



▶ 학습내용

- 01 정수, 실수, 문자열을 구별할 수 있습니다.
- 02 문자열을 숫자로, 숫자를 문자열로 변환할 수 있습니다.
- 03 문자열을 연결하고 반복하는 연산을 살펴봅니다.
- 04 필요한 문자열만 뽑아 쓰는 방법에 대해 살펴봅니다.

▶ LAB

- 01 소금물의 농도는?
- 02 간단한 챗봇 프로그램
- 03 거북이와 인사해봐요
- 04 암호프로그램 만들기
- 05 2050년에 나는 몇 살?

01. 파이썬에서 사용할 수 있는 자료의 종류

- 모든 프로그램은 자료(data)를 처리.
- 프로그래밍에서 사용되는 자료는 종류에 따라 자료형(data type)으로 구분 지어 놓음
- 프로그래밍에서 가장 기본적인 자료형: 정수, 실수, 문자열

자료형	예
정수(int)	..., -2, -1, 0, 1, 2, ...
실수(float)	3.2, 3.14, 0.12345
문자열(str)	'Hello World!', "123"

01. 파이썬에서 사용할 수 있는 자료의 종류



파이썬에서는 변수에 어떤 종류의 자료도 저장할 수 있습니다.

코드

```
x = 10
print("x =", x)

x = 3.14
print("x =", x)

x = "Hello World!"
print("x =", x)
```

실행 결과

```
x = 10
x = 3.14
x = Hello World!
```

02. 정수형과 실수형



- 수치형 데이터는 산술연산을 할 수 있음.

수치형 데이터	예
정수	$\dots, -2, -1, 0, 1, 2, \dots$
실수	3.2, 3.14, 0.12345
복소수	$1 + 2j, -3j$

02. 정수형과 실수형



정수를 어떻게 만들고 사용할까요?

코드 및 실행 결과

```
>>> x = 123
>>> print(x)
123
>>> type(x)
<class 'int'>
```

코드 및 실행 결과

```
>>> x = -123
>>> print(x)
-123
>>> type(x)
<class 'int'>
```

코드 및 실행 결과

```
>>> x = 0
>>> print(x)
0
>>> type(x)
<class 'int'>
```

02. 정수형과 실수형



실수를 어떻게 만들고 사용할까요?

코드 및 실행 결과

```
>>> x = 3.14
>>> print(x)
3.14
>>> type(x)
<class 'float'>
```

코드 및 실행 결과

```
>>> x = -3.14
>>> print(x)
-3.14
>>> type(x)
<class 'float'>
```

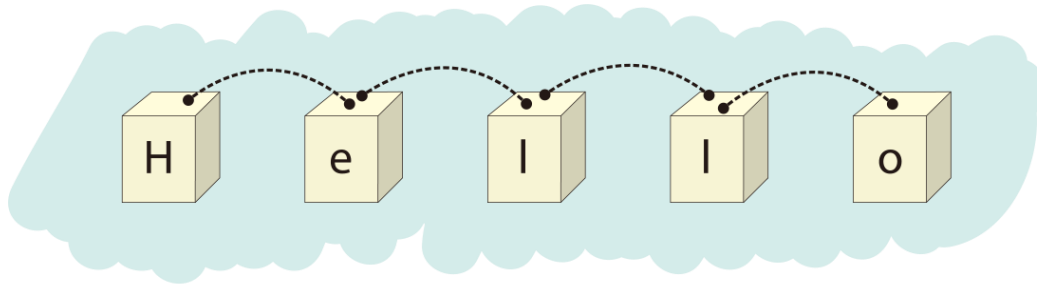
코드 및 실행 결과

```
>>> x=0.0
>>> print(x)
0.0
>>> type(x)
<class 'float'>
```

03. 문자열이란?



- 문자열(string)은 문자들의 나열(sequence of characters)입니다.



문자열은 문자들의 나열입니다.

"Hello World" = 'Hello World'

같은 표현

04. 문자열을 만드는 방법



- 작은따옴표('')와 큰따옴표("")를 사용

코드 및 실행 결과

```
>>> "Hello"  
'Hello'
```

코드 및 실행 결과

```
>>> msg = "Hello"  
>>> msg  
'Hello'  
>>> print(msg)  
Hello
```


04. 문자열을 만드는 방법



- 큰 따옴표로 열었으면 큰 따옴표로 닫아 줌 : ""
- 작은 따옴표로 열었으면 작은 따옴표로 닫아 줌: ''

코드 및 실행 결과

```
>>> msg = "Hello"
```

```
SyntaxError: EOL while scanning string literal
```

04. 문자열을 만드는 방법



그런데 말입니다. 왜 파이썬에서는 문자열을 나타내는데 큰따옴표와 작은따옴표 둘 다 사용할까요?

이유 : 문자열 안에 따옴표가 들어가는 경우를 처리하기 위해서입니다.

코드 및 실행 결과

```
>>> msg = 'She said 'Hi''  
SyntaxError: invalid syntax
```

코드 및 실행 결과

```
>>> msg = "She said 'Hi' "  
>>> print(msg)  
She said 'Hi'
```

05. 100과 “100” 을 구별해요



100

정수 수치형 데이터

"100" 또는 '100'

문자열



문자열 "100"과 숫자 100은 컴퓨터에서는 아주 다르게 취급됩니다.

코드 및 실행 결과

```
>>> print(100+200)
300
```

VS

코드 및 실행 결과

```
>>> print("100"+"200")
100200
```

05. 100과 “100” 을 구별해요



사용자로부터 2개의 정수를 입력받아 덧셈을 수행하는 프로그램을 작성해 봅시다.

코드

```
x = input("정수를 입력하시오: ")  
y = input("정수를 입력하시오: ")  
print(x+y)
```

실행 결과

```
정수를 입력하시오: 100  
정수를 입력하시오: 200  
100200
```

06. 문자열을 숫자로 변환



- input()은 사용자가 입력한 데이터를 문자열 형태로 돌려줍니다.



문자열을 숫자로 바꿀 수 없나요?

코드

```
t = input("정수를 입력하시오: ")
x = int(t)

t = input("정수를 입력하시오: ")
y = int(t)

print(x + y)
```

실행 결과

```
정수를 입력하시오: 100
정수를 입력하시오: 200
300
```

06. 문자열을 숫자로 변환



위 코드를 줄일 방법은 없나요?

```
t = input("정수를 입력하시오: ")  
x = int(t)
```

→ **합체** → `x = int(input("정수를 입력하시오: "))`

06. 문자열을 숫자로 변환



사용자가 입력한 문자열을 실수로 변환하여 계산하는 프로그램을 작성해 봅시다.

코드

```
x = float(input("실수를 입력하시오: "))  
y = float(input("실수를 입력하시오: "))  
  
print(x + y)
```

실행 결과

```
실수를 입력하시오: 3.14  
실수를 입력하시오: 9.87  
13.01
```

07. 숫자를 문자열로 변환



- 오류 발생

코드 및 실행 결과

```
>>> print('나는 현재 ' + 17 + '살이다.')
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    print('나는 현재 ' + 17 + '살이다.')
TypeError: can only concatenate str (not "int") to str
```

문자열과 문자열만 연결할 수 있다는 메시지입니다.

07. 숫자를 문자열로 변환



정수를 문자열로 바꿀 수 없나요?

코드

```
>>> print('나는 현재 ' + str(17) + '살이다.')
```

실행 결과

나는 현재 17살이다.



실수를 문자열로 바꿀 수 없나요?

코드

```
>>> print('원주율은 ' + str(3.14) + '입니다.')
```

실행 결과

원주율은 3.14입니다.

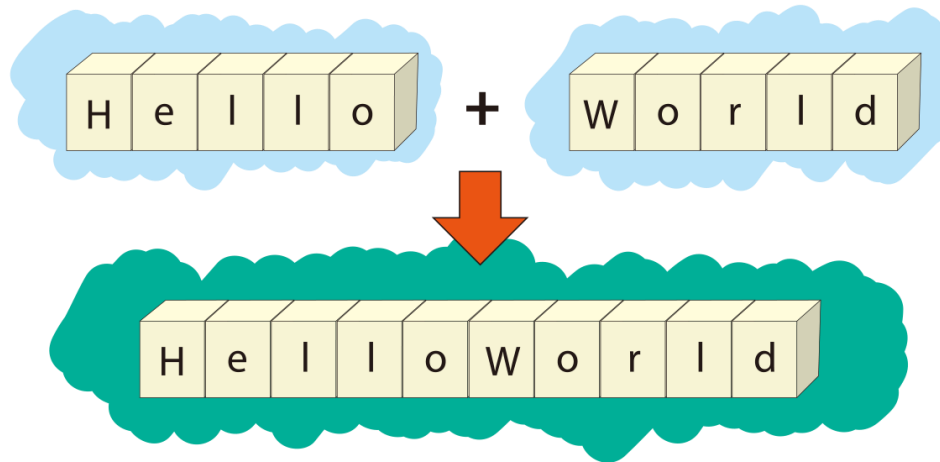
08. 파이썬은 문자열 처리의 마술사



'+' 연산자로 2개의 문자열을 하나의 문자열로 합칠 수 있습니다.

코드 및 실행 결과

```
>>> 'Hello' + 'World'  
'HelloWorld'
```



08. 파이썬은 문자열 처리의 마술사



변수에 저장된 문자열도 '+' 연산자로 합칠 수 있습니다.

코드 및 실행 결과

```
>>> first_name = "길동"
>>> last_name = "홍"
>>> name = last_name + first_name
>>> print(name)
홍길동
```

08. 파이썬은 문자열 처리의 마술사



'*' 연산자를 이용하셔서 반복시킬 수 있습니다.

코드 및 실행 결과

```
>>> message = "Congratulations!"
>>> print(message*3)
Congratulations!Congratulations!Congratulations!
```

코드 및 실행 결과

```
>>> line = "=" * 30
>>> print(line)
=====
```

08. 파이썬은 문자열 처리의 마술사



'%s'를 이용하여 문자열 사이에 변수의 값을 삽입하여 출력할 수도 있습니다.

코드 및 실행 결과

```
>>> price = 10000
>>> print("상품의 가격은 %s원입니다." % price)
상품의 가격은 10000원입니다.
```

08. 파이썬은 문자열 처리의 마술사



만약 여러 줄을 한꺼번에 입력하려 할 때는 줄을 바꾸는 개행문자("\n")를 이용합니다.

코드

```
poem = "이렇게 정다운\n너 하나 나 하나는\n어디서 무엇이 되어\n다시 만나랴."  
print(poem)
```

실행 결과

```
이렇게 정다운  
너 하나 나 하나는  
어디서 무엇이 되어  
다시 만나랴.
```

08. 파이썬은 문자열 처리의 마술사



줄 바꿈이 여러 번 있을 때 매번 개행문자('\n') 쓰는 것보다 더 편한 방법은 없나요?

코드

```
poem = """이렇게 정다운  
너 하나 나 하나는  
어디서 무엇이 되어  
다시 만나랴."""  
print(poem)
```

실행 결과

```
이렇게 정다운  
너 하나 나 하나는  
어디서 무엇이 되어  
다시 만나랴.
```



글자체에 따라 '\n'과 '\r\n'으로 입력될 수 있습니다. 동일한 코드이므로 당황하지 마세요.

09. 필요한 문자열을 뽑아서 쓰자



- 문자들은 서로를 구분하기 위해 인덱스(index)가 매겨져 있음.

S = "Hello Python"

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		P	y	t	h	o	n
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

09. 필요한 문자열을 뽑아서 쓰자



문자열에서 문자를 하나씩 추출하는 프로그램을 작성해 봅시다.

문자열이 저장된 변수 `s`에서 인덱스 `a`번인 문자열을 추출하려면 `s[a]`로 나타내면 됩니다.

코드

```
s = "Hello Python"
print(s[0])
print(s[1])
print(s[-1])
```

실행 결과

```
H
e
n
```

09. 필요한 문자열을 뽑아서 쓰자



- 문자열이 저장된 변수 s 에서 $s[a:b]$ 는 인덱스 a 부터 $b-1$ 까지의 문자열 ($a < b$)을 말합니다.
- $s[a:b:c]$ 는
 - $a < b$ 이고 $c > 0$ 이면 a 부터 $b-1$ 까지의 c 간격의 문자열
 - $a > b$ 이고 $c < 0$ 이면 a 부터 $b+1$ 까지의 c 간격의 문자열을 말합니다.
- $s[0: :1]$ 은 $s[0]$ 에서부터 문자열 끝까지 1간격의 문자열, 즉 s 문자열 전체를 의미합니다.
- $s[-1: :-1]$ 은 $s[-1]$ 에서부터 문자열 처음까지 -1간격의 문자열, 즉 s 문자열 거꾸로 전체를 의미합니다.
- $s[:]$ 는 s 문자열 전체를 의미합니다.

09. 필요한 문자열을 뽑아서 쓰자



문자열에서 연속된 문자열 추출하는 프로그램을 작성해 봅시다.

코드

```
s = "Hello Python"
print(s[6:10])
print(s[-6:-2])
print(s[0:10:2])
print(s[-1:-7:-1])
```

실행 결과

```
Pyth
Pyth
HloPt
nohtyP
```

S = "Hello Python"

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		P	y	t	h	o	n
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Lab. 소금물의 농도는?



과학 시간과 수학 시간에 소금물의 농도를 구하는 문제를 많이 보셨죠? 용액 속에 녹아 있는 용질의 질량을 백분율(%)로 나타낸 것을 농도라고 합니다. 농도를 구하는 식은 다음과 같습니다.

$$\text{소금물의 농도} = \left(\frac{\text{소금의 양}}{\text{소금물의 양}} \right) \times 100(\%)$$

그럼, 소금의 양과 물의 양을 입력받아 소금물의 농도를 구하는 프로그램을 작성해 보겠습니다.

실행 결과

소금물의 농도를 구하는 프로그램입니다

소금의 양은 몇 g입니까? 50

물의 양은 몇 g입니까? 80

소금물의 농도: 38.46153846153847%

Lab. 간단한 챗봇(ChatBot) 프로그램



챗봇은 음성이나 문자를 통한 인간과의 대화를 통해서 특정한 작업을 수행하도록 제작된 컴퓨터 프로그램입니다. 우리에게 이름과 나이를 물어보면서 다음과 같이 친근하게 대화를 하는 챗봇 프로그램을 작성해 보겠습니다.

실행 결과

안녕하세요

이름이 뭐예요? **홍길동**

만나서 반갑습니다. 홍길동 님

홍길동 님, 이름의 길이는 다음과 같군요: 3

나이가 어떻게 돼요? **20**

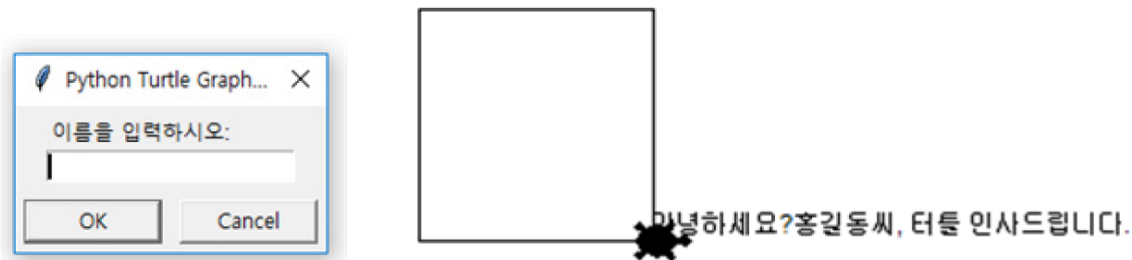
내년이면 21세가 되시는군요



Lab. 거북이와 인사해봐요



입력창을 아래와 같이 띄어 사용자로부터 이름을 입력받고, 다음과 같이 거북이가 인사 메시지를 보내고 기념으로 사각형을 그려주는 프로그램을 작성해 봅시다.



생각 1 : 프로그램의 순서를 생각해 봅시다.

터틀 그래픽을 사용할 준비 작업



입력창 띄워 사용자 이름 입력받기



인사 메시지 출력



사각형 그리기

Lab. 암호프로그램 만들기



암호는 비밀을 유지하기 위하여 당사자끼리만 알 수 있도록 꾸민 약속 기호입니다. 암호에서 사용되는 용어 중 ‘평문’은 일반인 누구나 읽고 이해할 수 있는 문장을 말하고, ‘암호문’은 원래의 평문을 암호키 및 암호 알고리즘을 사용하여 암호 처리된 문장을 말합니다.

암호를 만드는 방법은 여러 가지가 있습니다만 우선 우리는 평문을 거꾸로 바꾸어 암호문을 만들어 보겠습니다. ‘도서관에서 보자’라는 문장을 거꾸로 배열하여 암호로 만들어 주는 프로그램을 만들어 보세요.



실행 결과

평문: 도서관에서 보자

암호문: 자보 서예관서도

Lab. 2050년에 나는 몇살?



- 아래 코드를 적절하게 변경하여 2050년의 우리의 나이를 계산해주는 프로그램을 작성해 보겠습니다

코드

```
import time
now = time.time()
thisYear = int(1970 + now//((365 * 24 * 3600)))
print("올해는 " + str(thisYear) + "년입니다.")
```

실행 결과

올해는 2020년입니다.
당신의 나이를 입력하세요: 17
2050년에는 47살이군요.

2050