



2주차

개발환경구축과 파이썬 기초 문법

1. 개발환경구축

(1) 개발환경이란?

개발환경은 지난 시간에 살펴본 개발과정을 진행하는 데에 필요한 모든 하드웨어와 소프트웨어를 갖추고 나의 의도에 맞게 설정해 둔 상태를 의미합니다. 예전에는 컴퓨터, 네트워크, 프로그래밍 언어, 에디터, 라이브러리, 빌더, 디버거, 컴파일러 등을 각각 따로 준비해야 했습니다. 그러나 통합개발환경(IDE)이라는 개념이 등장하고 난 뒤부터는 하나의 프로그램으로 개발과정 전체를 구축하고 관리할 수 있게 되었습니다. IDE는 종류가 매우 다양하지만 언어나 용도별로 가장 많이 사용하는 프로그램이 거의 정해져 있습니다. 파이썬을 개발환경을 구축할 때는 JetBrains의 PyCharm과 Microsoft의 Visual Studio Code를 많이 사용합니다.

(2) Python 설치하기 (python.org)



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 2.0.0a4	June 20, 2000	Download	Release Notes
Python 3.7.11	June 28, 2021	Download	Release Notes
Python 3.9.5	May 3, 2021	Download	Release Notes
Python 3.8.10	May 3, 2021	Download	Release Notes
Python 3.9.4	April 4, 2021	Download	Release Notes
Python 3.8.9	April 2, 2021	Download	Release Notes
Python 3.9.2	Feb. 19, 2021	Download	Release Notes
Python 3.8.8	Feb. 19, 2021	Download	Release Notes

[View older releases](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		83d71c304acab6c578e86e239b42fa7e	24720640	SIG
XZ compressed source tarball	Source release		d9eee4b20155553830a2025e4dcaa7b3	18433456	SIG
macOS 64-bit intel installer	macOS	for macOS 10.9 and later	690ddb1be403a7efb202e93f3a994a49	29896827	SIG
macOS 64-bit universal2 installer	macOS	experimental, for macOS 11 Big Sur and later, recommended on Apple Silicon	ae8a1ae082074b260381c058d0336d05	37300939	SIG
Windows embeddable package (32-bit)	Windows		659adf421e90fba0f56a9631f79e70fb	7348969	SIG
Windows embeddable package (64-bit)	Windows		3acb1d7d9bde5a79fb40167b166bb633	8211403	SIG
Windows help file	Windows		a06af1ff933a13f6901a75e59247cf95	6597086	SIG
Windows installer (32-bit)	Windows		b355cfc84b681ace8908ae50908e8761	27204536	SIG
Windows installer (64-bit)	Windows	Recommended	62cf1a12a5276b0259e8761d4cf4fe42	28296784	SIG



(3) VS Code 설치하기

<https://code.visualstudio.com/>

II. 파이썬 기초 문법

(1) 변수 선언과 할당

어떤 데이터를 여러분이 개발할 프로그램에서 지속적으로 사용할 수 있도록 하려면 데이터를 그냥 입력하는 것만으로는 충분하지 않습니다. 단순히 입력'만' 된 데이터는 메모리 공간에 잠시 로드되었다가 곧바로 사라지기 때문입니다. 그러한 일을 방지하고 데이터를 저장하기 위해서 우리는 '변수'라는 도구를 사용할 것입니다. 변수는 데이터를 담는 그릇입니다. 변수를 사용하기 위해서는 먼저 변수이름을 지어 '선언'하고, '할당연산자'를 사용해서 원하는 데이터를 저장해주면 됩니다. 이때 변수이름을 짓는 몇 가지 규칙을 잘 기억해두는 것이 좋습니다. 한 번 해볼까요?

- 변수 이름짓기 규칙 (naming convention)

- a) 영문 대소문자로 만든다. (한글도 사용은 가능하나 권장하지 않음)
- b) 숫자를 사용할 수 있으나 첫 글자에는 숫자를 사용할 수 없다.
- c) 언더 바 _ 외에 다른 특수문자나 공백은 사용할 수 없다.
- d) 프로그래밍 언어에서 미리 예약된 키워드는 사용할 수 없다.

- snake_case : 변수명, 함수명을 표기할 때 사용하는 방식

- camelCase : 클래스명을 표기할 때 사용하는 방식

▲ 해봅시다

[예제1] 오늘 날짜를 입력하고 실행해 봅시다. ex) 20211111

[예제2] 변수 'today'를 선언하고 오늘 날짜를 저장해봅시다. ex) today = 20211111

[예제3] 변수 'atmo_temp'를 선언하고 현재기온을 저장해봅시다. ex) atmo_temp = 14.5

Q. 왜 아무 일도 일어나지 않는 걸까?

(2) 데이터와 데이터 타입

당연한 이야기지만 데이터에는 한 가지 종류만 있는 것이 아니라 셀 수 없이 다양한 형태가 존재합니다. 자연수와 글자에서부터 소수, 문장, 리스트, 표, 이미지, 소리, 동영상까지 우리가

기록하고 싶은 모든 정보는 다 각자의 ‘데이터타입’을 지닌 데이터가 될 수 있습니다. 각 데이터타입의 특성을 명확하게 알고 있어야 내가 원하는 작업을 구현할 수 있습니다. 또 같은 데이터타입을 가진 데이터끼리는 다양한 연산이나 알고리즘을 적용할 수도 있기 때문에 데이터와 데이터 타입에 대해 알아야 합니다. 여기에서는 파이썬에서 사용할 수 있는 기본 데이터 타입에 대해 알아보겠습니다.

a) 기본 자료형

▶ 숫자

자료형	설명	예시
정수	10진법을 사용하는 소수를 제외한 정수	123, -456, 0
실수	부동소수점 표기가 가능한 소수를 포함하는 숫자	1.23, -4.56, 1.23E2
16진수	16진법을 사용해 나타내는 숫자	0xA, 0xABC

▶ 문자열

하나 이상의 문자로 이루어진 데이터를 ‘문자열 (string)’이라고 부릅니다. 문자열을 저장하고 싶을 때는 데이터의 양쪽에 따옴표 “ ”를 추가하고 변수에 할당해주면 됩니다. 파이썬에서는 문자열을 쉽고 빠르게 가공해 사용할 수 있도록 문자열에 대해서도 숫자형 자료와 마찬가지로 다양한 연산을 기본 지원합니다. 대표적인 함수는 split, strip, find 등이 있고 그 외에도 아주 다양한 기능들이 내장되어있습니다. 지금은 문자열을 저장하는 것까지만 직접 해보고, 자료형과 연산자들을 다 살펴본 뒤에 문자열을 다뤄보도록 하겠습니다.

▲ 해봅시다

[예제1] 다음 문자열을 저장해봅시다: Hello World!

[예제2] 다음 문자열을 저장해봅시다: ‘나는 너다.’

[예제3] 다음 문자열을 저장해봅시다:

“훌륭하게 사는 것과
아름답게 사는 것과
정의롭게 사는 것은 같다“는 말일세.

[예제4] 저장한 문자열을 출력해봅시다.

Q. 여러 줄 문자열 저장하기? / 내 코드에 주석을 다는 방법

ex1)

“강아지 안녕!”

ex2)

‘‘떡볶이 먹고 싶다..

냉면도 먹고싶다..

빙수도 먹고싶다..’’

ex3)

“”강아지 안녕!

고양이도 안녕!

참새도 안녕!“”

※ 이스케이프 코드 (escape code) 란?

: 프로그래밍 할 때 사용하기 편리하도록 미리 정의된 특수문자입력 코드입니다. 주로 문장부호나 서식에 관련된 기능들을 담고 있습니다.

- 자주 사용되는 파이썬 이스케이프 코드들

\n : 줄바꿈

\t : tab키를 한 번 눌렀을 때 만큼의 공백을 준다

\\ : \ (백슬래쉬) 자체를 쓰고 싶을 때

\' : ' 자체를 쓰고 싶을 때

\“ : ” 자체를 쓰고 싶을 때

- 이스케이프 코드를 사용하여 출력하기

: 이스케이프 코드를 활용하면 ‘ ’와 “ ”만으로도 여러 줄 문자열을 표현할 수 있습니다.

ex1)

```
'떡볶이 먹고 싶다..\n냉면도 먹고싶다..\n빙수도 먹고싶다..'
```

ex2)

```
"강아지 안녕!\n고양이도 안녕!\n참새도 안녕!"
```

※ 주의할 점! :

- **숫자를 따옴표로 묶으면 무조건 문자열이 된다!**

type(2)와 type('2')는 다른 결과가 출력됩니다.

▶ 논리형 (Boolean)

어떤 문장이나 조건, 또는 값의 논리적 참 또는 거짓을 나타내는 데이터 타입 입니다. 참과 거짓을 True / False로 표기합니다. (첫글자는 반드시 대문자로!) 우리가 나중에 배울 조건문에서 매우 중요한 도구로 사용됩니다. Bool 대수의 창시자인 영국의 논리학자 조지 불의 이름을 따서 Boolean 이라고 부릅니다.

▶ None

‘비어 있음’ 또는 ‘없음’을 나타내는 데이터 타입 입니다. ‘0’을 의미하는 것으로 착각할 수도 있지만 0은 정수에 해당하기 때문에 숫자형 데이터이고, None은 말 그대로 아무것도 없는 상태 그 자체를 표시해줍니다. 어디에 쓸까 싶지만 생각보다 자주 사용되니 꼭 알아 둡시다.

b) 열거 자료형 (sequence types)

프로그램을 만들다보면 데이터를 한 건씩 저장하는 것 보다는 여러 개의 데이터를 한번 에 저장하거나 불러와야 하는 경우가 많습니다. 그럴 때 우리를 도와주는 것이 바로 열거 자료형입니다. 열거 자료형의 데이터에는 서로 다른 형태를 가진 여러 개의 데이터를 묶어서 하나의 변수에 저장할 수 있습니다. 대표적인 열거 자료형인 리스트, 튜플, 딕셔너리는 형태와 기능에 약간씩의 차이가 있으니 살펴보겠습니다.

▶ 리스트 list

리스트에는 어떤 타입의 데이터라도 함께 묶어 나열하는 방식으로 저장할 수 있습니다. 리스트를 만들기 위해서는 대괄호 []를 사용하면 됩니다. 저장된 리스트의 모든 자료들은 다양한 함수들을 이용해서 한 번에, 또는 원하는 부분만 출력하거나 수정할 수 있습니다.



ex)

리스트명 = [요소0, 요소1, 요소2,]

리스트명[0] = 요소0, 리스트명[1] = 요소1, 리스트명[2] = 요소2, ...

▶ 튜플 tuple

소괄호 ()를 사용해서 만들 수 있는 튜플은 리스트와 그 모양과 기능이 비슷하지만 안에 들어 있는 값의 수정이나 삭제가 불가능하다는 특징이 있습니다. 따라서 순서나 내용이 바뀌어서는 안되는 데이터를 저장할 때 사용합니다. 튜플의 내용을 바꾸고 싶으면 변수에 새로운 튜플을 재 할당하는 방법을 사용할 수밖에 없습니다. 하나의 요소만 가지는 튜플을 만들 때는 반드시 마지막에 쉼표를 붙여줍니다.

ex)

튜플명 = (요소0,)

▶ 딕셔너리 dictionary

딕셔너리는 데이터를 한 쌍의 키(key)와 값(value)으로 저장합니다. 데이터베이스에 바로 저장할 수 있도록 레이블링이 필요한 데이터를 저장할 때 사용합니다. 중괄호 { }와 콜론 :을 사용해 만들 수 있습니다.

ex)

user_1 = {"ID": "001", "이름": "홍길동", "주소": "서울", "전화번호": "01012345678"}

(3) 산술 연산자

연산자	기능	예제
+	덧셈	1+2 = 3
-	뺄셈	3-2 = 1
*	곱셈	2*4 = 8
**	제곱	4**2 = 16
/	나눗셈	7/2 = 3.5
//	나눈 몫	7//2 = 3
%	나눈 나머지	7%2 = 1

(4) 비교 & 논리 연산자

연산자	기능	예제
==	양쪽 값이 같다	1 == 1 → True
!=	양쪽 값이 같지 않다	1 != 2 → True
>	왼쪽 값이 오른쪽 값보다 크다	3 > 1 → True
<	왼쪽 값이 오른쪽 값보다 작다	1 < 5 → True
>=	왼쪽 값이 오른쪽 값보다 크거나 같다	1 >= 0 → True
<=	왼쪽 값이 오른쪽 값보다 작거나 같다	0 <= 1 → True
and	둘 다 참이면 참이다	(1 == 1) and (1 > 0) → True
or	둘 중 하나라도 참이면 참이다	(1 == 1) or (1 > 2) → True
not	거짓과 참을 반전한다	not (1 == 0) → True
in	왼쪽 값이 리스트에 포함되어있다	'a' in ['a', 'b', 'c'] → True
not in	왼쪽 값이 리스트에 포함되어있지 않다	'd' not in ['a', 'b', 'c'] → True

(5) 할당 연산자

연산자	기능	예제
=	왼쪽 변수에 오른쪽 값을 할당	apple = 3
+=	왼쪽 변수의 값에 오른쪽 값을 더한 후 할당	apple += 2
-=	왼쪽 변수의 값에서 오른쪽 값을 뺀 후 할당	apple -= 4
*=	왼쪽 변수의 값에 오른쪽 값을 곱해 할당	apple *= 8
/=	왼쪽 변수의 값을 오른쪽 값으로 나눈 값을 할당	apple /= 2
%=	왼쪽 변수의 값을 오른쪽 값으로 나눈 나머지를 할당	apple %= 5
**=	왼쪽 변수의 값을 오른쪽 값만큼 제곱하고 할당	apple **= 2
//=	왼쪽 변수의 값을 오른쪽 값으로 나눈 몫을 할당	apple //= 3

▲ 실습

[문제1] 변수 선언과 값 할당하기

마음대로 변수를 만들고 값을 할당해봅시다.

```
ex)
my_name = "이은재"
number = 20140040
```

[문제2] type함수 사용하기

문제1에서 만든 변수의 자료형을 알아보는 type함수를 사용해 보세요.

```
ex)
type(my_name)
type(number)
```

[문제3] print함수 사용하기

문제1에서 만든 변수를 print 함수로 하나씩 출력해 보세요.

```
ex)
print(my_name)
print(number)
```

[문제4] print함수 사용하기2

문제1에서 만든 변수들을 한 번에 출력해 보세요.

```
ex)
print(my_name, number)
```

[문제5] 데이터 다루기

문제1에서 만든 두 변수에 저장된 값을 서로 바꾸어 출력해 보세요.

```
ex)
my_name, number = number, my_name
```

[문제6~9] 데이터 활용하기

중앙대 앞 카페 터방내의 메뉴별 가격은 다음과 같습니다.

메뉴	가격
파르페	4,500
핫초코	3,500
팥빙수	6,000

[문제6] 터방내 메뉴들의 평균가격을 구하고 출력해봅시다.

[문제7] 영희와 친구들은 파르페 2잔, 핫초코 2잔, 팥빙수 1그릇을 먹었습니다. 계산해야할 가격을 구하고 출력해봅시다.

[문제8] 철수와 친구들은 파르페 1잔, 핫초코 3잔, 팥빙수 2그릇을 먹었습니다. 그런데 사실 사장님의 조카였던 철수는 30퍼센트를 할인 받았습니다. 계산해야할 가격을 구하고 출력해 봅시다.

[문제9] '영희네 그룹보다 철수네 그룹이 더 많은 돈을 지불해야한다'는 문장이 사실인지 판별하고 그 결과를 출력해봅시다.

[문제10] 나눗셈의 몫 구하기

20을 7로 나누었을 때 그 몫을 출력해 봅시다.

[문제11] 나눗셈의 나머지 구하기

20을 7로 나누었을 때 그 나머지 값을 할당연산자를 사용해 저장한 후 출력해 봅시다.

[문제12] 홀수 짝수 골라내기

주어진 자연수가 홀수인지 짝수인지 판별할 수 있는 방법을 생각해 봅시다.

[문제13] 문자열 출력하기

다음 문자열을 출력해 봅시다.

"인간들을 사랑하기 때문이오." 차라투스트라는 이렇게 말했다.

[문제14] 문자열 출력하기 2

다음 문자열을 출력해 봅시다.

첫 번째 줄에 뭐라고 써야할 지
두 번째 줄을 쓸 때까지도 모르겠다.

