

# 1주차 수업 내용

- 프로그래밍 소개
- 파이썬 소개
- 설치
- 개발환경
- 파이썬 사용법 기초

# 컴퓨터의 구성요소

- 컴퓨터는 다양한 데이터를 받아 계산, 저장, 처리하여 유용한 정보를 제공

## 하드웨어: 물리적인 기계 장치 (연산/제어장치, 기억장치, 입출력 장치)

- 연산 및 제어장치 (CPU)
- 기억장치 (주기억장치, 보조기억장치)
- 입출력 장치 (키보드, 마우스, 모니터, 프린터)

## 소프트웨어: 하드웨어를 동작시키는 프로그램의 집합

- 시스템 소프트웨어 (운영체제)
- 응용 소프트웨어 (문서 작성, 계산, 그래픽 처리 등)

# 프로그래밍 언어

- 사람이 원하는 것을 컴퓨터로 실행시키기 위해 만든 언어
- 프로그래밍(programming): 프로그램을 만드는 과정
- 프로그래머(programmer): 프로그램을 만드는 사람
- 고급 언어 (파이썬, C, C++, 자바 등) vs. 저급 언어 (기계어, 어셈블리어)
- 프로그래밍 언어를 배우는 이유: 논리적인 사고를 표현하기 위해

## 프로그래밍 경험 유무?

- 파이썬, C, Java, Visual Basic 등

# 컴파일러와 인터프리터

- 프로그래머가 프로그래밍 언어를 이용해 프로그램을 작성하면 컴파일러(또는 인터프리터)라는 소프트웨어가 프로그램을 기계어로 변환
- 컴파일러는 전체 코드를 전부 번역한 후에 실행
- 인터프리터는 코드 한줄씩 기계어로 번역하여 실행 (이전에 실행했던 내용을 기억)
- 컴파일러는 책의 전체 번역에 비유, 인터프리터는 한 문장씩 동시 통역에 비유

# 컴퓨터에서 자료의 표현

- 비트(bit): 컴퓨터에서 표현할 수 있는 자료의 최소 단위 (0 또는 1)
- 바이트(byte): 8개의 비트가 모인 것
- 문자 표현: 아스키(ASCII) 코드 사용
- 숫자 표현: 10진수를 2진수로 변환하여 표현
- 영상 표현: 화소(pixel) 단위로 표현, 흑백은 1 바이트, 컬러는 3 바이트 사용

## 컴퓨터 역사를 흥미있게 알 수 있는 자료

만화로 나누는 자유/오픈소스 소프트웨어 이야기 (총 52편)

- <https://joone.net/2016/11/05/찰스-배비지와-에이다-러브레이스/> (연결 찾기 어려움)
- <https://joone.net/2016/11/15/앨런-튜링과-폰노이만/>

틈날 때 마다 모두 읽어 보기 바람

# 파이썬(python) 언어 소개

- 1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그래밍 언어
- BBC의 Monty Python's Flying Circus 라는 TV 프로그램의 이름을 따서 지었다고 함
- 로고는 비단뱀이 사용되고 있음

# 파이썬 언어의 장점

- 문법이 쉬워 초보자도 쉽게 배울 수 있음
- 우수한 자료형과 다양한 모듈, 패키지 등을 제공
- 개발 기간을 단축할 수 있어 생산성이 좋음
- 비영리의 파이썬 소프트웨어 재단이 관리하는 오픈 소스로 제공됨
- 다양한 플랫폼(여러 운영체제)에서 사용 가능

## 단점

- 인터프리터 언어이기 때문에 속도가 느림
- 모바일 컴퓨팅의 지원이 약함
- 하드웨어 제어가 비교적 쉽지 않음



## 파이썬 언어 활용분야

- Web Development: Frameworks like Django, Flask
- Data Science and Analysis: Libraries like Pandas, NumPy, Matplotlib
- Machine Learning and AI: TensorFlow, PyTorch, Scikit-learn
- Automation and Scripting: Automate repetitive tasks
- Game Development: Libraries like Pygame
- Web Scraping: Tools like BeautifulSoup, Scrapy
- Desktop Applications: GUI frameworks like Tkinter, PyQt
- Scientific Computing: SciPy, SymPy
- Internet of Things (IoT): MicroPython, Raspberry Pi
- DevOps and Cloud: Automation scripts and APIs
- Cybersecurity: Penetration testing and ethical hacking tools

<https://www.geeksforgeeks.org/introduction-to-python/>

- Key Features of Python
- Why learn python? for beginners / for experienced
- Famous Application Built using Python

## 파이썬 관련 참고자료

- <https://www.w3schools.com/python/default.asp> : 항목별로 정리된 파이썬 튜토리얼
- <https://www.geeksforgeeks.org/introduction-to-python/>
- <https://joone.net/2023/04/14/52-python-프로그래밍-언어-1/>
- <https://joone.net/2023/04/14/52-python-프로그래밍-언어-2/>

# 파이썬 설치

- python homepage <https://www.python.org> 접속 후 Downloads 메뉴를 눌러 원하는 버전으로 설치 (2025. 2월 기준 Python 3.13.2)
- 설치할 때 파이썬 설치 폴더의 위치를 PATH에 추가하도록 설정하는 것을 강력 권고함



# 파이썬의 여러 실행 방법

파이썬은 여러 방법으로 실행할 수 있다.

- IDLE
- DOS 명령어창(shell, 셸, 쉘)
- 구글 코랩(Colab)
- VScode
- 이외에도 아나콘다, PyCharm 등

# IDLE

- IDLE은 파이썬에 기본 제공되는 통합 개발 환경  
(Integrated Development & Learning Environment)
- IDLE에서 프로그램을 입력하고 실행 가능
- IDLE 실행하기: <시작> - 'P' 항목에서 IDLE 클릭(바탕화면에 바로가기를 만들면 편리)

# IDLE 사용

- IDLE은 대화형 파이썬 shell(셸, 셸) 모드(>>>)와 파일을 편집하고 저장 실행하는 편집기 모드로 사용 가능 (시연)
- shell 모드와 편집기 모드를 구분할 줄 알아야 함

# 산술 연산

```
>>> 1+2 # 덧셈
3
>>> 4-2 # 뺄셈
2
>>> 2*4 # 곱셈
8
>>> 5/2 # 실수 나눗셈
2.5
>>> 5//2 # 정수 나눗셈 (몫을 계산)
2
>>> 5%2 # 나머지 계산
1
>>> 2**4 # 거듭제곱
16
```

- shell에서 옆의 여러 연산을 직접 입력해 실행해 결과를 확인해 보자
- 총 7종류의 산술연산
- #기호부터 줄의 끝까지는 주석문(프로그램의 실행 결과와 무관)



# 산술 연산

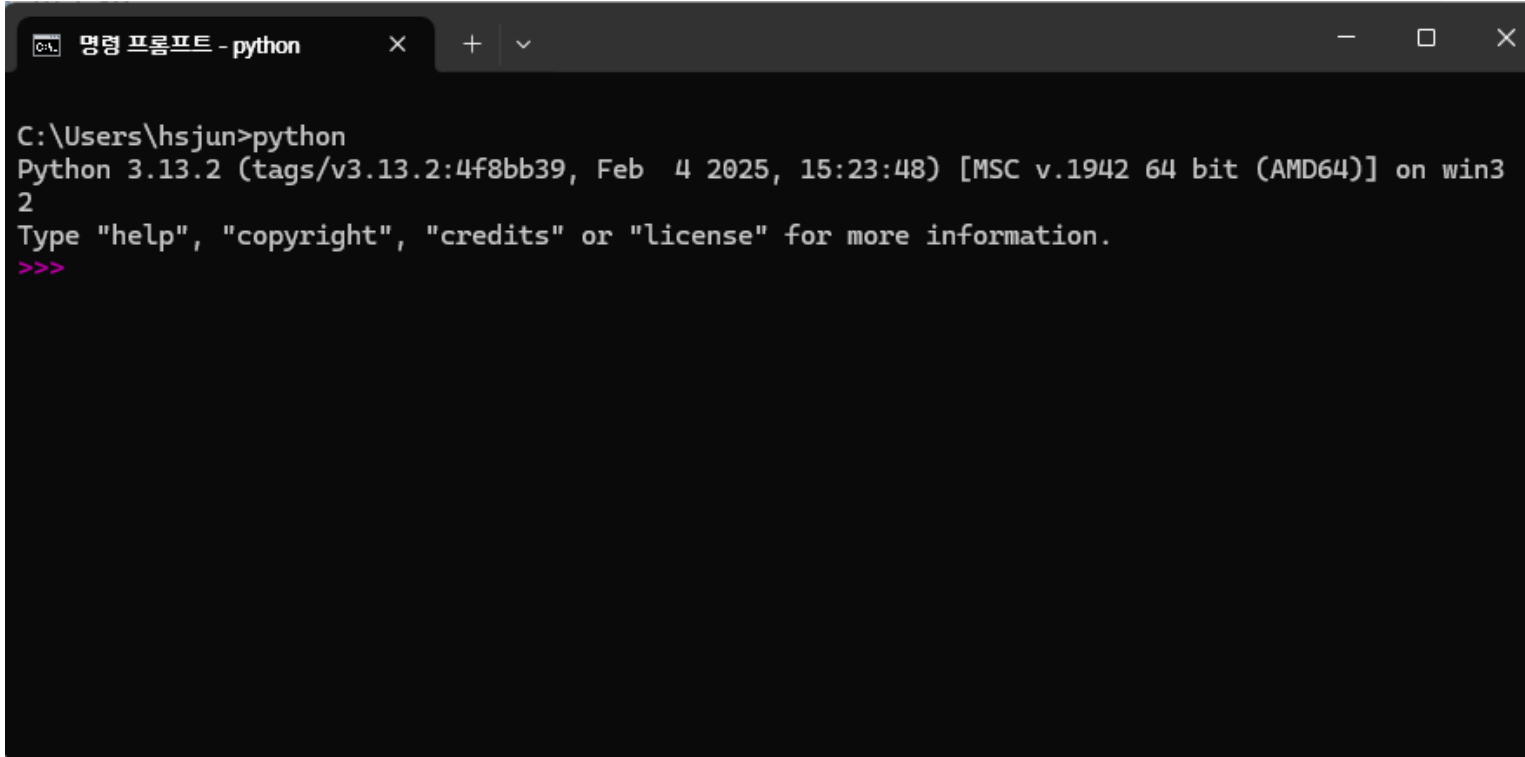
```
>>> 50 - 5*6 # 곱셈 먼저, 그 다음에 뺄셈
20
>>> (50 - 5*6) / 4 # 연산 순서-괄호 먼저
5.0
>>> 8 / 5 # 실수 나눗셈하면 실수가 얻어짐
1.6
>>> 17 // 3 # 정수 나눗셈은 소수점이하를 버림
5
>>> 17 % 3 # % 연산자는 나눗셈의 나머지를 남김
2
>>> 2 ** 7 # 2의 7 제곱
128
```

- 계산의 순서는 괄호안 먼저
- 연산자 우선 순위는 거듭제곱, 곱셈, 나눗셈, 덧셈 순서
- 같은 결과를 print() 함수를 이용해 얻을 수도 있다

```
>>> print(2+3)
5
```

# 명령어(DOS)창에서 파이썬 shell 사용

- 명령어 창을 연 후 python 입력



```
C:\Users\hsjun>python
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- 명령어 창의 shell에서도 IDLE의 shell과 같이 입력 가능

# 구글 코랩 이용

- 구글 계정 있어야 함. 계정 없으면 만들기 바람
- <https://colab.research.google.com>
- 간단한 시연

# 문자열(string)

- 큰따옴표 또는 작은따옴표로 둘러싸인 글자들
- 문자열 합치기

```
>>> 'my ' + 'dog'  
'my dog'
```

- 문자열의 반복 (문자열에 \*을 적용하면 반복을 의미)

```
>>> 'abc' * 3  
'abccabccabc'
```

# 대화형 모드와 스크립트 모드

- 대화형 모드(interactive mode):  
콘솔( `>>>` )에서 문장을 한 줄씩 입력하여 실행
- 스크립트 모드(script mode):  
프로그램 파일을 저장 후 파이썬 인터프리터가 이 파일의 문장들을 차례대로 전부 실행

# print() 함수

- print() – 괄호() 속의 내용을 화면에 출력하는 내장 함수(built-in function)
- 문자열, 수식 계산 결과 등의 출력 가능

```
>>> print('hello python')
hello python
>>> print(2*4)
8
```

# 프로그램

- 프로그램은 보통 변수, 연산자, 예약어, 함수 호출 문장 등으로 구성됨
- 변수: 값을 저장하는 문자열
- 연산자: `=, +, -` 등
- 예약어 : if, for 등

# 예약어(keyword, reserved word)

- 파이썬 언어에서 기능과 사용 용도가 이미 정해진 문자열(단어)
- 예약어는 변수명, 함수명, 클래스명으로 사용할 수 없음
- 예약어 확인하기

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await',
'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return',
'try', 'while', 'with', 'yield']
```



# 식별자(identifier)

- 식별자는 프로그램에서 사용할 변수, 사용자 정의 함수, 클래스, 모듈 등의 이름을 지정할 때 사용
- 식별자 이름은 다음과 같은 규칙을 따라 붙인다.
  - 영문자, 숫자, 밑줄(\_), 유니코드문자를 사용할 수 있으나 **숫자로 시작할 수는 없음**
  - 영문 대문자와 소문자는 구별됨 (예: name과 Name은 다름)
  - 예약어를 명칭으로 사용할 수 없음
  - 한글도 유니코드 문자이므로 사용할 수는 있지만 **가급적 사용하지 않는 게 좋음**

# 변수(variable)

- 프로그램에서 실행 중에 변할 수 있는 값을 가지고 있는 것
- 변수는 어떤 값을 저장하는 메모리 공간을 말하며 일상 생활의 그릇 또는 상자에 비유 가능
- 그릇, 상자의 크기가 여러 종류이듯이 프로그램에서도 여러 종류의 변수가 사용됨
- 파이썬은 C나 자바 언어와 달리 변수를 사용하기 전에 미리 선언할 필요 없음

# 변수이름의 명명 규칙

- 앞의 식별자 규칙을 따라야 함
- 보통 변수이름에는 소문자, 상수이름에는 대문자를 사용
- 변수명은 관련있는 이름의 사용을 권장 (예: 면적을 표시하려면 area)
- 변수명이 긴 경우 밑줄(\_)을 사용해 읽고 구분하기 쉽게 함
- 또는 중간에 대문자를 사용해 단어 경계를 표시(낙타체: 중간에 대문자가 낙타의 혹처럼 보이기 때문)
- 지역변수에는 가급적 짧은 이름을, 전역변수에는 가급적 긴 이름을 사용 (나중에 배움)

## 변수의 사용 예

- 올바른 사용 예 :

Count, \_MaxSpeed, total\_amount, user\_name, userName, tree3

- 변수의 잘못된 사용 예 (이유):

if (예약된 키워드 사용), 72street(숫자로 시작), user name(공백 포함), user-name(중간에 '-' 문자 사용), total#amount(특수 문자의 사용)

# 들여쓰기(indent, indentation)

- 파이썬에서 공백은 주요한 역할을 한다
- 들여쓰기는 명령의 범위를 지정
- 같은 들여쓰기를 사용하는 명령의 집합을 블록이라 함
- 들여쓰기를 맞추지 않으면 에러 발생
- 빈칸 4개를 사용하는 것이 거의 표준임

```
>>> i = 5
>>>     j = 6
SyntaxError: unexpected indent
```

# 논리적/물리적 명령행

- 논리적 명령행은 파이썬 인터프리터 관점에서의 한 명령 단위 문장을 의미
- 물리적 명령행이란 프로그램 코드 내에 직접 표현된 한 줄을 의미
- 파이썬에서 보통 하나의 논리 문장을 한 라인에 표시
- 2개의 논리 문장을 2 라인으로 표시

```
i = 10  
print(i)
```

- 2개의 논리 문장을 한 라인으로 표시. 세미콜론으로 구분함

```
i = 10; print(i);
```

## 간단한 계산과 출력

- 임의의 식에 값을 넣어 할 수 있는 계산은 모두 가능함
- 예) 가로가 4, 세로가 3인 직사각형의 넓이 계산

```
>>> 4 * 3
12
>>> w = 4
>>> h = 3
>>> a = w*h
12
```

# 비교연산자

- 크기와 같음/다름을 비교하는 연산자

> >= < <= == !=



## Homeworks

- 집 컴퓨터에 파이썬 설치
- 오늘 수업내용 복습
- 다음 주 수업내용 예습