

# Lecture 02

Python Basic Syntax for Data Processing

## 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

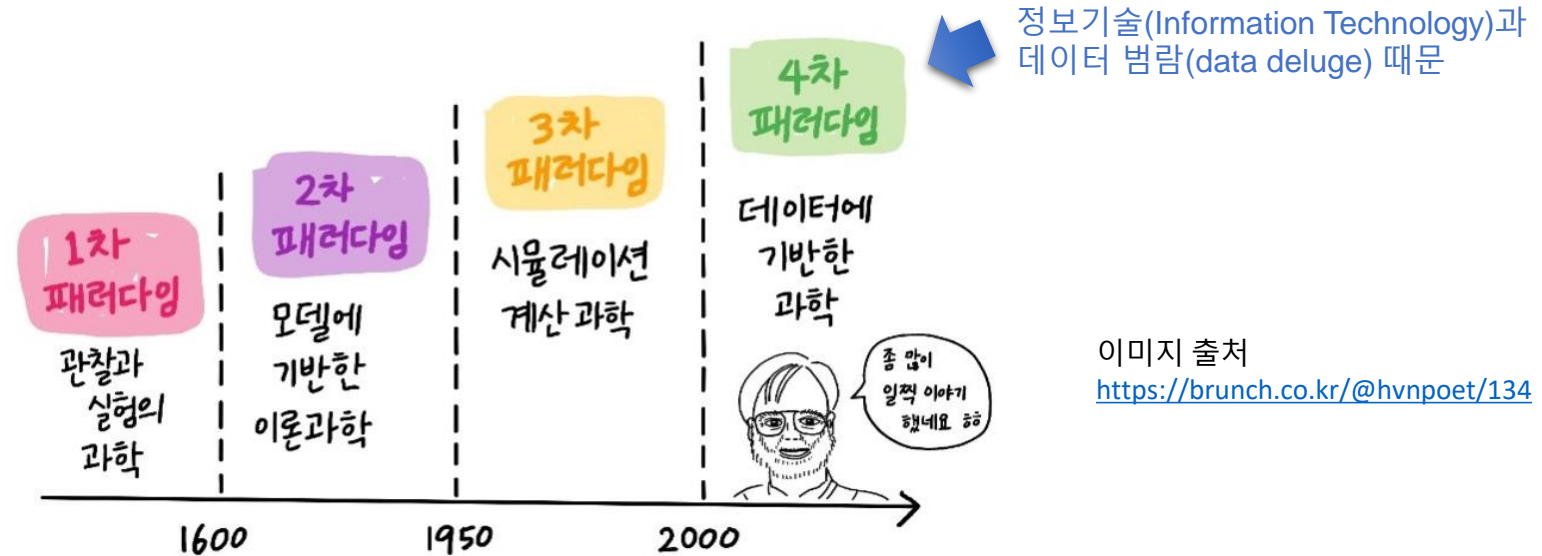
- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## ■ Data Science?

- Structured data(정형 데이터), Unstructured data(비정형 데이터)를 포함한 다양한 데이터로부터 knowledge와 insight를 추출하는 convergence 분야
- 1998년, Turing(튜링) 상을 수상한 Jim Gray 박사는 과학의 4번째 Paradigm으로 **Data Science**를 정의



# Data의 표현 단위

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

기호	데이터 크기 (10진수)	데이터 크기 (2진수)
KB (Kilo Byte)	$1000^1 = 10^3$	$1024^1 = 2^{10}$
MB (Mega Byte)	$1000^2 = 10^6$	$1024^2 = 2^{20}$
GB (Giga Byte)	$1000^3 = 10^9$	$1024^3 = 2^{30}$
TB (Tera Byte)	$1000^4 = 10^{12}$	$1024^4 = 2^{40}$
PB (Peta Byte)	$1000^5 = 10^{15}$	$1024^5 = 2^{50}$
EB (Exa Byte)	$1000^6 = 10^{18}$	$1024^6 = 2^{60}$
ZB (Zetta Byte)	$1000^7 = 10^{21}$	$1024^7 = 2^{70}$
YB (Yotta Byte)	$1000^8 = 10^{24}$	$1024^8 = 2^{80}$

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## ■ Data의 증가

- 현재 연간 16.3 zetta byte의 데이터를 생산  
(zetta byte는 10억 tera byte에 해당)
- IDC(International Data Corporation)와 Seagate의 조사에 따르면  
2025년에는 163 zetta byte(163,000,000,000,000,000,000,000 byte)로 증가
- 향후 데이터 양을 yotta byte( $10^{24}$  byte)로 표기하게 될 것이라 예상

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## ■ Big Data 정의

- Big Data의 사이즈는 단일 데이터 집합의 크기가 수십 Tera byte에서 수 Peta byte에 이르며, 그 크기가 끊임없이 변화하는 것이 특징
- Big Data라는 용어는 1990년대부터 사용되어 왔으며, John Mashey(존 매쉬)가 대중화 함

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## ■ Big Data의 3V 특징

### ● Variety(다양성)

- ✓ Big Data는 다양한 형태의 데이터를 포함
- ✓ Structured Data뿐만 아니라, 사진, 오디오, 비디오 소셜 미디어 데이터, 위치 데이터, 로그 파일 등과 같은 Unstructured data까지 포함

### ● Velocity(속도)

- ✓ Big Data는 데이터의 생성 속도가 매우 빠름
- ✓ Big Data는 새로 생성되고, 저장되며, 시각화 과정이 매우 빠르게 이루어짐

### ● Volume(양)

- ✓ Big Data는 저장되는 물리적 데이터 양이 매우 많음
- ✓ Big Data의 가장 기본적인 특징에 해당함

## 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## ■ Internet Search

- 키워드 기반의 인터넷 검색은 Algorithm을 사용하여 몇 초 안에 최상의 검색 결과를 제공함

## ■ Digital Advertisement

- 디지털 광고는 디스플레이 배너부터 디지털 광고판까지의 모든 디지털 마케팅 스펙트럼을 Algorithm을 사용하여 개인별 맞춤형 광고를 제공함

## ■ Recommender systems

- Algorithm을 사용하여 쇼핑몰에서 소비자의 이전 검색 기록 데이터를 기반으로 수십억 개의 제품에서 관련 상품을 쉽게 찾아 추천하고 소비자 경험을 더욱 풍성하게 제공함

## 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## ■ Data Scientist가 되기 위한 필요한 요소

- Programming Language 활용 능력이 필요함
  - ✓ Java, C언어, Perl과 함께 Data Science자들에게 가장 대중적으로 쓰이는 **Python** 코딩 능력
  - ✓ SAS, R에 대한 깊이 있는 지식
  - ✓ SQL 기반의 데이터베이스 관리
- Unstructured data 처리
  - ✓ Data Scientist에게 Unstructured data를 다룰 수 있는 능력이 매우 중요해지고 있음
- 데이터 분석과 시각화 기술
- 통계적 기술 및 수학 분야의 이해 능력과 원활한 의사 소통 방법이 필요
- Machine Learning 및 Deep Learning 기술이 필요



# Python

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

- 1991년, Guido van Rossum(귀도 반 로섬)이 발표한 high-level programming language
- Interpreter 방식
- Object-oriented Conversational Language
- Guido van Rossum
  - 2005년 ~ 2012년 : [Google](#)에서 근무
  - 2013년 ~ 2019년 10월 : [Dropbox](#)에서 근무
  - 2020년 11월 13일 ~ 현재 : [MS](#)에서 근무 중



# Variable

## 1. Data Science

## 2. Python Basics

## 2.1. Variable

## 2.2. Operator

## 2.3. print()

## 2.4. input()

## 2.5. type()

## 3. Numerical data

## 3.1. int() / float()

## 3.2. round()

## 3.3. trunc()

## 3.4. ceil()

## 3.5. floor()

## 4. String data

## 4.1. lower() / upper()

4.2. lstrip() / rstrip()  
/ strip()

## 4.3. replace()

## 4.4. split()

## 4.5. type() / sizeof()

## 4.6. len()

4.7. String processing  
using operators

## 5. List

## 5.1. append() / insert()

## 5.2. del() / remove()

## 5.3. sort() / reverse()

## 6. Ascii code

## 6.1. ord() / chr()

변수이름은 반드시 영문자나 언더바( \_ )로 시작해야 하고, 숫자나 특수문자로 시작하면 안됩니다.

단, 숫자나 특수문자를 변수이름으로 사용하는 것은 가능해요.

대문자와 소문자는 다른 글자로 생각합니다.

파이썬에서 이미 사용되고 있는 단어들(reserved words, 예약어라고 합니다)은 변수 이름으로 사용할 수 없습니다.

- 변수 사용법과 입출력 배우기

1) var1 이름의 변수를 만들고 10의 값을 할당하기

```
1 var1 = 10
2 print(var1)
```

10

2) var2 이름의 변수를 만들고 20의 값을 할당하기

```
1 var2 = 20
2 print(var2)
```

20

# Operator

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip()

#### 4.3. strip()

#### 4.4. replace()

#### 4.5. split()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

기호	설명	사용 예
+	두 수를 더한 값	$a + b$
-	두 수를 뺀셈한 값	$a - b$
*	두 수를 곱한 값	$a * b$
/	두 수를 나눈 실수형 몫에 해당하는 값	$a / b$
//	두 수를 나눈 정수형 몫에 해당하는 값	$a // b$
%	두 수를 나눈 나머지 값	$a \% b$

# Operator

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip() / strip()

#### 4.3. replace()

#### 4.4. split()

#### 4.5. type() / sizeof()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

### 3) 다양한 연산자 사용하기

```
1 print(var1 + var2)
2 print(var1 * var2)
3 print(var1 / var2)
```

30

200

0.5

### 4) % 연산자 활용하기

```
1 var5 = 3
2 var6 = 14
3 print(var6 % var5)
```

2

# print()

## 1. Data Science

## 2. Python Basics

## 2.1. Variable

## 2.2. Operator

## 2.3. print()

## 2.4. input()

## 2.5. type()

## 3. Numerical data

## 3.1. int() / float()

## 3.2. round()

## 3.3. trunc()

## 3.4. ceil()

## 3.5. floor()

## 4. String data

## 4.1. lower() / upper()

## 4.2. lstrip() / rstrip()

## / strip()

## 4.3. replace()

## 4.4. split()

## 4.5. type() / sizeof()

## 4.6. len()

## 4.7. String processing using operators

## 5. List

## 5.1. append() / insert()

## 5.2. del() / remove()

## 5.3. sort() / reverse()

## 6. Ascii code

## 6.1. ord() / chr()

함수 이름	설명	사용 예
print( )	변수나 데이터를 화면에 출력합니다	print("출력할 문장") print(출력할 변수)

## 5) print( ) 함수 사용하기

```

1  # 출력할 문장을 직접 쓰기
2  print("오늘도 굿모닝이지 말입니다!")
3
4  # 출력할 문장을 변수에 넣고 변수값 출력하기
5  str = "내일도 굿모닝이지 말입니다!"
6  print(str)

```

오늘도 굿모닝이지 말입니다!

내일도 굿모닝이지 말입니다!

# print()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip() / strip()

#### 4.3. replace()

#### 4.4. split()

#### 4.5. type() / sizeof()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

형식	의미
%d	정수(1,2,3,4,5 등)
%f	실수(1.234 등)
%s	문자열('this is string') 및 숫자
%c	문자('a','c' 등 문자 하나)

## 6) 리터럴 문자와 함께 사용하기

```
1 hungry = 5
2 print(' 배가 무지 고파서 밥을 %s 그릇도 먹겠네!' %hungry)
```

배가 무지 고파서 밥을 5 그릇도 먹겠네!

# print()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip()

#### / strip()

#### 4.3. replace()

#### 4.4. split()

#### 4.5. type() / sizeof()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

### 7) 특수문자와 함께 사용해야 할 경우

```
1 up = 20
2 print(' 뉴스에서 물가가 %s% 까지 오른다는데....' %up)
```

**TypeError** Traceback (most recent call last)

<ipython-input-21-b4746ac92cd6> in <module>

```
1 up = 20
```

```
----> 2 print(' 뉴스에서 물가가 %s% 까지 오른다는데....' %up)
```

**TypeError**: not enough arguments for format string

```
1 up = 20
2 print(' 뉴스에서 물가가 %s%% 까지 오른다는데....' %up)
```

뉴스에서 물가가 20% 까지 오른다는데....

## 1. Data Science

## 2. Python Basics

## 2.1. Variable

## 2.2. Operator

## 2.3. print()

## 2.4. input()

## 2.5. type()

## 3. Numerical data

## 3.1. int() / float()

## 3.2. round()

## 3.3. trunc()

## 3.4. ceil()

## 3.5. floor()

## 4. String data

## 4.1. lower() / upper()

4.2. lstrip() / rstrip()  
/ strip()

## 4.3. replace()

## 4.4. split()

## 4.5. type() / sizeof()

## 4.6. len()

4.7. String processing  
using operators

## 5. List

## 5.1. append() / insert()

## 5.2. del() / remove()

## 5.3. sort() / reverse()

## 6. Ascii code

## 6.1. ord() / chr()

## 8) 여러개의 변수를 한꺼번에 사용할 경우

```
1 up = 20
2 sal = 10
3
4 print('뉴스에서 물가는 %s%% 오른다는데 내 월급도 %s%%라도..' %(up , sal))
```

뉴스에서 물가는 20% 오른다는데 내 월급도 10%라도..



# input()

## 1. Data Science

## 2. Python Basics

2.1. Variable

2.2. Operator

2.3. print()

2.4. input()

2.5. type()

## 3. Numerical data

3.1. int() / float()

3.2. round()

3.3. trunc()

3.4. ceil()

3.5. floor()

## 4. String data

4.1. lower() / upper()

4.2. lstrip() / rstrip()

/ strip()

4.3. replace()

4.4. split()

4.5. type() / sizeof()

4.6. len()

4.7. String processing  
using operators

## 5. List

5.1. append() / insert()

5.2. del() / remove()

5.3. sort() / reverse()

## 6. Ascii code

6.1. ord() / chr()

함수 이름	설명	사용 예
input()	사용자에게 문자형의 데이터를 입력 받습니다 숫자를 입력해도 문자형으로 처리됩니다.	input('입력 전에 출력될 문장')

## 9) input() 함수로 정보 입력받기

```

1 txt1 = input()
2
3 print("txt1 값을 출력하면 :", txt1)

```

짬뽕

txt1 값을 출력하면 : 짬뽕

## 10) input() 함수에서 안내 멘트 보이기

```

1 txt2 = input('점심 뭐 드셨어용? ')

```

점심 뭐 드셨어용? 짬뽕

# type()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. round( )
- 3.2. trunc( )
- 3.3. ceil( )
- 3.4. floor( )

### 4. String data

- 4.1. lower( ) / upper( )
- 4.2. lstrip( ) / rstrip( ) / strip( )
- 4.3. replace( )
- 4.4. split( )
- 4.5. type( ) / sizeof( )
- 4.6. len( )
- 4.7. String processing using operators

### 5. List

- 5.1. append( ) / insert( )
- 5.2. del( ) / remove( )
- 5.3. sort( ) / reverse( )

### 6. Ascii code

- 6.1. ord( ) / chr( )

## 11) 입력 데이터 형식 변환하기

```

1 area_no = input('
2 1. 서울          2.대전          3.광주          4.부산
3 5. 경남          6.경북          7.충청          8.제주
4
5 위 지역중 정보를 조회할 지역의 번호를 입력하세요: ')
6
7 # input으로 입력한 값은 str 형의 데이터
8 print(type(area_no))
9
10 # input으로 입력한 값을 숫자형으로 사용하려면 형식을 변환 해야함
11
12 print(type(int(area_no)))

```

```

1. 서울          2.대전          3.광주          4.부산
5. 경남          6.경북          7.충청          8.제주

```

위 지역중 정보를 조회할 지역의 번호를 입력하세요: 3

```
<class 'str'>
```

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

# Exercise (1)

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

input( ), print( ) 함수를 이용하여 아래와 같은 결과가 나오도록 코딩하시오.

```
첫번째 주문하실 음식은 무엇인가요? 
두번째 주문하실 음식은 무엇인가요? 
첫번째 주문하신 음식은 짜장면이고 두번째 주문하신 음식은 짜장면입니다
```

위에서 **짜장면**, **짜장면**은 input()에 의해 입력한 값입니다.

그 외는 print()로 출력된 문장입니다.

# Numerical Data Processing

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## ● 파이썬에서 산술 연산자를 사용

## 1) 사칙 연산자와 나머지 연산자를 이용한 연산

```
1 #더하기
2 print("1.더하기 예 : 2 + 4 = " , 2 + 4)
3
4 #빼기
5 print("2.빼기 예 : 4 - 2 = " , 4 - 2)
6
7 #곱하기
8 print("3.곱하기 예 : 4 X 2 = " , 4 * 2)
9
10 #나누기
11 print("4.나누기 예 : 5 / 2 = " , 5 / 2)
12
13 #나누기에서 몫만 출력하기
14 print("5.나누기에서 몫만 출력하기 예 : 5 // 2 = " , 5 // 2)
15
16 #나누기에서 나머지값만 출력하기
17 print("6.나누기에서 나머지값만 출력하기 예 : 5 % 2 = " , 5 % 2)
18
19 # 주어진 숫자의 승수값 출력하기
20 print("7.주어진 숫자의 승수값 출력하기 예 : 5 ** 2 = " , 5**2)
```

1.더하기 예 :  $2 + 4 = 6$   
2.빼기 예 :  $4 - 2 = 2$   
3.곱하기 예 :  $4 \times 2 = 8$   
4.나누기 예 :  $5 / 2 = 2.5$   
5.나누기에서 몫만 출력하기 예 :  $5 // 2 = 2$   
6.나누기에서 나머지값만 출력하기 예 :  $5 \% 2 = 1$   
7.주어진 숫자의 승수값 출력하기 예 :  $5 ** 2 = 25$

## 1. Data Science

## 2. Python Basics

## 2.1. Variable

## 2.2. Operator

## 2.3. print()

## 2.4. input()

## 2.5. type()

## 3. Numerical data

## 3.1. int() / float()

## 3.2. round()

## 3.3. trunc()

## 3.4. ceil()

## 3.5. floor()

## 4. String data

## 4.1. lower() / upper()

## 4.2. lstrip() / rstrip()

## / strip()

## 4.3. replace()

## 4.4. split()

## 4.5. type() / sizeof()

## 4.6. len()

## 4.7. String processing using operators

## 5. List

## 5.1. append() / insert()

## 5.2. del() / remove()

## 5.3. sort() / reverse()

## 6. Ascii code

## 6.1. ord() / chr()

## 2) +와 - 연산자를 사용한 연산

```
1  # Case 1
2  i = 3
3  i = i + 2
4  print("i 에 저장된 값은 %s 입니다" %i)
5
6  # Case 2
7  j = 3
8  j += 3
9  print("j 에 저장된 값은 %s 입니다" %j)
10
11 # Case 3
12 j = 10
13 j -= 3
14 print("j 에 저장된 값은 %s 입니다" %j)
```

i 에 저장된 값은 5 입니다

j 에 저장된 값은 6 입니다

j 에 저장된 값은 7 입니다

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## 3) input으로 입력받은 데이터의 산술 연산

```
1 no1 = input("첫번째 숫자 입력: ")
2 no2 = input("두번째 숫자 입력: ")
3
4 print('첫번째 숫자는 %s 이고 두번째 숫자는 %s 이며
5 두 숫자의 합은 %s 입니다' % (no1, no2, no1+no2))
```

첫번째 숫자 입력: 10

두번째 숫자 입력: 20

첫번째 숫자는 10 이고 두번째 숫자는 20 이며

두 숫자의 합은 1020 입니다

## 4) input으로 입력받은 데이터를 숫자형으로 변환한 후 산술 연산

```
1 no1 = int(input("첫번째 숫자 입력: "))
2 no2 = int(input("두번째 숫자 입력: "))
3
4 print('첫번째 숫자는 %s 이고 두번째 숫자는 %s 이며
5 두 숫자의 합은 %s 입니다' % (no1, no2, no1+no2))
```

첫번째 숫자 입력: 10

두번째 숫자 입력: 20

첫번째 숫자는 10 이고 두번째 숫자는 20 이며

두 숫자의 합은 30 입니다

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

- 파이썬에서 숫자 형식 데이터 처리

1) int : 정수형 데이터 형식 / float : 실수형 데이터 형식

```
1 print("정수형 숫자 타입:", int(1) )
2 print("실수형 숫자 타입:", float(1) )
```

정수형 숫자 타입: 1

실수형 숫자 타입: 1.0

# round()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## 2) 숫자 형식에서 많이 사용되는 주요 함수 : round() 반올림 함수

```
1 no1 = 3.45678
2 no2 = 3.56789
3 print("원래값: %s , 반올림후 값: %s" %(no1 , round(no1) ) )
4 print("원래값: %s , 반올림후 값: %s" %(no2 , round(no2) ) )
```

원래값: 3.45678 , 반올림후 값: 3

원래값: 3.56789 , 반올림후 값: 4

## round() 반올림 함수의 소수점 이하 자리수 지정

```
1 no1 = 3
2 no2 = 5
3 print("반올림하기 전 원래값:", no2 / no1)
4 print("소수 첫째자리까지 반올림하기:", round(no2 / no1,1) )
5 print("소수 둘째자리까지 반올림하기:", round(no2 / no1,2) )
```

반올림하기 전 원래값: 1.6666666666666667

소수 첫째자리까지 반올림하기: 1.7

소수 둘째자리까지 반올림하기: 1.67



# trunc()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip()

#### / strip()

#### 4.3. replace()

#### 4.4. split()

#### 4.5. type() / sizeof()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

3) 숫자 형식에서 많이 사용되는 주요 함수 : trunc() 버림 함수

```
1 no1 = 3
2 no2 = 5
3 print("버림하기 전 원래값:", no2 / no1)
4 print("소수 첫째자리까지 버림하기:", math.trunc(no2 / no1) )
5 print("음수 값 :", math.trunc(-3.14))
```

버림하기 전 원래값: 1.6666666666666667

**NameError** Traceback (most recent call last)

<ipython-input-11-c0be94c3b3f2> in <module>

```
2 no2 = 5
3 print("버림하기 전 원래값:", no2 / no1)
----> 4 print("소수 첫째자리까지 버림하기:", math.trunc(no2 / no1) )
5 print("음수 값 :", math.trunc(-3.14))
```

**NameError**: name 'math' is not defined

```
1 import math
2 no1 = 3
3 no2 = 5
4 print("버림하기 전 원래값:", no2 / no1)
5 print("소수 첫째자리까지 버림하기:", math.trunc(no2 / no1) )
6 print("음수 값 :", math.trunc(-3.14))
```

버림하기 전 원래값: 1.6666666666666667

소수 첫째자리까지 버림하기: 1

음수 값 : -3

# ceil( )

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float( )
- 3.2. round( )
- 3.3. trunc( )
- 3.4. ceil( )
- 3.5. floor( )

### 4. String data

- 4.1. lower( ) / upper( )
- 4.2. lstrip( ) /rstrip( )  
/ strip( )
- 4.3. replace( )
- 4.4. split( )
- 4.5. type( ) / sizeof( )
- 4.6. len( )
- 4.7. String processing  
using operators

### 5. List

- 5.1. append( ) / insert( )
- 5.2. del( ) / remove( )
- 5.3. sort( ) / reverse( )

### 6. Ascii code

- 6.1. ord( ) / chr( )

#### 4) 숫자 형식에서 많이 사용되는 주요 함수 : ceil() 천장 함수

```
1 cnt = int( input("총 몇 건의 데이터를 수집할까요? :") )
2
3 # 웹 사이트의 한 페이지에 10건의 게시물이 있다고 가정할 경우
4
5 page_cnt = math.ceil( cnt / 10 )
6 print("총 %s 페이지까지 데이터를 수집해야 합니다" %page_cnt)
```

총 몇 건의 데이터를 수집할까요? :98

총 10 페이지까지 데이터를 수집해야 합니다

# floor()

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## 5) 숫자 형식에서 많이 사용되는 주요 함수 : floor() 바닥 함수

```

1 no1 = 4.9
2 no2 = 4.1
3
4 print("%s 보다 작으면서 가장 가까운 정수는 %s 입니다" %(no1 , math.floor(no1)))
5 print("%s 보다 작으면서 가장 가까운 정수는 %s 입니다" %(no2 , math.floor(no2)))
6 print("음수 값 : ",math.floor(-3.14))

```

4.9 보다 작으면서 가장 가까운 정수는 4 입니다

4.1 보다 작으면서 가장 가까운 정수는 4 입니다

음수 값 : -4

## 6) 숫자 형식에서 많이 사용되는 주요 함수 : trunc()와 floor()의 차이점

```

1 print(math.trunc(-3.14)) # trunc()함수는 0을 향해 버림
2 print(math.floor(-3.14)) # floor() 함수는 무조건 바닥을 향해 내림

```

-3

-4

# String Data Processing

## 문자열

파	이	션	완	전	쉽	쥬	?
0번 요소	1번 요소	2번 요소	3번 요소	4번 요소	5번 요소	6번 요소	7번 요소
변수명[0]	변수명[1]	변수명[2]	변수명[3]	변수명[4]	변수명[5]	변수명[6]	변수명[7]

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

- 파이썬에서 문자 형식 데이터 처리

## 1) 인덱싱을 활용한 문자 데이터 추출

```
1 str1 = '파이썬완전쉽쇼?'  
2 print(str1)  
3 print(str1[0])  
4 print(str1[1])
```

파이썬완전쉽쇼?

파

이

# String Data Processing

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## 2) 슬라이싱을 활용한 문자 데이터 추출

```
1 str1 = '파이썬완전쉽쇼?'  
2 print(str1)  
3  
4 print(str1[0:4])
```

파이썬완전쉽쇼?  
파이썬완

```
1 str1 = '빅데이터수집및분석'  
2  
3 print(str1[5:])  
4 print(str1[:5])
```

집및분석  
빅데이터수

# lower( ) / upper( )

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float( )
- 3.2. round( )
- 3.3. trunc( )
- 3.4. ceil( )
- 3.5. floor( )

### 4. String data

- 4.1. lower( ) / upper( )
- 4.2. lstrip( ) / rstrip( )  
/ strip( )
- 4.3. replace( )
- 4.4. split( )
- 4.5. type( ) / sizeof( )
- 4.6. len( )
- 4.7. String processing  
using operators

### 5. List

- 5.1. append( ) / insert( )
- 5.2. del( ) / remove( )
- 5.3. sort( ) / reverse( )

### 6. Ascii code

- 6.1. ord( ) / chr( )

3) 문자 형식에서 많이 사용되는 주요 함수 : lower( ) 소문자 변환 함수, upper( ) 대문자 변환 함수

```
1 str2 = "PyThoN" # 대소문자가 섞여 있습니다
2 print(str2.lower())
3 print(str2.upper( ))
```

python  
PYTHON

# lstrip() / rstrip() / strip()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()**
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

4) 문자 형식에서 많이 사용되는 주요 함수 : lstrip() 왼쪽 공백 제거 함수 / rstrip() 오른쪽 공백 제거 함수 / strip() 양쪽 공백 제거 함수

```
1 str3 = "    <- 이쪽 끝에 공백 있었어요"
2 str4 = "오른쪽 끝에 공백 있었어요->    "
3 str5 = "    <- 양쪽 끝에 공백 있었어요 ->    "
4
5 print( str3.lstrip( ) )
6 print( str4.rstrip( ) )
7 print( str5.strip( ) )
```

```
<- 이쪽 끝에 공백 있었어요
오른쪽 끝에 공백 있었어요->
<- 양쪽 끝에 공백 있었어요 ->
```



# replace( )

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float( )
- 3.2. round( )
- 3.3. trunc( )
- 3.4. ceil( )
- 3.5. floor( )

### 4. String data

- 4.1. lower( ) / upper( )
- 4.2. lstrip( ) /rstrip( )  
/ strip( )
- 4.3. replace( )
- 4.4. split( )
- 4.5. type( ) / sizeof( )
- 4.6. len( )
- 4.7. String processing  
using operators

### 5. List

- 5.1. append( ) / insert( )
- 5.2. del( ) / remove( )
- 5.3. sort( ) / reverse( )

### 6. Ascii code

- 6.1. ord( ) / chr( )

5) 문자 형식에서 많이 사용되는 주요 함수 : replace( ) 문자 치환 함수

```
1 str6 = '새우깡도 해산물 인가요?'  
2 print(str6.replace("새우깡" , "새우"))
```

새우도 해산물 인가요?

# split( )

## 1. Data Science

## 2. Python Basics

## 2.1. Variable

## 2.2. Operator

## 2.3. print()

## 2.4. input()

## 2.5. type()

## 3. Numerical data

## 3.1. int() / float()

## 3.2. round()

## 3.3. trunc()

## 3.4. ceil()

## 3.5. floor()

## 4. String data

## 4.1. lower() / upper()

## 4.2. lstrip() / rstrip()

## / strip()

## 4.3. replace()

## 4.4. split()

## 4.5. type() / sizeof()

## 4.6. len()

4.7. String processing  
using operators

## 5. List

## 5.1. append() / insert()

## 5.2. del() / remove()

## 5.3. sort() / reverse()

## 6. Ascii code

## 6.1. ord() / chr()

6) 문자 형식에서 많이 사용되는 주요 함수 : split( ) 문자 분리 함수

```
1 tel = '02-1234-5678'
2 print(tel.split('-'))
3 print(tel.split('-',1))
```

['02', '1234', '5678']

['02', '1234-5678']

< 참고 > split()후 데이터 타입 확인

```
1 tel_1 = "02-1234-5647"
2 print(type(tel_1))
3 print(tel_1, "\n")
4
5 tel_2 = tel_1.split('-')
6 print(type(tel_2))
7 print(tel_2, "\n")
8
```

<class 'str'>

02-1234-5647

<class 'list'>

['02', '1234', '5647']

# type() / sizeof()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

### <참고> type / getsizeof 활용법

```
1 import sys
2 a = 2
3 b = 3.5
4 c = "A"
5 print("a:", a, "/ b:", b, "/ c:", c)
6 print(type(a), "/", type(b), "/", type(c))
7 print(sys.getsizeof(a), "byte / ", sys.getsizeof(b), "byte / ", sys.getsizeof(c), "byte")
```

a: 2 / b: 3.5 / c: A  
<class 'int'> / <class 'float'> / <class 'str'>  
28 byte / 24 byte / 50 byte

```
1 aa = "A"
2 bb = 40
3 cc = 23.5
4
5 print("aa:", aa, "/ bb:", bb, "/ cc:", cc)
6 print(type(aa), "/", type(bb), "/", type(cc))
7 print(sys.getsizeof(aa), "byte / ", sys.getsizeof(bb), "byte / ", sys.getsizeof(cc), "byte")
```

aa: A / bb: 40 / cc: 23.5  
<class 'str'> / <class 'int'> / <class 'float'>  
50 byte / 28 byte / 24 byte

# len()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

7) 문자 형식에서 많이 사용되는 주요 함수 : len() 문자열 또는 단어의 길이를 출력하는 함수

```
1 str1 = '파이썬 완전 좋아요'
2 str2 = ['파이썬', '웹크롤러', '가치랩스']
3
4 print(len(str1))
5 print(len(str2))
```

```
10
3
```

# String processing using operators

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

#### 2.1. Variable

#### 2.2. Operator

#### 2.3. print()

#### 2.4. input()

#### 2.5. type()

### 3. Numerical data

#### 3.1. int() / float()

#### 3.2. round()

#### 3.3. trunc()

#### 3.4. ceil()

#### 3.5. floor()

### 4. String data

#### 4.1. lower() / upper()

#### 4.2. lstrip() / rstrip() / strip()

#### 4.3. replace()

#### 4.4. split()

#### 4.5. type() / sizeof()

#### 4.6. len()

#### 4.7. String processing using operators

### 5. List

#### 5.1. append() / insert()

#### 5.2. del() / remove()

#### 5.3. sort() / reverse()

### 6. Ascii code

#### 6.1. ord() / chr()

## 8) 연산자를 이용한 문자열 처리

```
1 print("=" * 80)
2 print("문자열로 연산하기 실습")
3 print("+ " * 80)
```

문자열로 연산하기 실습

+++++

# List

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip() / strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

리스트 타입은 다양한 형식의 데이터를 한꺼번에 저장할 때 사용하면 편리함

예시)

'1982-07-15'	'홍길동'	30
0번 요소	1번 요소	2번 요소
문자형	문자형	숫자형

#### ● 파이썬에서 리스트 활용

#### 1) 리스트 생성

```
1 list1 = ['1982-07-15', '홍길동', 30]
2 print(list1)
```

['1982-07-15', '홍길동', 30]

# append( ) / insert( )

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

## 2) 리스트에서 append( ) 데이터 추가 함수 / insert( ) 데이터 삽입 함수

```
1 list2 = ['첫째', '둘째', '셋째']  
2 list2.append('append로 추가한 것')  
3 print(list2)  
4  
5 list2.insert(2, 'insert로 추가한 것')  
6 print(list2)
```

['첫째', '둘째', '셋째', 'append로 추가한 것']

['첫째', '둘째', 'insert로 추가한 것', '셋째', 'append로 추가한 것']

# del( ) / remove( )

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

## 3) 리스트에서 데이터 삭제 함수 del()과 remove()

< del()과 remove()의 차이점 >

del a[인덱스] : 리스트 a의 해당 인덱스 원소가 삭제

a.remove(원소) : 리스트 a의 해당 원소가 삭제

```
1 a = ['첫째', '둘째', 'three', '셋째', '넷째', 'five', '다섯째']
2 print('a 리스트 ->', a)
3 del a[2]
4 print("")
5
6 print('del a[2]로 삭제후 a 리스트:', a)
7 print("")
8
9 a.remove("five")
10 print('a.remove("five")로 삭제후 목록:', a)
```

a 리스트 -> ['첫째', '둘째', 'three', '셋째', '넷째', 'five', '다섯째']

del a[2]로 삭제후 a 리스트: ['첫째', '둘째', '셋째', '넷째', 'five', '다섯째']

a.remove("five")로 삭제후 목록: ['첫째', '둘째', '셋째', '넷째', '다섯째']



# sort() / reverse()

## Python Basics for Data Processing

### 1. Data Science

### 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

### 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

### 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

### 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

### 6. Ascii code

- 6.1. ord() / chr()

#### 4) 리스트에서 sort() 오름차순 정렬 함수 / reverse() 내림차순 정렬 함수

```
1 list3 = [ 3,1,5,8,2]
2 list3.sort()
3 print('오름차순(기본)정렬후', list3, '\n')
4
5 list3.reverse()
6 print('내림차순 정렬후:', list3, '\n')
7
8 list4 = ['banana', 'apple', 'cherry', 'Apple']
9 list4.sort()
10 print('영어 정렬:', list4, '\n')
11
12 list5 = ['홍길동', '전우치', '김유신', '이순신']
13 list5.sort()
14 print('한글 정렬:', list5)
```

오름차순(기본)정렬후 [1, 2, 3, 5, 8]

내림차순 정렬후: [8, 5, 3, 2, 1]

영어 정렬: ['Apple', 'apple', 'banana', 'cherry']

한글 정렬: ['김유신', '이순신', '전우치', '홍길동']

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

### ■ American Standard Code for Information Interchange(미국정보교환표준부호)

- 축약형 **ASCII(아스키)**는 영문 알파벳을 사용하는 대표적인 문자 encoding
- ASCII는 컴퓨터와 통신 장비를 비롯한 문자를 사용하는 많은 장치에서 사용

# ord() / chr()

## 1. Data Science

## 2. Python Basics

- 2.1. Variable
- 2.2. Operator
- 2.3. print()
- 2.4. input()
- 2.5. type()

## 3. Numerical data

- 3.1. int() / float()
- 3.2. round()
- 3.3. trunc()
- 3.4. ceil()
- 3.5. floor()

## 4. String data

- 4.1. lower() / upper()
- 4.2. lstrip() / rstrip()  
/ strip()
- 4.3. replace()
- 4.4. split()
- 4.5. type() / sizeof()
- 4.6. len()
- 4.7. String processing  
using operators

## 5. List

- 5.1. append() / insert()
- 5.2. del() / remove()
- 5.3. sort() / reverse()

## 6. Ascii code

- 6.1. ord() / chr()

<참고> ASCII 코드 값 활용

ord() 함수 : 문자를 아스키 코드 값으로 출력 / chr() 함수 : 아스키 코드 값을 문자로 출력

```
1 list_A = "A"
2 list_B = "B"
3
4 list_a = "a"
5 list_b = "b"
6
7 print("A 아스키 코드 값 : ", ord(list_A))
8 print("B 아스키 코드 값 : ", ord(list_B))
9 print("a 아스키 코드 값 : ", ord(list_a))
10 print("b 아스키 코드 값 : ", ord(list_b))
```

```
A 아스키 코드 값 : 65
B 아스키 코드 값 : 66
a 아스키 코드 값 : 97
b 아스키 코드 값 : 98
```

```
1 print("아스키 코드 값 65 : ", chr(65))
2 print("아스키 코드 값 66 : ", chr(66))
3 print("아스키 코드 값 97 : ", chr(97))
4 print("아스키 코드 값 98 : ", chr(98))
```

```
아스키 코드 값 65 : A
아스키 코드 값 66 : B
아스키 코드 값 97 : a
아스키 코드 값 98 : b
```