컴퓨터 및 프로그래밍입문(005) 1주차

한상곤(sangkon@pusan.ac.kr)

0. 강의 소개

- 교과 : 컴퓨터 및 프로그래밍 입문(005)
- 학과 : 컴퓨터공학전공
- 담당 : 한상곤(sangkon@pusan.ac.kr)
- 평가 : 중간고사 20% / 기말고사 40% /실습 및 과제 20% / 출결 20%
- 주의사항
  - 과제 및 중간고사 진행 시 '코드 실행시 에러/오류' 발생하지 않도록 주의
  - 출결 미달 및 중간/기말 미응시 하지 않도록 주의
  - 노트북 및 컴퓨터 고장은 구매처에 문의, 그것으로 인한 불이익이 없도록 주의
  - 문의사항은 이메일(sangkon@pusan.ac.kr)로 문의 바람

0. 강의 소개 - 수업 세부 사항

- 수업관련
  - 수업의 모든 내용은 교재(`"으뜸 파이썬," 박동규/강영민, 생능출판사, 2020.`)를 우선
  - 추가적으로 제시하는 내용은 별도의 `Ref.`를 제시하고 수업에 활용
  - 중간/기말 고사는 '교재' 및 실습 내용을 중점으로 제출

0. (예상) 교재 목차

1. 파이썬 소개(01장)
2. 변수와 연산자(02장)
3. 제어문(03장)
4. 리스트(05장)
5. 함수와 입출력(04장)
6. 예외 처리와 파일(08장)
7. 딕셔너리 / 튜플(06장)
8. 모듈과 활용(07장)
9. 클래스와 객체 지향 프로그래밍(09장)

Prequel. 컴퓨터란 무엇인가?

컴퓨터와 프로그래밍

"Computer Science is no more about computers than astronomy is about telescopes(컴퓨터과학의 대상은 컴퓨터가 아니다. 천문학의 대상이 망원경이 아닌 것처럼) - Edsger Dijkstra(다익스트라)[1], 1930.05.11.~2002.08.06.

---

1. a. 네덜란드 출신의 컴퓨터과학자, 1972년 전산학 및 프로그래밍 언어 분야에 대한 공헌을 인정받아 튜링상을 수상, 대표적인 업적은 다익스트라 알고리즘을 개발하여 최단 경로 알고리즘 문제(`Shortest Path Problem`)에 대한 학문적 접근을 제시, 다익스트라 알고리즘은 학부에서 자료구조와 알고리즘을 배울 때 꼭 짚고 넘어가는 아주 중요한 알고리즘 중 하나임, b 세마포어 개념을 정립하여 임계 구역 문제에 대한 하나의 솔루션을 제시, 1968년에는 GOTO문의 해로움이라는 논문을 발표하여 프로그램을 여러 단위로 나누고 서로가 서로를 호출하는 구조적 프로그래밍이라는 새로운 패러다임을 제시, 해당 연구 덕분에 프로그래머들은 각 서브프로그램의 관계만 알면 코드를 쉽게 수정할 수 있게 되었고 이는 훗날 객체 지향 프로그래밍과 같은 패러다임이 등장하는 데 큰 영향을 줌

앨런 튜링과 컴퓨터의 발명

오늘날 튜링의 업적이 왜 의미가 있는지 이해하기 위해서는 그가 당대의 가장 큰 수학 난제 중 하나를 어떻게 풀려고 했는지, 그리고 그 과정에서 어떻게 모든 컴퓨터의 기본을 정의했는지에 대한 이야기를 해야 합니다.

제2차 세계대전 전까지, 컴퓨터(혹은 계산원)^{computer}란 단어는 수작업으로 또는 기계적 계산기를 써서 계산을 하는 사람을 지칭하였습니다. 컴퓨터(계산원)은 산업혁명의 중요한 축을 담당했고 종종 상용로그표를 만들기 위해 필요한 계산과 같은 반복적인 작업을 수행하였습니다.

결정문제(Decision problem, Entscheidungsproblem)

1928년 수학자 다비트 힐베르트가 제안한 문제(독일어로 Entscheidungsproblem, 일반적으론 결정 문제라 함)입니다. 힐베르트는 $2 + 2 = 4$와 같은 수학적 명제가 모두 결정가능성(decidability)[2]이 가지고 있는지 알고자 했습니다.

"어떤 수학적 명제가 주어져도 이 식이 참인지 거짓인지 결정할 수 있는 단계적 절차가 존재하는가?"

2. 논리학에서 결정가능성(decidability)은 해답을 내어놓는 기계적인 절차가 존재하는가에 관한 성질을 뜻합니다.

힐베르트가 말한 단계적 절차는 오늘날 `알고리듬(algorithm)`[3]이라고 부르기도 합니다. 하지만 1930년대에는 컴퓨터도 없었고 프로그램도 없었기 때문에 튜링은 '결정문제'를 공략하기 위해 `컴퓨터연산(computation)` 등과 같은 기초적인 개념 및 도구를 정의해야 하였습니다.

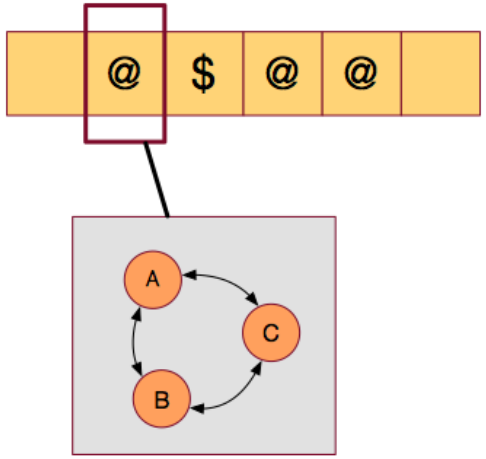
튜링이 결정문제를 해결하기 위해서 작성한 논문으로 해당 논문의 결론은 `"멈춤문제(Halting Problem)를 풀 수 있는 튜링기계는 없다"`입니다. 그런데 `튜링기계`(Turing Machine)가 뭘까요? 그리고 앞선 결정문제가 아닌 `멈춤문제`(Halting Problem)를 풀 수 없다는 것은 무슨 뜻일까요?

---

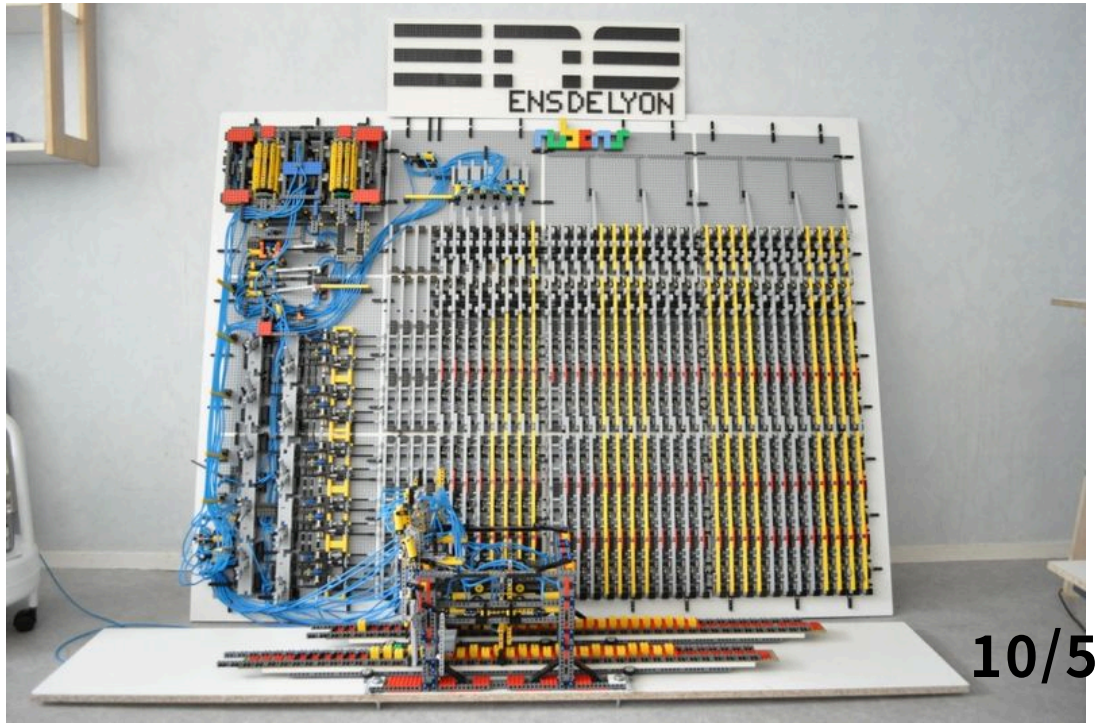
3. 알고리즘(algorism, 숫자를 이용한 연산)과 알고리듬(algorithm, 연산의 단계나 과정)은 전혀 다른 의미를 가지고 있지만, 한국의 경우 알고리듬을 알고리즘으로 발음 및 표기하는 문헌이 많기 때문에 두 단어에 대해서 '문맥(context)'에 맞게 이해하도록 하자.

튜링기계(Turing Machine)

1. 옆의 그림에서 왼쪽 상단은 테이프를 의미하며, 데이터를 읽고 쓸 수 있습니다.
2. 붉은 색 박스가 헤드(head)이며, 이것을 움직여서 테이프 위치를 지정할 수 있습니다.
3. 왼쪽 하단의 박스가 상태(state)를 나타내는 상자이며, 현재 {A,B,C} 중 하나의 상태를 표시합니다.
4. 오른쪽이 액션 테이블로 각 상태에서 전이 가능한 행동들이 기술되어 있습니다.



Current State	Read Symbol	Write Symbol	Next Position	Next State
A	@	@	->	A
A	$	$	->	B
B	@	@	<-	C
C	$	@	->	C
C	@	$	->	B



멈춤문제(Halting Problem)

멈춤문제는 결정문제의 한 갈래로, "주어진 프로그램이 해결하고자 하는 문제가 해결 가능한지 말해줄 수 있는 일반화된 알고리즘이 존재하는가?" 라는 질문입니다. 아래와 같이 "정지문제를 판별하는 알고리즘이 있다"고 가정해보겠습니다.

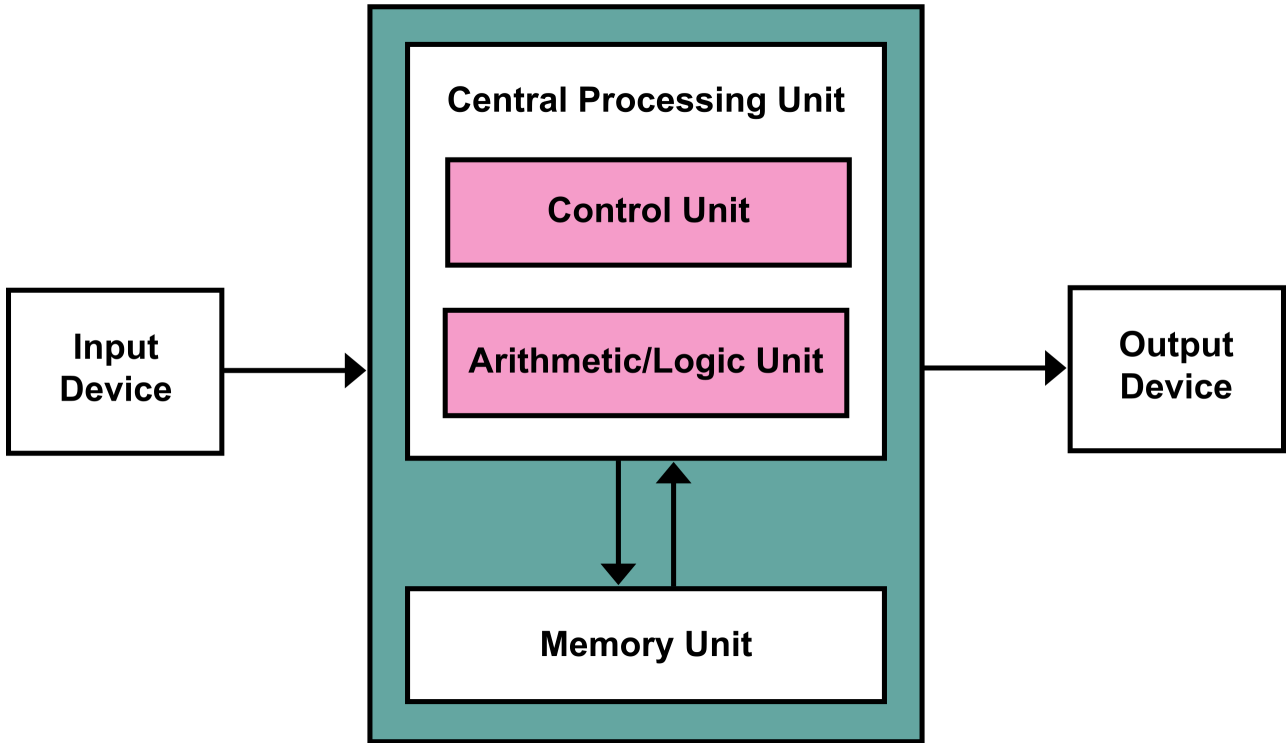
```
def K(P):
    if (Halts(P, P)):     # P 프로그램에 P를 입력으로 사용
        while True: pass # 멈추지 않음
    else:
        return 42         # 멈춤
```

1. 만약 Halts(K,K) is 참(True), 이면 K(K) 멈추지 않음
2. 만약 Halts(K,K) is 거짓(False), 이면 K(K) 멈춤
   => 어느 쪽이든, Halts(K,K)는 잘못되었음 => 따라서, Halts 함수는 존재할 수 없음

Von-Neumann architecture

에드박의 보고서 최초 초안(First Draft of a Report on the EDVAC)에서 수학자이자 물리학자 존 폰 노이만과 다른 사람들이 서술한 1945년 설명에 기반한 컴퓨터 아키텍처로 CPU, 메모리, 프로그램 구조를 갖는 범용 컴퓨터 구조로 정의됩니다.

프로그램 내장 방식 컴퓨터를 제시하였고, 이후에 나온 컴퓨터는 모두 폰 노이만의 설계를 기본 구조로 디자인되고 있습니다.



Von-Neumann and Turuing / Bottleneck

- Von-Neumann and Turuing
  - 폰노이만 기계의 작동은 튜링 기계로 표현가능
  - 튜링기계의 작동은 폰노이만 기계로 표현가능(단, 메모리가 무한해야 함)
- Von-Neumann Bottleneck
  - 일반적으로 경로의 병목현상 또는 지연 현상을 이르는데, 이는 나열된 명령을 순차적으로 수행하고, 그 명령은 일정한 기억장소의 값을 변경하는 작업으로 구성되는 폰 노이만 구조에서 기인

우리에게 컴퓨터란?

- CPU(연산장치)
  - 연산은 순차적으로 진행된다.
- Memory(주기억장치)
  - 모든 프로그램은 메모리에서 실행
- Input(입력)
  - 코드
- Output(출력)
  - 결과
- 모든 프로그램은 특수한 상황을 제외하고 일반적으로 종료되어야 함

1. Welcome to Python!

1. 파이썬 소개

1. ~~일상생활에서 경험하는 프로그램과 소프트웨어~~
2. 파이썬 소개
3. 파이썬 설치와 사용
4. ~~파이썬 대화창 사용하기~~
5. `Hello World` 출력하기
6. 연습문제

1. 파이썬 소개 - 학습목표

- 아래 용어(term)을 이해하고, 설명할 수 있다.
  - 프로그램, 프로그래밍, 소프트웨어, 하드웨어
- 파이썬의 역사와 파이썬이 가진 특징을 이해한다.
  - 컴파일러와 인터프리터의 차이를 이해한다.
- 파이썬의 설치방법과 사용법을 이해하고 할 수 있다.
  - 파이썬 프로그램의 **명령이 해석되고 실행되는 과정**에 대해 이해한다.
  - 파이썬 **통합 개발환경**의 필요성을 이해하고, 활용할 수 있다.
- 간단한 파이썬 프로그램을 만들 수 있다.
  - 파이썬 프로그램을 통해 원하는 메시지를 출력할 수 있다.
- ~~대화형 모드를 이용하여 간단한 계산을 수행할 수 있다.~~

1.1 일상생활에서 경험하는 프로그램

- 스마트폰에는 시계의 기능이 내장되어 있다. 시계 프로그램에 "스마트폰의 시간이 오전 7시가 되면 알람 소리를 들려주도록 하라"라는 명령을 넣을 수 있다.
- 에어콘에는 온도계의 기능이 내장되어 있다. 온도계 프로그램에 "온도가 30도가 되면 에어콘을 켜라"라는 명령을 넣을 수 있다.

하드웨어

- 중앙처리장치(혹은 CPU), 저장 장치, 출력장치, 입력장치 등으로 구성
  - 컴퓨터와 스마트폰 정보통신 기계의 물리적 부품
  - 컴퓨터와 스마트폰, 태블릿 등

소프트웨어

- 하드웨어에서 구동되는 특정한 프로그램

운영체제

- 응용 프로그램이 동작할 수 있도록 도와주는 컴퓨터 프로그램
  - 하드웨어를 관리하여 응용 프로그램이 실행될 수 있는 환경을 제공
  - 윈도우 11, 리눅스, 맥오에스(macOS) 등

응용 프로그램

- 하드웨어에서 수행될 아래아 한글, 파워 포인트, 엑셀, 크롬 브라우저와 같은 응용 프로그램

1.1.c 응용 소프트웨어를 활용

스마트폰은 지정된 명령어를 수행할 수 있는 하드웨어와 소프트웨어가 있기 때문에 명령 수행이 가능

	Android	iOS
Development & maintained	Google LLC	Apple Incorporation
Released In	2008	2007
First Version	Android 1.0, Alpha	iPhones OS 1
Targeted Systems	Smartphones and tablets	Smartphones, music players
Kernel Type	Linux-based	Hybrid
License	Apache 2.0 and GNU GPLv2	Proprietary, APSL, GNU GPL
Programming Languages	C, C++, Java, Kotlin	C, C++, Object-C, assembly Language, Swift
Source Model	Open Source	Commercial Based
Customizability	Limited	Can Change Anything

프로그램(program)

컴퓨터가 실행할 특정한 작업을 지시하는 일련의 명령어들의 모임, 이 명령을 조직적으로 모아 놓은 것이다.

프로그래머(programmer)

프로그램을 작성하는 사람, 컴퓨터에 명령을 내리는 명령어를 작성할 수 있어야 한다.

프로그래밍(programming)

하나 이상의 명령어들을 입력하여 프로그램을 작성하는 과정, 다른 표현으로 '코딩'이라고도 한다.

왜 프로그래밍을 하는가?

- 컴퓨터는 0과 1의 이진 값만을 이해하고 저장
- 컴퓨터가 수행하는 명령은 0011100010001000 . . . 의 값으로 되어 있음
- 0011100010001000 . . .과 같은 명령은 사람이 이해하기 어렵고 작성 시간이 오래 걸리며 오류가 많고 수정이 힘듬
- 사람이 이해할 수 있는 프로그래밍 언어를 이용하여 명령 입력

1.1.g 프로그래밍의 예

- 프로그래머는 hello.py라는 이름의 파이썬 명령어를 이용한 프로그램을 작성
- 인터프리터라는 프로그램이 hello.py라는 프로그램을 기계어 명령어로 변환하여 컴퓨터에서 실행

```
print('Hello Python!!')
```

프로그래밍 언어

- 컴퓨터 시스템을 구동시키는 소프트웨어를 만들기 위한 형식을 제대로 갖춘 언어
  - 컴퓨터가 수행할 수 있어야 함
  - 자동차나 프린터 등의 미리 틀이 정해진 기계(임베디드 시스템Embedded System)에서 제한된 목적과 용도로 사용되는 프로그래밍 언어는 수행 속도가 중요하므로 C 언어를 주로 사용
  - 웹 서비스를 위해서는 HTML과 같은 마크업 언어와 JavaScript, PHP와 같은 언어가 많이 사용
- 배우기 쉬움

Review(2023.03.06)

- 공리0. "$\forall P \Leftrightarrow$ Memory"
- 명제1. "연산은 순차적으로 진행된다."
- 명제2. "컴퓨터는 메모리와 연산장치로 구성된다"
- "하드웨어", "소프트웨어", "운영체제", "응용 소프트웨어"
- "프로그램", "프로그래머", "프로그래밍"
- "프로그래밍의 필요성"과 "프로그래밍 언어"

파이썬

- 1989년 귀도 반 로썸(Guido Van Rossum)에 의해 개발
- 인터프리터 방식의 객체지향 프로그래밍 언어

소스 코드(source code)

- 프로그래밍 언어로 작성된 명령어들의 목록

소스 파일(source file)

- 소스 코드가 저장된 파일

```
age = input('나이를 입력하세요')
if int(age) < 19:
    print('할인되었습니다.')
else:
    print('할인이 안 됩니다.')
```

컴파일 방식

- 프로그램 명령어를 기계어로 번역한 후 이 기계어를 실행하는 방식
- C, C++, 파스칼 언어등이 있음


center

인터프리터 방식

- 프로그램 명령어를 한 번에 한 줄씩 읽어 번역한 후 바로 실행
- 파이썬, BASIC 등의 언어가 있음


center

인터프리터 방식과 컴파일 방식 비교

	인터프리터 방식	컴파일 방식
정의	명령어들을 한 번에 한 줄씩 읽어 들여서 실행하는 방식이다.	명령어를 기계어로 번역하는 방식이다.
장점	컴파일 단계를 거칠 필요가 없다.	일반적인 경우 속도가 더 빠르다.
단점	실행 시간이 느리다.	원시 프로그램의 크기가 크다면 상당한 시간이 소요된다.
사용되는 언어	파이썬, BASIC 등	C, FORTRAN, PASCAL 등

파이썬의 특징

1. 직관적이고 단순한 문법
2. 빠르게 학습 가능
3. 축약된 코드 작성 문법을 제공
4. 짧은 코드로 다양한 기능을 수행할 수 있음
5. 오픈소스^{Opensource} 방식을 채택하여 방대한 도구(라이브러리/프레임워크)를 무료로 활용할 수 있음
6. 객체지향 프로그래밍 언어의 특징도 제공함

1.3 파이썬 설치와 사용

- 홈페이지 접속 (http://www.python.org/)
- 파이썬 설치 파일 다운로드
  - 운영체제(macOS, Windows, Linux 등)를 선택 후 다운
- 설치시
  - "Install launcher for all users(recommended)"
  - "Add Python ..." 선택

1.4 파이썬 대화창 사용하기

- 시작 버튼을 눌러 "python"을 검색
- "Python"을 실행
- 사용자의 입력을 받을 수 있는 프롬프트prompt에 파이썬 명령어(커서가 깜박일 것이다)를 입력
- 프롬프트(`>>>`)에 print(`'Hello Python!!'`) 을 입력 후 엔터키
- 프롬프트 아래에 Hello Python!!이 출력


center

1.4.a 오류가 발생할 때

- 오류가 날 경우 오류가 발생한 문장과 함께 오류의 내용을 출력
- print(`'Hello Python!!"`) 와 같이 작은 따옴표로 시작해서 큰 따옴표로 끝나는 경우에는 파이썬이 잘못된 명령으로 인식해서 오류를 발생
- 들여쓰기와 대소문자에도 민감한 특징이 있음
- 대화창을 종료하고 싶으면 키보드의 Control 키와 'Z'키를 입력하거나(`^Z` 로 표기함) 또는 `exit()` 함수를 호출


center

1.4.b 파일 실행

1. C: 드라이버 아래에 workspace라는 이름의 폴더 생성
2. 이 폴더 아래에 hello.py라는 프로그램을 만들기
3. Windows 명령 처리기에서 다음 명령을 순차적으로 입력

```
C:\Users\me> c:
C:\Users\me> cd \ (주의 : 역슬래쉬 기호 \는 한글 윈도우에서 ₩로 나타남)
C:\> mkdir workspace
C:\> cd workspace
C:\workspace>
```

1.4.c hello.py 프로그램을 작성

1. hello.py 파일 만들기 위해서 메모장 활용

```
C:\workspace> notepad hello.py
```

2. 메모장에서 다음과 같은 내용을 입력하고 hello.py 작성

```
print("Hello Python")
```

3. dir 명령으로 현재 폴더에 hello.py 파일이 있는지 확인
4. 인터프리터 실행을 위해 python hello.py를 입력

```
C:\workspace> dir
C:\workspace> python hello.py
```

IDLE 활용

1. integrated development environment for Python
2. 파이썬 개발을 위한 통합 개발환경으로 파이썬 코딩, 실행, 디버깅, 다양한 설정이 가능해서 편리

VSCode 사용

1. 모든 수업은 `VSCode`(이하 VSC)를 활용해서 사용 자신의 컴퓨터(혹은 데스크탑)^{desktop}/노트북(혹은 랩탑)^{laptop}에 아래 2가지 프로그램을 반드시 설치(1주차 과제로 제출)

- Python $\leq$ 3.10.x
- VSCode $\leq$ 1.76.x

VSCode 확장 프로그램 설치

2. VSC에서 Extensions(`Ctrl+Shift+X`) 선택 후 python 검색
3. `Microsoft`에서 제작한 `Python` 선택 후 재실행
4. VSC에서 실행 후 File > Open Folder > 자신의 코드를 작성하는 위치 선택
5. VSC에서 View > Terminal(`Ctrl+`` `)

1.5 Hello World 출력하기

문자열이나 수치값을 화면상에 출력하는 중요한 일을 하는 함수인 `print()`는 print('Hello World')와 같은 코드를 표현문^{expressionstatement} 혹은 표현식이라고 하며, 표현식은 간단하게 문장^{statement}이라고 부르기도 함


center

1.5.a 문자열과 숫자 출력하기

```
>>> print('Hello World!')
Hello World!
>>> print('Hello', 'World!')
Hello World!
>>> print('Hello'+'World!')
HelloWorld!
>>> print(100)
100
>>> print(2+4)
6
```

1.5.b 계산하기

```
>>> 100
100
>>> 10 + 20
30
>>> 10 - 20
-10
>>> 10 * 20
200
>>> 10 / 20
0.5
>>> 123 * 8765
1078095
>>> 10 + 20 * 30
610
```

1.5.c 변수를 사용해서 연산

```
>>> x = 100
>>> y = 200
>>> x + y
300
>>> x - y
-100
>>> x * y
20000
>>> x / y
0.5
>>> x ** 2
10000
>>> n = 5
>>> n ** 2
25
```

연습문제 풀이

선택된 연습 문제 1.3

다음 파이썬 프로그램의 출력 결과는?

```
(1) print(100)
(2) print(100 + 200)
(3) print('100 + 200')
(4) print(100, 200)
(5) print('100', '200')
(6) print('Hello Python!')
(7) print('Hello','Python', '!')
(8) print('Hello'+'Python'+'!')
(9) print('**********')
(10) print('*' * 10)
```

선택된 연습 문제 1.6

다음과 같은 출력이 되도록 intro.py라는 프로그램을 작성한 후 파이썬 인터프리터에서 수행시켜 보시오(아래 코드에서 홍길동, 대한대학교등의 정보 대신에 실제 자신의 이름과 소속을 적어서 출력해 보세요).

```
**************************************
안녕하세요~
저는 홍길동입니다.
대한대학교 정보통신공학과 1학년입니다
**************************************
```

선택된 연습 문제 1.9

파이썬 대화창을 이용하여 다음과 같은 수식 계산을 해 보도록 하자

```
(1) 400 - 200
(2) 45 * 89
(3) 32 / 8
(4) 9 * 3
(5) 9 ** 3
(6) 9 / 3
(7) 9 // 3
(8) 9 % 3
```

선택된 연습 문제 1.15

100과 200의 두 값이 있을 경우, 이 값들의 평균값을 다음과 같이 출력하는 프로그램을 작성하시오.

```
100과 200의 평균값 : 150.0
```

선택된 연습 문제 1.16

50과 30의 두 값이 있을 경우, 이 값에 대한 사칙 연산을 다음과 같이 수행하는 프로그램을 작성하시오. 이때 각 출력문은 print('1 + 1 = ', 1 + 1) 과 같이 출력문과 연산을 쉼표로 구분하시오.

```
50 + 30 = 80
50 - 30 = 20
50 * 30 = 1500
50 / 30 = 1.6666666666666667
```