효율적인 프로그래밍과 딥러닝 코드 작성에 필요한 지식
3일차	함수, 패키지, 모듈, VS code 설치 및 가상환경
4일차	문자열 다루기, 파일 입출력 및 예외처리
5일차	클래스 [객체 지향 프로그래밍]

Our Schedule

1일차	오리엔테이션, 컴퓨터에서의 수, 기본 자료형 및 연산
2일차	조건문과 반복문
3일차	함수, 패키지, 모듈, VS code 설치 및 가상환경
4일차	문자열 다루기, 파일 입출력 및 예외처리
5일차	클래스 [객체 지향 프로그래밍]

없어도 딥러닝 코드 작성은 가능하지만
많은 사람들이 사용하고 가장 어려우며 가장 필요한 지식
코드를 효율적으로 유지보수 할 수 있도록 함

Survey

프로그래밍이란?

프로그래밍이란 프로그램을 개발하는 행위를 일컫는다.

여기서 프로그래밍 언어는 컴퓨터가 이해할 수 있는 언어로 컴퓨터에게 특정 명령이나 연산을 수행할 수 있도록 하기 위해 만들어진 언어이다.

Python은 이 중에서도 매우 쉽고 데이터 처리에 능한 언어이다.

C와 C++에 비해서 데이터를 쉽게 처리할 수 있으며 문법이 매우 쉬워 프로그래밍을 처음 입문하는 사람들에게 매우 적합한 언어이다.

연산 속도가 C와 C++에 비해 느림에도 불구하고 데이터를 매우 쉽게 처리할 수 있어 딥러닝과 머신러닝에서 필수적으로 사용하는 언어가 되었다.

개발자에게 “솔직히” 프로그래밍이란?

주어진 문제의 해답을 제시하는 프로그램을 코딩을 통해서 만들어내는 것

그러나 프로그래밍을 ”잘 한다”라는 것은 문법에 능한 것이 아니라

- 1) 모르는 내용을 구글링을 통해 잘 찾아내고 이를 자신의 코드에 잘 적용하는 것
- 2) 문제가 주어졌을 때 컴퓨터적 사고 능력으로 코드를 잘 구성할 수 있는 것

→ 기본 문법을 익힌 후, 순서도와 실전 코딩 연습을 통해 컴퓨터적 사고 능력을 향상

→ 이를 진행하면서 구글링 검색 방식도 함께 익숙해지도록 할 예정

프로그래밍은 수학?!

프로그래밍은 완벽히 수학이라 할 수 없지만 수학 문제 풀이와 비슷한 구조입니다.


종수가 사과 5개를 가지고 있다. 종수는 어머니에게 사과를 3개 더 추가적으로 받게 되었다. 그렇다면 종수가 가지게 될 사과의 개수는?

Solution)

종수가 가진 사과의 개수를 a 라고 하면 $a=5$ 이다.

사과를 3개 더 받았으므로 종수가 가지게 될 사과의 개수는 $a+3$ 이다.

따라서, $a+3 = 5+3 = 8$ 개 이다.



```
a = 5
# a는 5라는 정수를 담는 변수

answer = a + 3
# answer는 a+3의 결과값을 담는 변수

print(answer)
# print는 괄호 안의 값을 콘솔에 출력하는 함수
```

8

변수(Variable)는 데이터를 담는 상자

자료형은 변수나 함수의 형태

int : 정수형 (-1, -100, 0, 2, 30 등)

float : 실수형 (소수점이 있는 값)

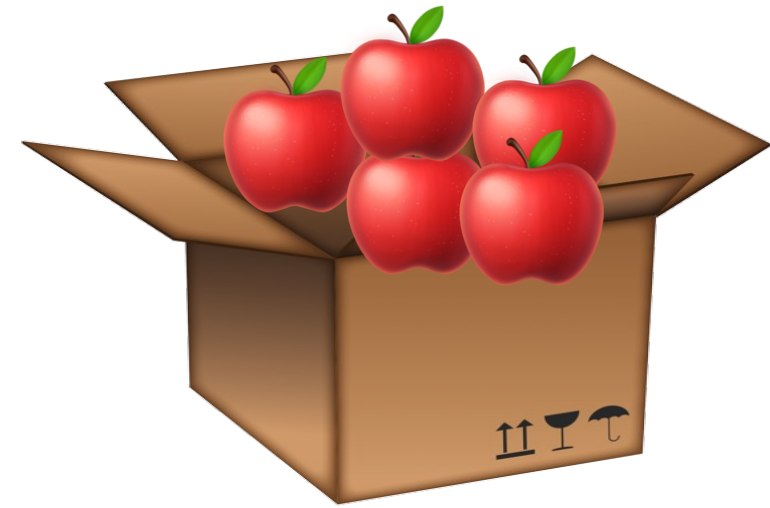
string: 문자열

함수는 연산들의 단축키

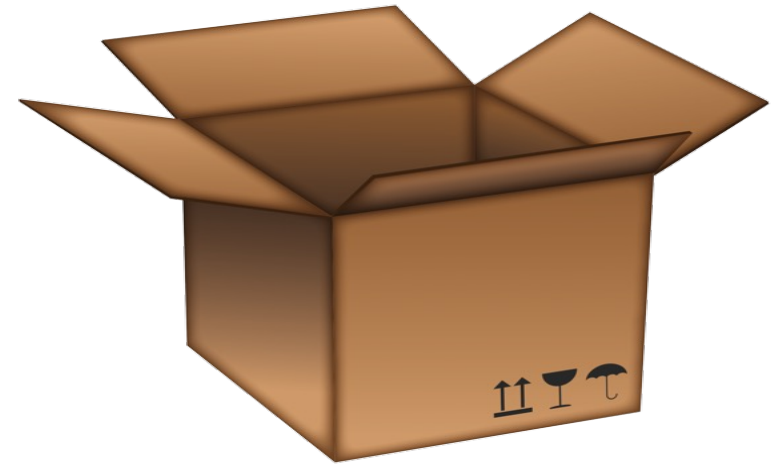
print 등

주석은 코드가 아닌 메모

Variable

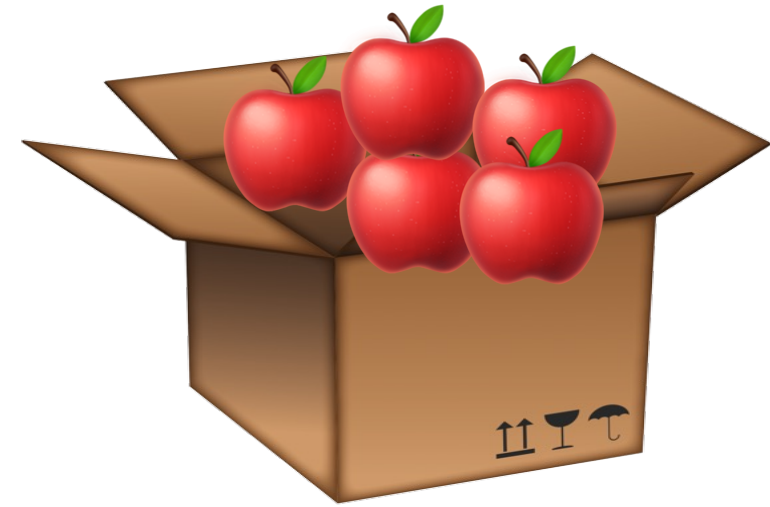


a

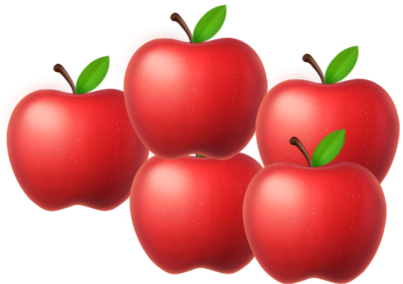


answer

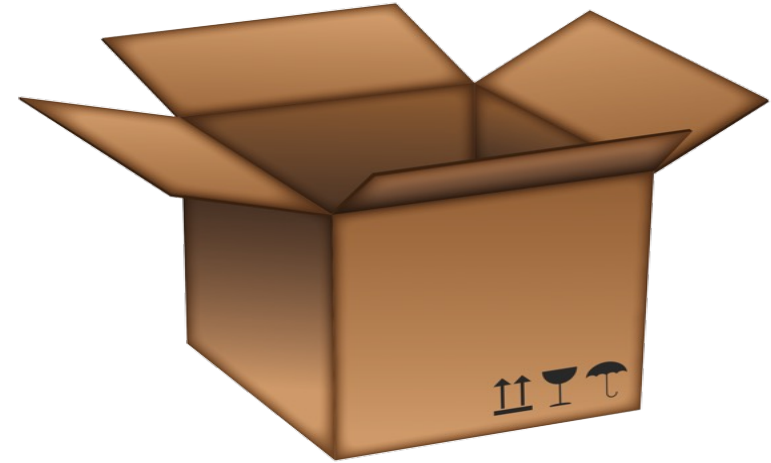
Variable



a

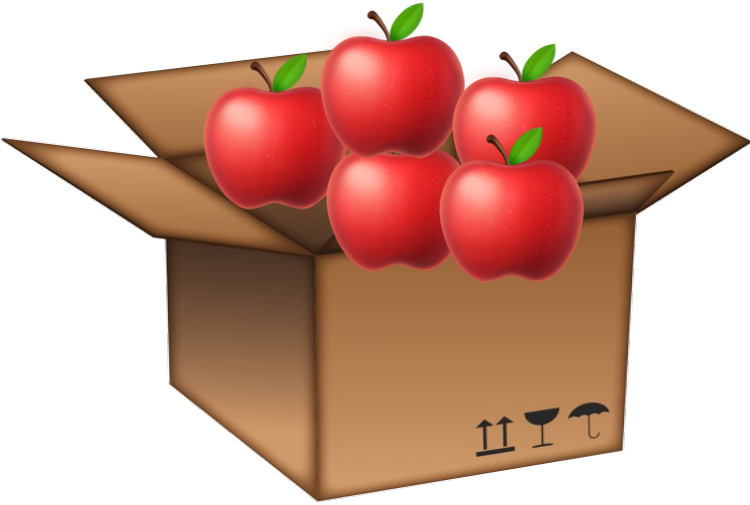


a 상자에 든 사과 개수만큼
똑같이 사과를 준비

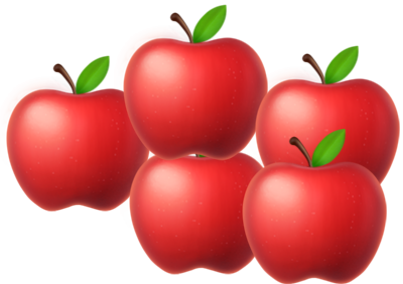


answer

Variable

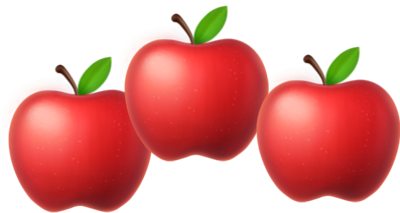


a



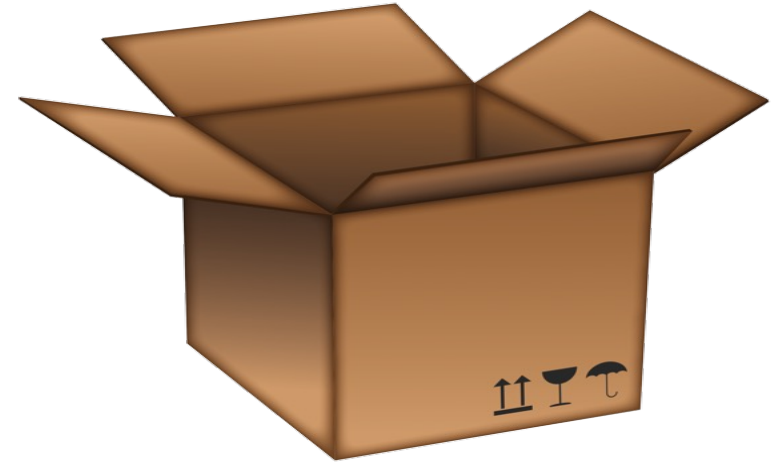
a

+



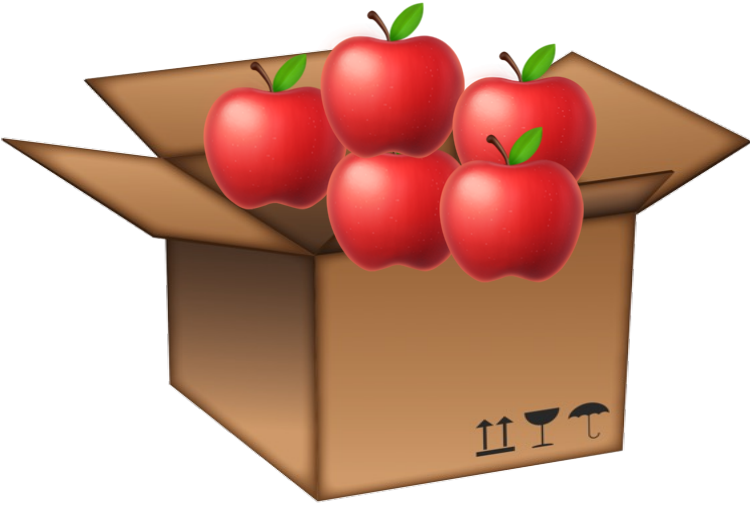
3

+3 연산을 위해
사과 3개 더 준비

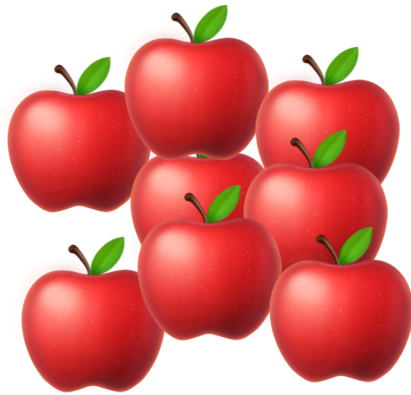


answer

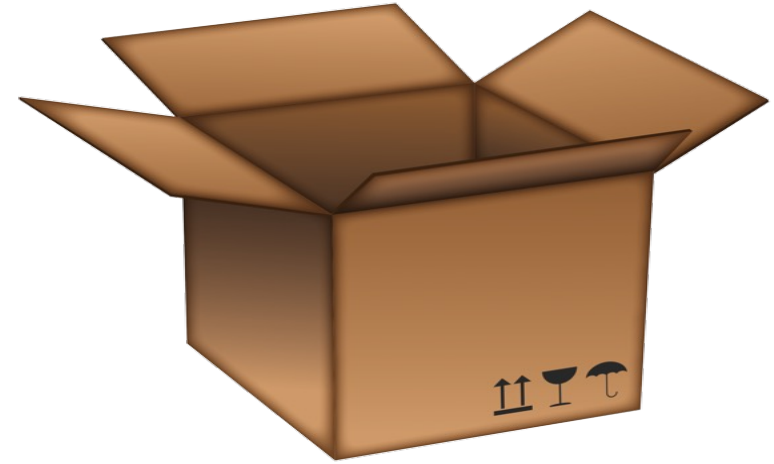
Variable



a

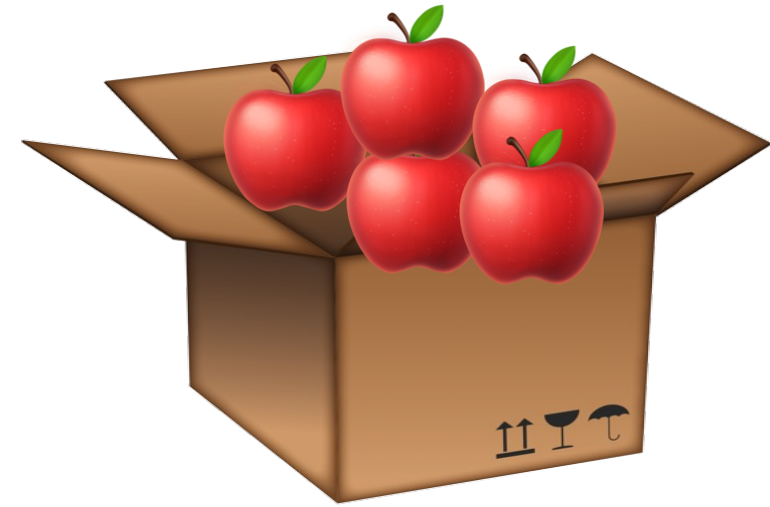


총 8개의 사과



answer

Variable



a

$$\text{answer} = a + 3$$



answer

answer라는 상자에
사과 8개를 담기

자료형 (Data Type)

Python에는 다음과 같은 기본 자료형이 있음.

수와 관련된 자료형

- 정수형 (`int`) (-100, 200, 5, 0, -3 등과 같이 소수점 없이 딱 떨어지는 수)
- 실수형 (`float`) (-100.03, 30.23, 0.0, -1.0 등과 같이 소수점이 존재하는 수)

문자열과 관련된 자료형

- 문자열 (`str`)
"String", '안녕' 과 같이 큰따옴표 `"` 혹은 작은따옴표 `'` 안에 문자로 구성된 집합을 의미)

자료형 (Data Type)

C 계열의 언어는 Python과 달리 변수와 함수를 사용하기 전에 먼저 그것의 자료형을 선언해야 함.

컴퓨터에 미리 변수의 자료형을 선언함으로써 컴퓨터의 메모리를 보호

C 계열의 언어에는 다음과 같은 기본 자료형이 있음.

- 정수형
- 실수형
- 문자

각 자료형 안에도 다양한 자료형이 존재함.

정수형 (Integer Data Type)

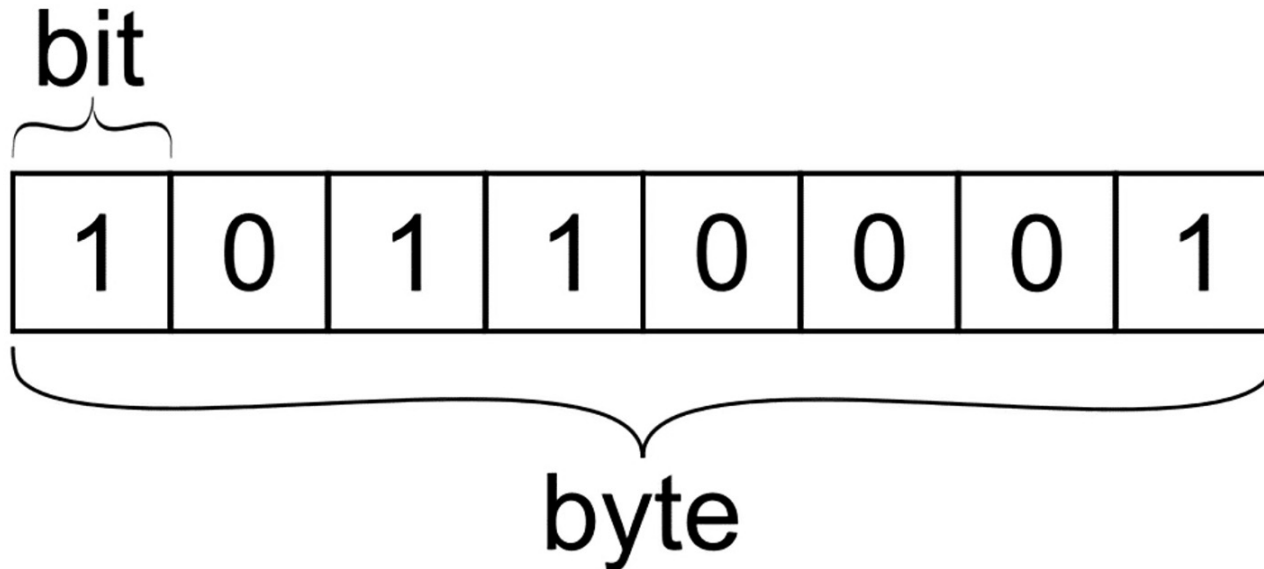
자료형			크기 (byte)	범위
정수	signed	short	2	-32,768 ~ 32,767
		int	4	-2,147,483,648 ~ 2,147,483,647
		long	4	-2,147,483,648 ~ 2,147,483,647
		long long	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
	unsigned	unsigned short	2	0 ~ 65,535
		unsigned int	4	0 ~ 4,294,967,295
		unsigned long	4	0 ~ 4,294,967,295
		unsigned long long	8	0 ~ 18,446,744,073,709,551,615

비트와 바이트 (bit & byte)

컴퓨터는 기본적으로 0과 1을 담는 가장 작은 단위인 비트(bit)가 존재

한 비트에는 0 아니면 1의 값을 가짐

비트가 8개 모이면 1 바이트가 됨 (1 byte = 8 bits)

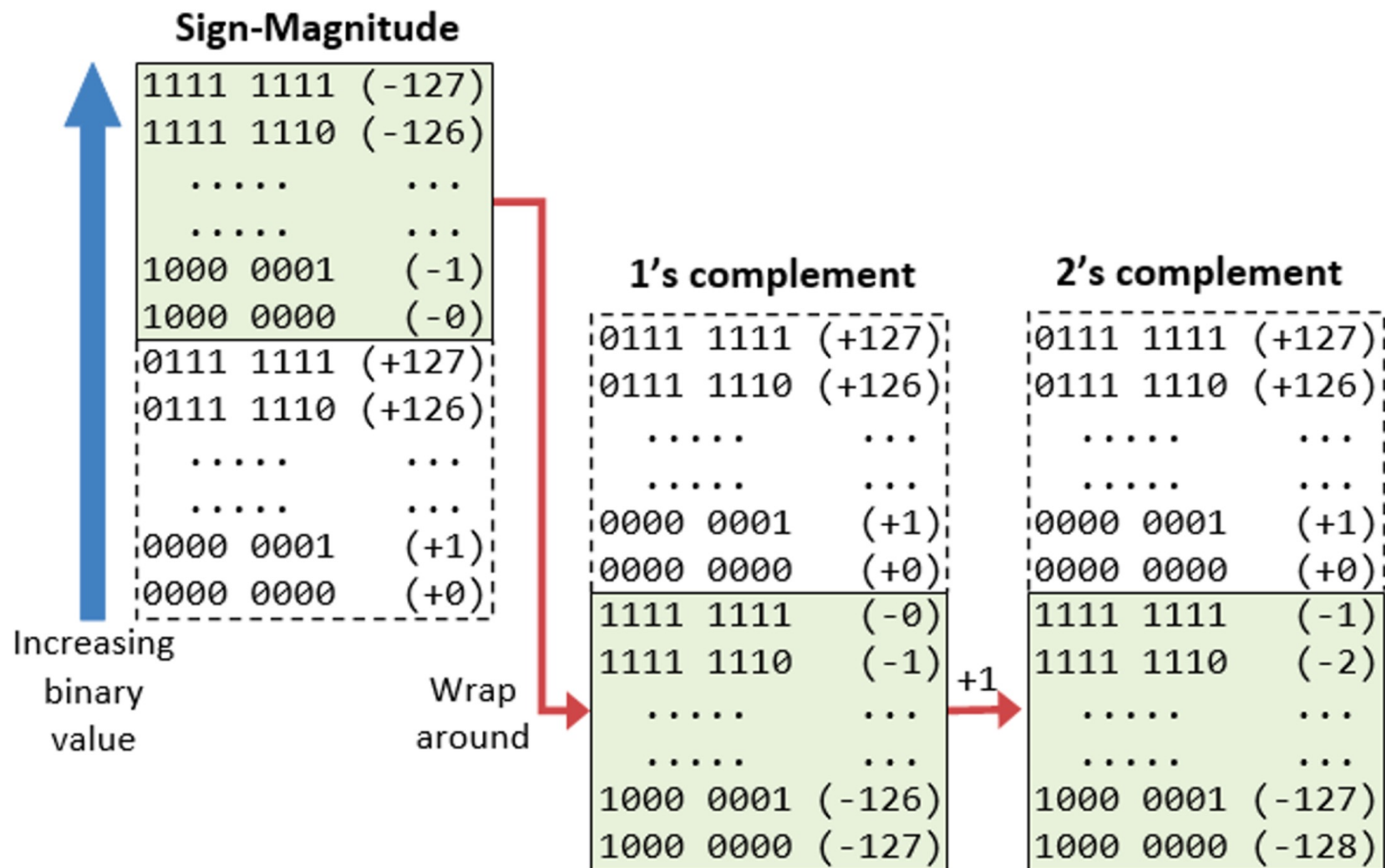


이진법 계산 (Binary Number)

	2진수						
자릿수	2^6	2^5	2^4	2^3	2^2	2^1	2^0
숫자표현	0	0	0	0	0	0	0

<input type="checkbox"/> 0	as 0000 0000
<input type="checkbox"/> 1	as 0000 0001
<input type="checkbox"/> ...	
<input type="checkbox"/> 254	as 1111 1110
<input type="checkbox"/> 255	as 1111 1111

음수 계산법 (Negative Numbers)

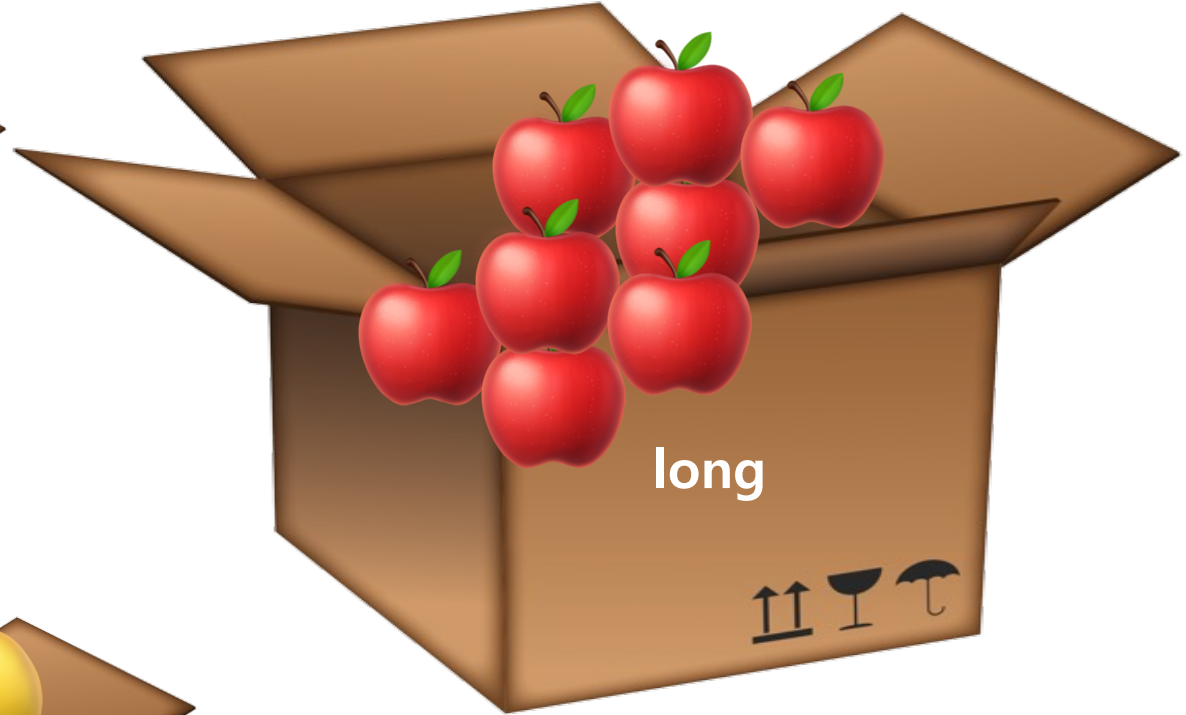
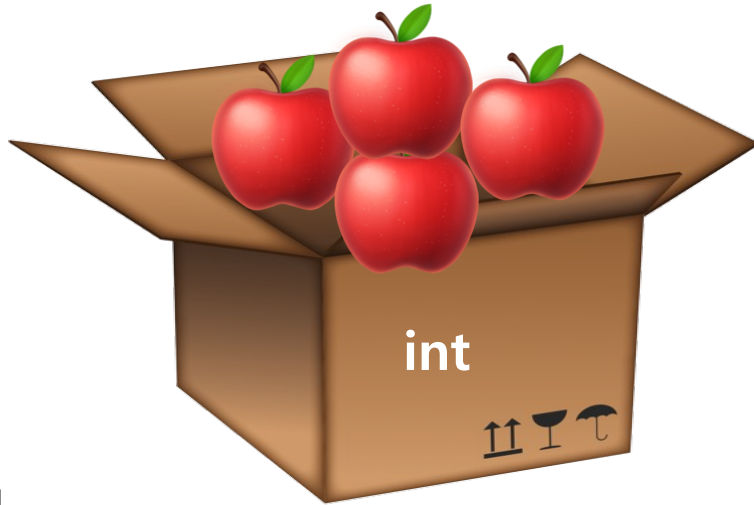
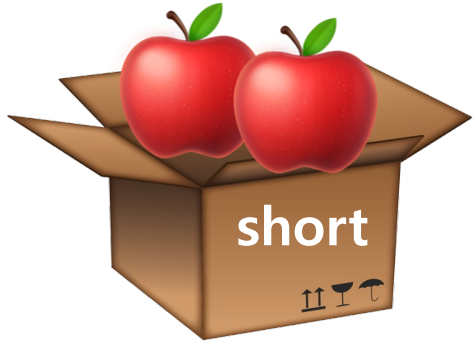


Key Point: 자료형마다 표현할 수 있는 수의 범위가 한정적!!!

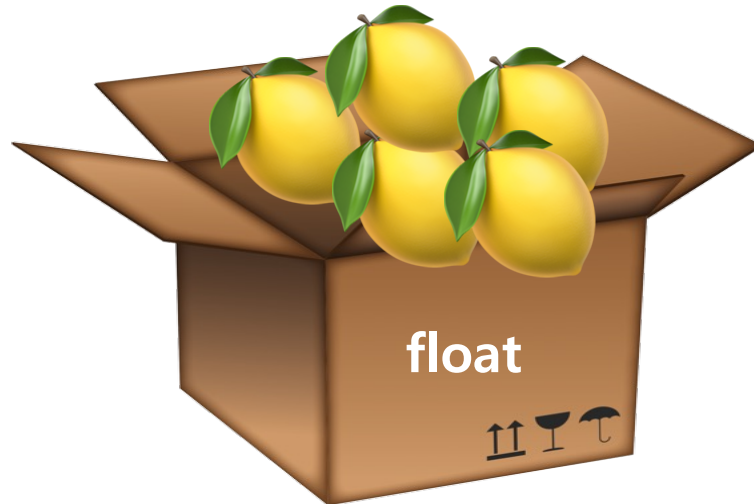
정수형 (Integer Data Type)

자료형			크기 (byte)	범위
정수	signed	short	2	-32,768 ~ 32,767
		int	4	-2,147,483,648 ~ 2,147,483,647
		long	4	-2,147,483,648 ~ 2,147,483,647
		long long	8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
	unsigned	unsigned short	2	0 ~ 65,535
		unsigned int	4	0 ~ 4,294,967,295
		unsigned long	4	0 ~ 4,294,967,295
		unsigned long long	8	0 ~ 18,446,744,073,709,551,615

Data Type



- 변수는 과일을 담은 상자
- 자료형은 상자가 무슨 과일
을 최대 몇개 담을 수 있는지
(상자의 종류)
- 변수에 담긴 데이터는 상자
에 담긴 과일들

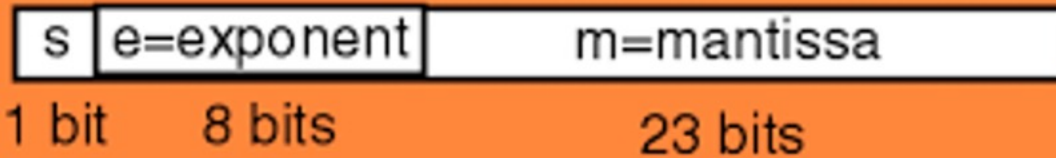


Key Point: 정확한 값이 아닌 근사값을 사용

실수형 (Floating Point Data Type)

자료형		크기 (byte)	범위	유효 자리수
실수	float	4	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$	소수점 이하 6자리
	double	8	$-1.8 \times 10^{308} \sim 1.8 \times 10^{308}$	소수점 이하 15자리
	long double	8	$-1.8 \times 10^{308} \sim 1.8 \times 10^{308}$	소수점 이하 15자리

IEEE 754 Floating Point Standard



$$\text{number} = (-1)^s * (1.m) * 2^{e-127}$$

기본적으로 컴퓨터에서는
실수형 데이터들을
정확한 값이 아닌
*근사값을 사용함

문자형 (Character Data Type)

```
char a = 'a';
```

```
char a = 97;
```

1byte -128~127, 0~255

이미지 픽셀 당 가지는 값을 intensity

수와 관련된 연산 (Operation)

Python에서 수와 관련된 기본 연산은 다음과 같다.

- 대입 연산자
- 덧셈, 뺄셈
- 곱셈, 나눗셈, 몫, 나머지, 거듭제곱 연산
- 할당 연산자

대입 연산자

변수의 특정 값을 대입하는 연산자 (=)

수학에서 '='은 equal의 의미를 가지고 있지만 프로그래밍 언어에서는 조금은 다른 의미

`a=5` 는 a라는 변수에 5라는 값을 대입한다는 의미

덧셈, 뺄셈

수를 더하거나(+) 빼는(-) 연산자

`a=5+3;` 은 a라는 변수에 5+3 연산을 수행한 결과값을 대입한다는 의미

곱셈, 나눗셈, 몫, 나머지, 거듭제곱

곱셈과 나눗셈 연산은 '*', '/'을 이용해 덧셈, 뺄셈과 동일하게 사용

몫 연산(//)은 몫을 반환하고 나머지 연산(%)는 나머지 값을 반환함

거듭제곱 연산(**)은 앞에 나오는 수를 뒤에 나오는 수만큼 거듭제곱 연산함

예시)

$5 + 2 \triangleright 7$	$5 - 2 \triangleright 3$	$5 * 2 \triangleright 10$	$5 / 2 \triangleright 2.5$
$5 // 2 \triangleright 2$	$5 \% 2 \triangleright 1$	$5 ** 2 \triangleright 25$	

할당 연산자

`+=`, `-=`, `*=`, `/=`, `%=` 등 다양하게 존재

해당 변수에 할당된 값에 특정 산술 연산을 시행한 후 결과값을 다시 해당 변수에 재할당

예를 들어,

- `a += 5;`
- `a = a + 5;`

위 두 식은 동일한 의미를 가진다

실습 활동 (Practice Time)

실습 활동을 통해 수와 관련된 연산을 연습해보자

구글 Colab을 이용해 코드 작성 시작!

문자열과 관련된 연산 (Operation)

Python에서 문자열과 관련된 기본 연산은 다음과 같다.

- 대입 연산자
- 덧셈, 곱셈
- 할당 연산자
- 인덱싱
- 슬라이싱

대입 연산자

수의 대입 연산자와 동일한 의미로 `a="hello"` 는 a라는 변수에 hello라는 값을 대입한다는 의미

덧셈, 곱셈

두 문자열을 연결하거나 혹은 문자열을 반복시키는 연산

예시)

<code>"Hello" + "World" ➤ "HelloWorld"</code>	<code>"Hello" * 3 ➤ "HelloHelloHello"</code>
---	--

할당 연산자

`+=`, `*=` 등이 존재하며 수에서의 할당 연산자와 방식은 동일

인덱싱 (Indexing)

문자열에서 인덱스는 문자의 위치 정보를 의미한다. 인덱스라는 것을 이용해 우리는 문자열이 특정 위치에서 어떤 값을 가지고 있는지 알 수 있다. 문자열의 가장 첫 번째 문자는 인덱스 값이 0이며 마지막 값은 문자열의 길이에서 1을 뺀 값이다.

<code>a = "Hello, World"</code>			
<code>a[0] ➤ "H"</code>	<code>a[6] ➤ " "</code>	<code>a[-1] ➤ "d"</code>	<code>a[-3] ➤ "r"</code>

슬라이싱 (Slicing)

문자열의 인덱스 정보를 이용해 문자열의 일부분만 추출할 수 있다. 시작 인덱스 값과 끝 인덱스 값만 안다면 해당 인덱스 값들에 맞는 문자열의 일부를 추출할 수 있다. 인덱싱과 유사함.

예시)

<code>a = "Hello, World"</code>			
<code>a[1:4] ➤ "ell"</code>	<code>a[5:] ➤ ", World"</code>	<code>a[:6] ➤ "Hello,"</code>	<code>a[10:-1] ➤ "l"</code>

문자열과 관련된 함수 (Function)

Python에서 문자열과 관련된 함수는 다음과 같다.

`len (문자열)`

: 소괄호 내에 입력된 문자열의 길이를 반환한다.

예시)

<code>a = "Hello, World"</code>			
<code>len(a) ➤ 12</code>	<code>len(a[1:8]) ➤ 7</code>	<code>len(a[8:]) ➤ 4</code>	<code>len("hello") ➤ 5</code>

실습 활동 (Practice Time)

실습 활동을 통해 문자열과 관련된 연산 및 함수를 연습해보자

구글 Colab을 이용해 코드 작성 시작!

배열 자료형 (Array Data Type)

Python에는 다음과 같은 배열 관련 자료형이 있음.

- 리스트 (`list`)
 - `[0, 2, 3, 1]` 과 같이 대괄호를 이용해 작성
 - 리스트 내 원소들 간에는 순서가 존재 (원소 중복 가능)
 - 원소 수정, 삭제, 변경 가능 (가변)
- 튜플 (`tuple`)
 - `(0, 2, 3, 1, 0)` 과 같이 소괄호를 이용해 작성
 - 튜플 내 원소들 간에는 순서가 존재 (원소 중복 가능)
 - 원소 수정, 삭제, 변경 불가능 (불변)

배열 자료형 (Array Data Type)

Python에는 다음과 같은 배열 관련 자료형이 있음.

- 집합 (`set`)
 - `{0, 2, 3, 1}` 과 같이 중괄호를 이용해 작성
 - 리스트 내 원소들 간에는 순서가 존재하지 않음 (원소 중복 불가능)
 - 원소 수정, 삭제, 변경 가능 (가변)
- 딕셔너리 (`dict`)
 - `{1:"a", 2:"b", 3:"c", 4:"d"}` 과 같이 중괄호를 이용해 작성
 - `key:value` 형태로 원소가 구성되어 있음
 - 원소 수정, 삭제, 변경 가능 (가변)
 - `key`를 통해 `value` 값 접근 가능

리스트와 관련된 연산 (Operation) 및 함수(Function)

Python에서 리스트와 관련된 기본 연산과 함수는 다음과 같다.

- 연산
 - 대입 연산자, 덧셈, 곱셈, 할당 연산자, 인덱싱, 슬라이싱
- 함수
 - `.append()` : 리스트 맨 뒤에 새로운 값을 추가
 - `.sort()` & `.reverse()` : 리스트를 정렬 & 역순으로 정렬
 - `.insert()` & `.remove()` & `.pop()` : 리스트 특정 인덱스에 값 추가 & 리스트 특정 값 제거 & 리스트 특정 인덱스의 값 추출
 - `.index()` & `.count()` : 리스트 내 특정 값의 인덱스 값 반환 & 리스트 내 특정 값 개수 세기
 - `len()` : 리스트 길이 반환
 - `sorted()` : 리스트의 정렬된 버전을 반환
 - `del` : 값을 제거
 - `.clear()` : 모든 원소 제거

튜플과 관련된 연산 (Operation) 및 함수(Function)

Python에서 튜플과 관련된 기본 연산과 함수는 다음과 같다.

- 연산
 - 대입 연산자, 덧셈, 곱셈, 인덱싱, 슬라이싱
- 함수
 - len() : 튜플의 길이 반환
 - list() : 튜플을 리스트로 변환
 - .index() & .count() : 리스트 내 특정 값의 인덱스 값 반환 & 리스트 내 특정 값 개수 세기
 - del : 값을 제거
 - sorted() : 튜플의 정렬된 버전을 리스트로 반환

집합과 관련된 연산 (Operation) 및 함수(Function)

Python에서 집합과 관련된 기본 연산과 함수는 다음과 같다.

- 연산
 - 대입 연산자, 할당 연산자, 합집합 (|), 교집합 (&), 차집합 (-), 대차집합 (^)
- 함수
 - .union() & .intersection & .difference() & .symmetric_difference() : 합집합 & 교집합 & 차집합 & 대차집합
 - len() : 집합 원소 개수 반환
 - .add() : 특정 원소 추가
 - .pop() : 임의의 원소 추출
 - .remove() & .discard() : 제거(없으면 에러) & 제거(없으면 패스)
 - del : 값을 제거
 - .clear() : 모든 원소 제거

딕셔너리와 관련된 연산 (Operation) 및 함수(Function)

Python에서 딕셔너리와 관련된 기본 연산과 함수는 다음과 같다.

- 연산
 - 대입 연산자
- 함수
 - .keys() & .values() : key 값들 반환 & value 값들 반환
 - .items() : key & value의 pair들을 반환
 - .get() : key 값을 통해 사전 내 일치하는 원소 반환
 - del : 값을 제거
 - .clear() : 사전 내 모든 원소들 삭제

실습 활동 (Practice Time)

실습 활동을 통해 배열 자료형과 관련된 연산 및 함수를 연습해보자

구글 Colab을 이용해 코드 작성 시작!