

# Python Basic Syntax for Data Processing

강의 : 김 희 숙  
Christina H.S.Kim

- Data Science

- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## ■ Data Science ?

- Data Mining과 유사하게 정형, 비정형 형태를 포함한 다양한 데이터로부터 knowledge와 insight를 추출하는 convergence 분야
- 튜링상(Turing Award)을 수상한 짐그레이(Jim Gray) 박사는 과학의 네번째 Paradigm으로 **데이터 과학(Data Science)**을 정의
  - ✓ 1st Paradigm : 수천 년 전 자연 현상을 관찰하는 **실험** 중심의 과학
  - ✓ 2nd Paradigm : 수백 년 전 **이론** 중심의 과학
  - ✓ 3rd Paradigm : 수십 년 전 **계산** 중심의 과학
  - ✓ 4th Paradigm : **오늘날 데이터 중심의 과학(Data-Intensive Science)**



정보기술(Information Technology)과 데이터 범람(data deluge) 때문

## • Data Science

## • Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과  
주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 /  
upper( ) 함수
  - 2) lstrip( ) 함수 /  
rstrip( ) 함수 /  
strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 /  
insert( ) 함수
  - 2) del( ) 함수 /  
remove( ) 함수
  - 3) sort( ) 함수 /  
reverse( ) 함수

## ■ Data 표현 단위

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

## • Data Science

## • Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## ■ Data의 증가

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

- Data Science

- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## ■ Gartner사의 Big Data에 대한 정의

- Big Data란 높은 통찰력, 의사결정, 프로세스 자동화를 위해 효과가 높은 혁신적인 정보처리 과정을 요구한다.
- 다음과 같은 3가지 특징을 지닌 정보 자산이다.
  - ① 대용량의 데이터 규모(high-volume)
  - ② 빠른 속도(high-velocity)
  - ③ 높은 다양성(high-variety)

- Data Science

- Python basics for data processing

1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## ■ Big Data의 활용 분야

- 인터넷 검색(Internet Search)

- ✓ 인터넷 검색은 데이터 과학 알고리즘을 사용하여 몇 초 안에 최상의 검색 결과를 제공

- 디지털 광고(Digital Advertisement)

- ✓ 디지털 광고는 디스플레이 배너부터 디지털 광고판까지의 모든 디지털 마케팅 스펙트럼을 데이터 과학 알고리즘을 사용

- 추천 시스템(Recommender systems)

- ✓ 쇼핑몰에서 추천 시스템은 데이터 과학 알고리즘을 사용하여 수십억 개의 제품에서 관련 상품을 쉽게 찾게 해줄 뿐만 아니라 소비자 경험을 더욱 풍부하게 해줌
- ✓ 많은 기업이 소비자의 니즈와 정보를 기반으로 제품과 서비스를 홍보하기 위해서 사용
- ✓ 이러한 추천 메커니즘은 소비자의 이전 검색 기록 데이터를 활용함

- Data Science

- Python basics for data processing

1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## ■ 데이터 과학자가 되기 위한 필요한 요소

- 프로그래밍 언어 활용 능력이 필요함

- ✓ Java, C언어, Perl과 함께 데이터 과학자들에게 가장 대중적으로 쓰이는 Python 코딩 능력
- ✓ SAS, R에 대한 깊이 있는 지식(데이터 과학자들은 R을 선호하는 추세)

- SQL 데이터베이스 관리

- ✓ 데이터 과학자들에게 SQL의 정교하고 복잡한 쿼리를 능숙하게 사용할 수 있는 능력이 여전히 선호되고 있음

- 비정형 데이터 처리

- ✓ 데이터 과학자들에게 비정형 데이터(웹 상의 텍스트, 이미지, 영상, 음성 등)를 다룰 수 있는 능력이 매우 중요해지고 있음

- ✓ 데이터 분석 후 시각화 기술

- 통계적 기술 및 수학 분야의 이해 능력과 원활한 의사 소통 방법이 필요

- 머신러닝 및 딥러닝 기술이 필요

# Python basics for data processing

• Data Science

• Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

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



# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

### 1.1 다양한 연산자 살펴보기

- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

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	$a \% b$

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## 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

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

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!

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 / strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

형식	의미
%d	정수(1,2,3,4,5 등)
%f	실수(1.234 등)
%s	문자열('this is string') 숫자도 포함
%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!

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round()
    - 2) trunc()
    - 3) ceil()
    - 4) floor()

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower() 함수 / upper() 함수
    - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
    - 3) replace() 함수
    - 4) split() 함수
    - 5) len() 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append() 함수 / insert() 함수
    - 2) del() 함수 / remove() 함수
    - 3) sort() 함수 / reverse() 함수

## 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....

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## 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..

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 **사용자에게 값을 입력 받아 변수에 저장하기**

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

함수 이름	설명	사용 예
input( )	사용자에게 값을 입력 받습니다	input( )

## 9) input( ) 함수로 정보 입력받기

```

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

▶▶▶

txt1 값을 출력하면 : ▶▶▶

## 10) input() 함수에서 안내 멘트 보이기

```

1 txt2 = input('점심 뭐 드셨어용? ')
```

점심 뭐 드셨어용? ▶▶▶

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 **사용자에게 값을 입력 받아 변수에 저장하기**

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round()
    - 2) trunc()
    - 3) ceil()
    - 4) floor()

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower() 함수 / upper() 함수
    - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
    - 3) replace() 함수
    - 4) split() 함수
    - 5) len() 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append() 함수 / insert() 함수
    - 2) del() 함수 / remove() 함수
    - 3) sort() 함수 / reverse() 함수

## 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

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 **사용자에게 값을 입력 받아 변수에 저장하기**
2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )
3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용
4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

아래와 같은 결과가 나오도록 코딩하시오.

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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

### 2.1 정수형 타입과

#### 주요 연산자 사용 방법

- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## • 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 몫 만 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 :  $5 ** 2 = 25$

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

### 2.1 정수형 타입과

#### 주요 연산자 사용 방법

- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 / rstrip( ) 함수 / strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기**
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입**
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

## ● 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들

### 1) round()

### 2) trunc()

### 3) ceil()

### 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### 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



# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## 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 내림을 하더라도 0을 향해 내림
2 print(math.floor(-3.14)) # floor() 함수는 무조건 바닥을 향해 내림

```

-3

-4

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 / strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## 문자열

파	이	썬	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]

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기
2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )
3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱**
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용
4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## ● 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

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기
2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )
3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱**
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용
4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round( )
  - 2) trunc( )
  - 3) ceil( )
  - 4) floor( )

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower( ) 함수 / upper( ) 함수
  - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
  - 3) replace( ) 함수
  - 4) split( ) 함수
  - 5) len( ) 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append( ) 함수 / insert( ) 함수
  - 2) del( ) 함수 / remove( ) 함수
  - 3) sort( ) 함수 / reverse( ) 함수

### 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

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

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 ->
```

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기
2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )
3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용
4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## 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?

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 /rstrip() 함수 /strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

## 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']
```



# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### <참고> type / sizeof 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

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기
2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )
3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용
4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## 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

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 /strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

## 8) 연산자를 이용한 문자열 처리

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

=====

문자열로 연산하기 실습

+++++

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round( )
    - 2) trunc( )
    - 3) ceil( )
    - 4) floor( )

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower( ) 함수 / upper( ) 함수
    - 2) lstrip( ) 함수 /rstrip( ) 함수 / strip( ) 함수
    - 3) replace( ) 함수
    - 4) split( ) 함수
    - 5) len( ) 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append( ) 함수 / insert( ) 함수
    - 2) del( ) 함수 / remove( ) 함수
    - 3) sort( ) 함수 / reverse( ) 함수

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]

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

### 4.1 리스트

### 4.2 리스트 관련 주요 함수들

- 1) append() 함수 / insert() 함수
- 2) del() 함수 / remove() 함수
- 3) sort() 함수 / reverse() 함수

## 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']

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### 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']

# Python basics for data processing

- Data Science
- Python basics for data processing

1. 변수
  - 1.1 다양한 연산자 살펴보기
  - 1.2 변수 값을 화면에 출력하기
  - 1.3 사용자에게 값을 입력 받아 변수에 저장하기

2. 숫자형 데이터
  - 2.1 정수형 타입과 주요 연산자 사용 방법
  - 2.2 정수형으로 변환하기
  - 2.3 실수형 숫자 타입
  - 2.4 숫자 관련 주요 함수들
    - 1) round()
    - 2) trunc()
    - 3) ceil()
    - 4) floor()

3. 문자열 데이터
  - 3.1 문자열
  - 3.2 인덱싱과 슬라이싱
  - 3.3 문자열 관련 주요 함수들
    - 1) lower() 함수 / upper() 함수
    - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
    - 3) replace() 함수
    - 4) split() 함수
    - 5) len() 함수
  - 3.4 연산자를 이용한 문자열 활용

4. 리스트(list)
  - 4.1 리스트
  - 4.2 리스트 관련 주요 함수들
    - 1) append() 함수 / insert() 함수
    - 2) del() 함수 / remove() 함수
    - 3) sort() 함수 / reverse() 함수

## 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']

# Python basics for data processing

- Data Science
- Python basics for data processing

## 1. 변수

- 1.1 다양한 연산자 살펴보기
- 1.2 변수 값을 화면에 출력하기
- 1.3 사용자에게 값을 입력 받아 변수에 저장하기

## 2. 숫자형 데이터

- 2.1 정수형 타입과 주요 연산자 사용 방법
- 2.2 정수형으로 변환하기
- 2.3 실수형 숫자 타입
- 2.4 숫자 관련 주요 함수들
  - 1) round()
  - 2) trunc()
  - 3) ceil()
  - 4) floor()

## 3. 문자열 데이터

- 3.1 문자열
- 3.2 인덱싱과 슬라이싱
- 3.3 문자열 관련 주요 함수들
  - 1) lower() 함수 / upper() 함수
  - 2) lstrip() 함수 / rstrip() 함수 / strip() 함수
  - 3) replace() 함수
  - 4) split() 함수
  - 5) len() 함수
- 3.4 연산자를 이용한 문자열 활용

## 4. 리스트(list)

- 4.1 리스트
- 4.2 리스트 관련 주요 함수들
  - 1) append() 함수 / insert() 함수
  - 2) del() 함수 / remove() 함수
  - 3) sort() 함수 / reverse() 함수

### <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

```