

1장. 프로그래밍 소개

숭실대학교
전자정보공학부 IT융합전공
담당교수: 권민혜
minhae@ssu.ac.kr

이번 장에서 학습할 내용

- 프로그램이란 무엇인지 이해한다.
- 프로그래밍 언어가 왜 필요한지 이해한다.
- 알고리즘이 무엇인지 이해한다.
- 프로그램의 개발 과정을 이해한다.

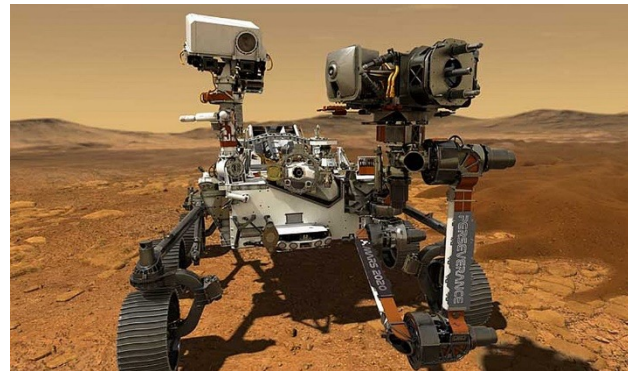
프로그램을 작성하기에 앞서서 중요한 개념들을 살펴봅니다!

이번 장에서 만들 프로그램

- "Hello World!"는 많은 사람들이 프로그래밍을 학습할 때, 가장 먼저 작성하는 프로그램이다.



"Hello, world. My first look at my forever home. "



프로그램이란?

- 프로그램 안에는 무엇이 들어 있을까?



프로그램

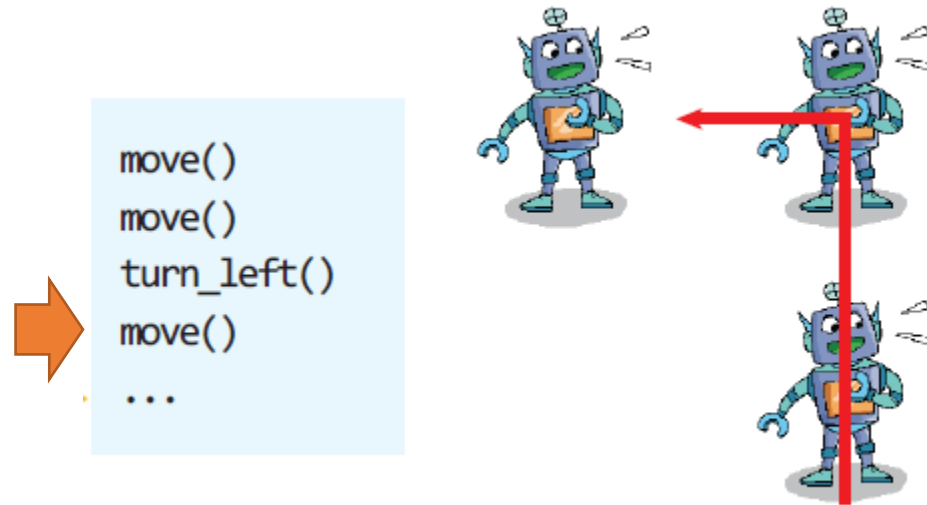
```
move()  
move()  
turn_left()  
move()  
...
```

프로그램은
CPU 명령어의
나열이다.

그림 1.1 프로그램은 작업 지시서와 같다.

명령어란?

- 로봇을 움직이는 프로그램이라면 로봇을 움직이기 위한 명령어들이 프로그램 안에 나열되어 있다.



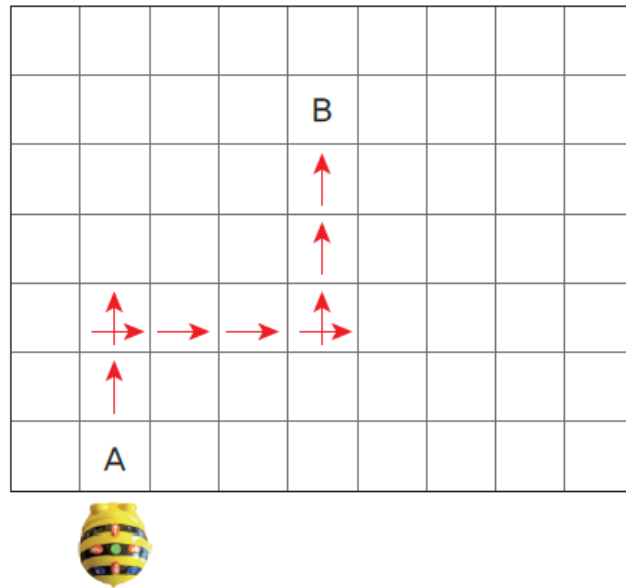
컴퓨터와 인간

- 컴퓨터는 강인공지능이 아니라면 프로그래머가 지시한 대로만 행동한다.

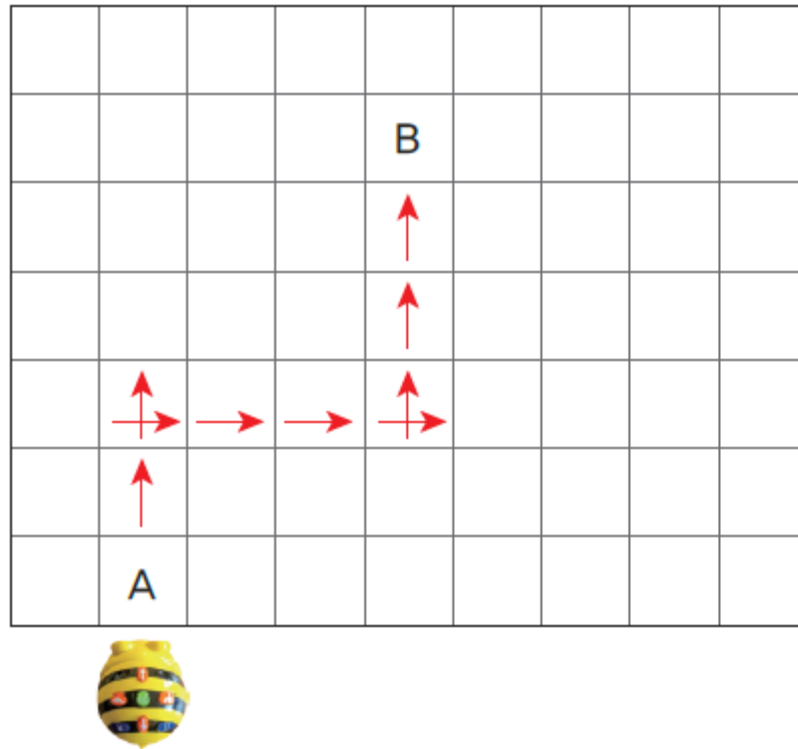


Lab: 로봇 움직이기

- 비봇(bee-bot)을 A 지점에서 B 지점으로 이동하게 하려면 어떤 명령어들을 어떤 순서로 입력해야 할까?



Sol: 로봇 움직이기

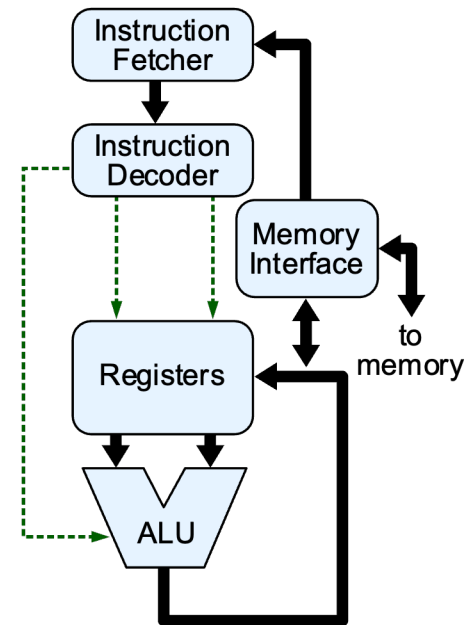


한 칸 전진
한 칸 전진
오른쪽으로 90도 회전
한 칸 전진
한 칸 전진
한 칸 전진
왼쪽으로 90도 회전
한 칸 전진
한 칸 전진

비봇에 대한 프로그램

실제 컴퓨터의 명령어

- 우리가 사용하는 컴퓨터의 명령어에는 어떤 것들이 있을까?
- 인텔의 x86 CPU가 제공하는 명령어에는 ADD, SUB, MOV, IMUL 등이 있다(아주 많음). ADD는 덧셈 연산이고 SUB는 뺄셈 연산, MOV는 데이터 이동 연산, IMUL은 곱셈 연산이다. CPU의 주된 동작은 계산과 데이터 이동이다.



컴퓨터가 이해하는 언어

- 한국어나 영어 등으로 작업을 지시한다면 컴퓨터는 전혀 이해할 수 없을 것이다.

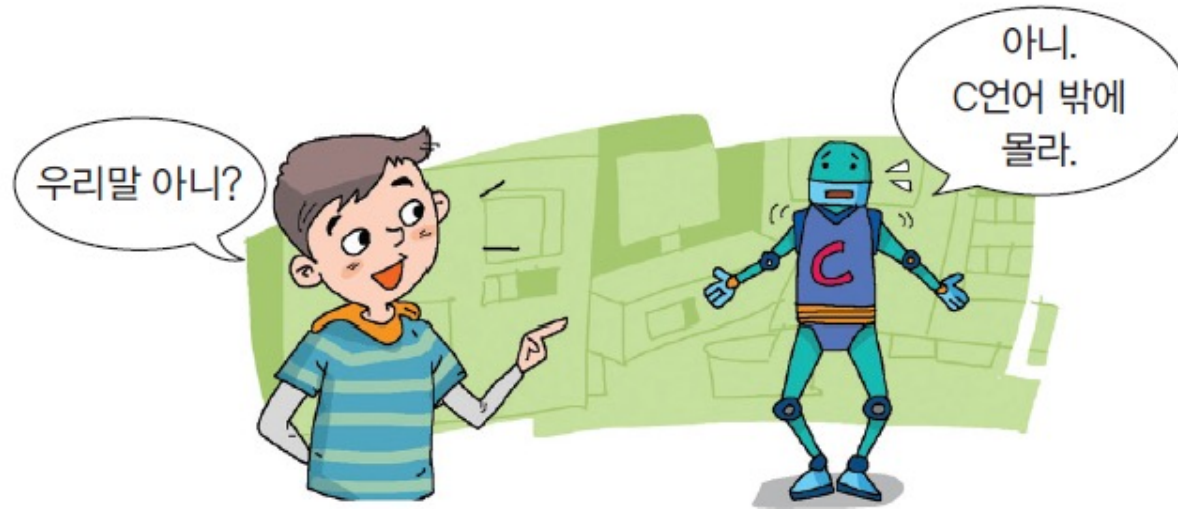
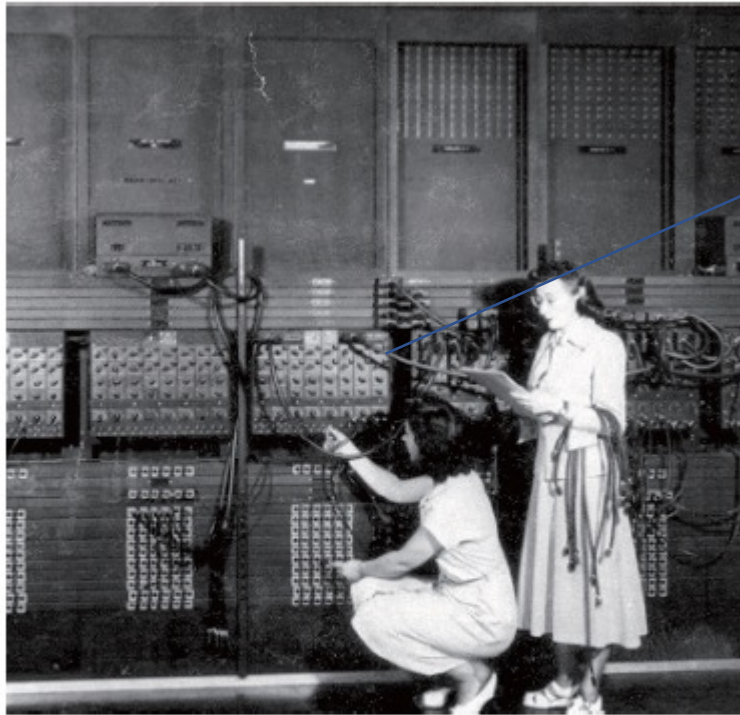


그림 1.2 컴퓨터는 한글로 된 작업 지시서는 이해하지 못하는 반면 기계어로 된 작업 지시서는 이해할 수 있다.

컴퓨터가 이해하는 언어

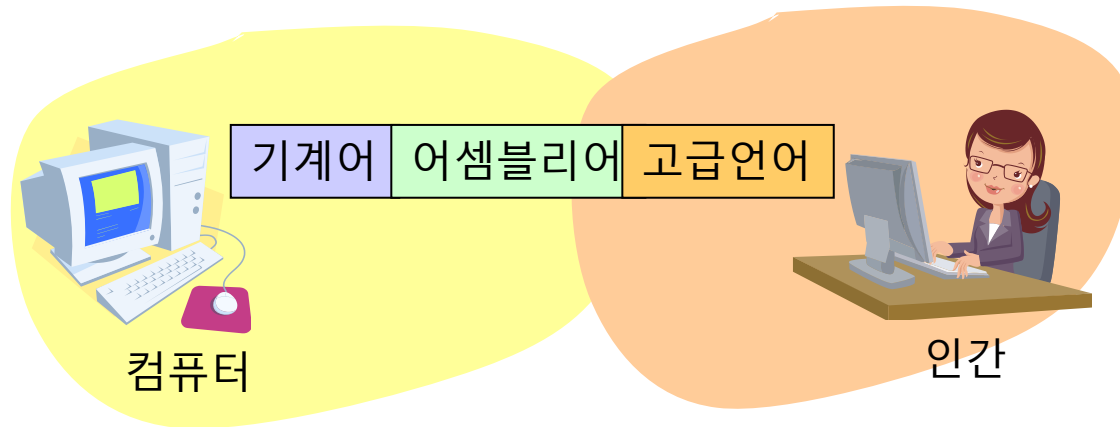
- 0과 1로 구성되어 "001101110001010..."과 같은 이진수로 된 기계어(machine language)이다.



전선을 연결하여
프로그래밍하고 있다.

프로그래밍 언어의 분류

- 기계어(machine language)
- 어셈블리어(assembly language)
- 고급 언어(high-level language)



컴파일러의 역할

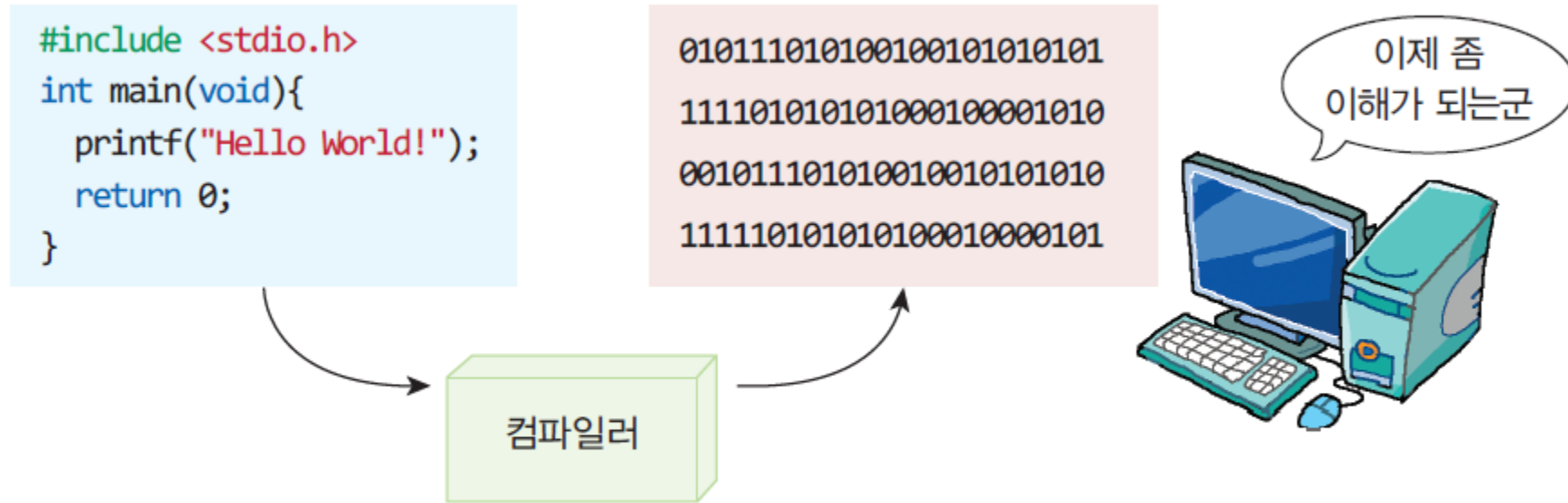
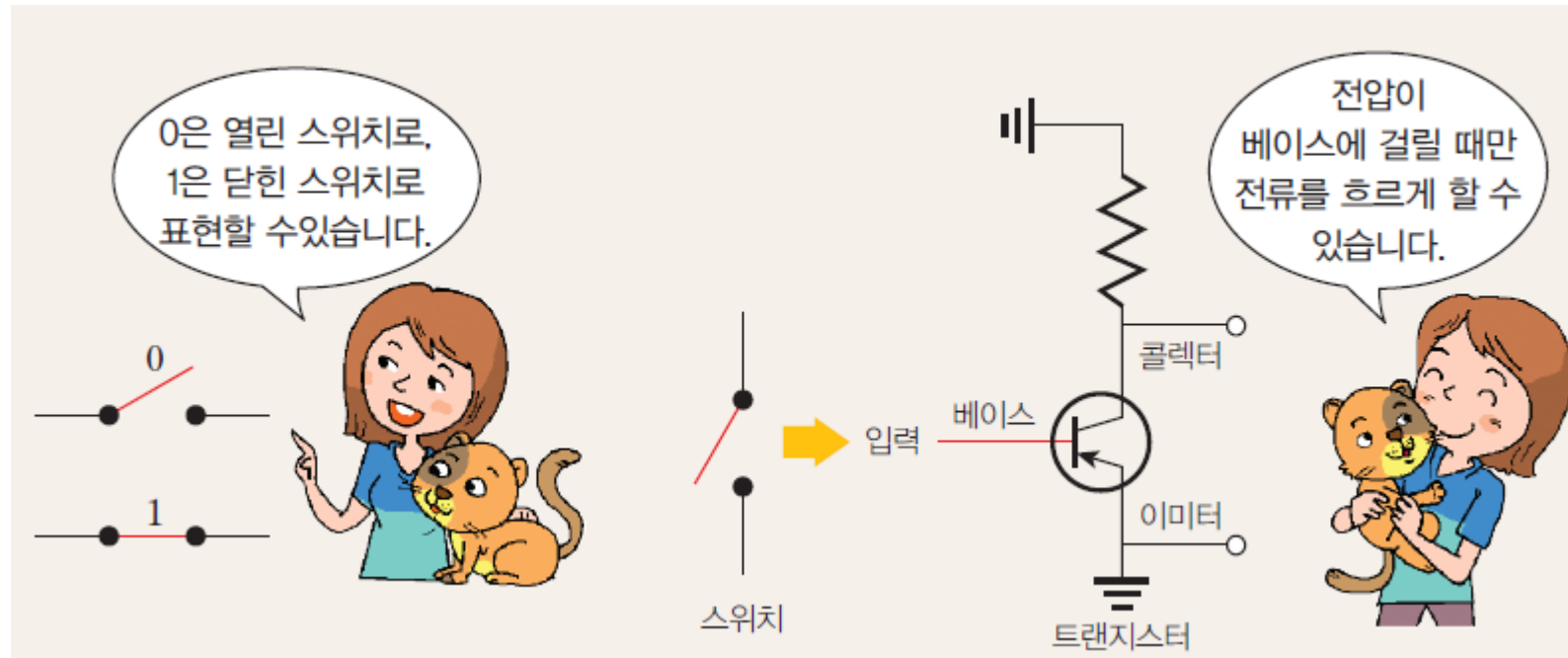


그림 1.4 컴파일러는 프로그램을 기계어로 변환한다.

왜 컴퓨터는 2진법을 사용할까?

- 2진수는 하드웨어로 구현하기가 쉽다. 컴퓨터 내부에서는 모든 것을 2진수 형태로 표현하여 처리한다.



중간점검



중간점검

1. 컴퓨터가 바로 이해할 수 있는 언어는 _____ 이다.
2. 컴파일러가 하는 일은 무엇인가?
3. 컴퓨터가 내부적으로 사용하는 진법은 _____ 진법이다.



C언어

- 1970년대 초 AT&T의 Dennis Ritchie 에 의하여 개발
- B언어->C언어
- UNIX 운영 체제 개발에 필요해서 만들어짐
- 처음부터 전문가용 언어로 출발



Ken Thomson과
Dennis Ritchie가
클린턴
대통령으로부터
National Medal of
Technology 상을
받는 장면

C언어의 특징



그림 1.6 C언어의 특징

C언어의 특징

- 간결하다.
- 효율적이다.
(다른 언어로 작성된 프로그램보다 크기가 작으며 실행 속도가 빠르고 메모리를 효과적으로 사용함)
- C 언어는 하드웨어를 직접 제어하는 하는 저수준의 프로그래밍도 가능하고 고수준의 프로그래밍도 가능하다.
(실제로 TV, 핸드폰, 세탁기 등의 여러 가지 전자 기기 안에 들어가는 임베디드(내장) 프로그램은 대부분 C언어로 개발됨)
- C언어는 이식성(portability)이 뛰어나다.
(한번 작성된 프로그램을 다른 종류의 CPU로 만들어진 하드웨어로 쉽게 이식할 수 있음 - 많은 종류의 CPU에 대하여 C컴파일러가 개발되어 있기 때문)
- 초보자가 배우기가 어렵다.

중간점검



중간점검

1. 이번 절에서 새롭게 등장한 용어들의 의미를 인터넷을 이용하여서 찾아보자.

"어셈블리 언어", "모듈", "임베디드 프로그램", "소프트웨어 공학"



알고리즘

Q) 오븐의 사용법만 배우고 음식 재료만 있으면 누구나 요리가 가능한가?

A) 요리법을 알아야 한다.



그림 1.7 오븐(컴퓨터)이 준비되고 오븐을 작동하는 방법(프로그래밍 언어)을 안다고 하더라도 요리를 하는 절차(알고리즘)를 모르면 요리(프로그램)를 만들 수 없다.

알고리즘

- 문제를 풀기 위하여 컴퓨터가 수행하여야 할 단계적인 절차를 알고리즘(algorithm)이라고 한다.
- (예) 전화번호부에서 친구인 "박철수"의 전화번호를 찾는 문제



빵을 만드는 알고리즘

- ① 빈 그릇을 준비한다.
- ② 이스트를 밀가루, 우유에 넣고 저어준다.
- ③ 버터, 설탕, 계란을 추가로 넣고 섞는다.
- ④ 따뜻한 곳에 놓아두어 발효시킨다
- ⑤ 170~180도의 오븐에서 굽는다

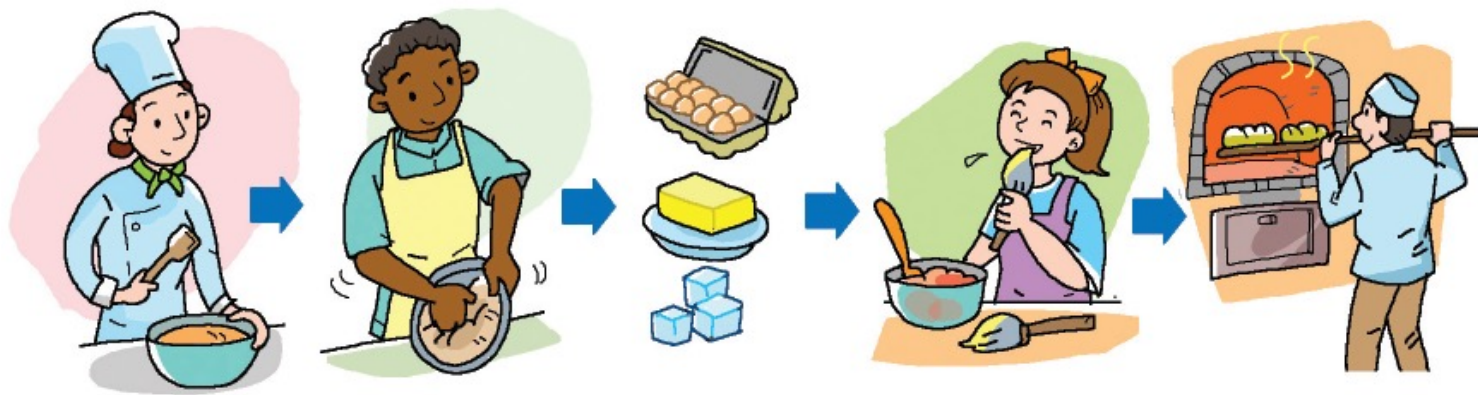


그림 1.8 알고리즘은 요리법과 같다.

알고리즘의 기술

- 순서도(flow chart): 프로그램에서의 논리 순서 또는 작업 순서를 그림으로 표현하는 방법



수행의 시작/종료



처리



판단



연결자(흐름도가 너무 커서 한페이지에 나타낼수 없을때 어디로 가라는 표시에 사용)



입출력

순서도의 예

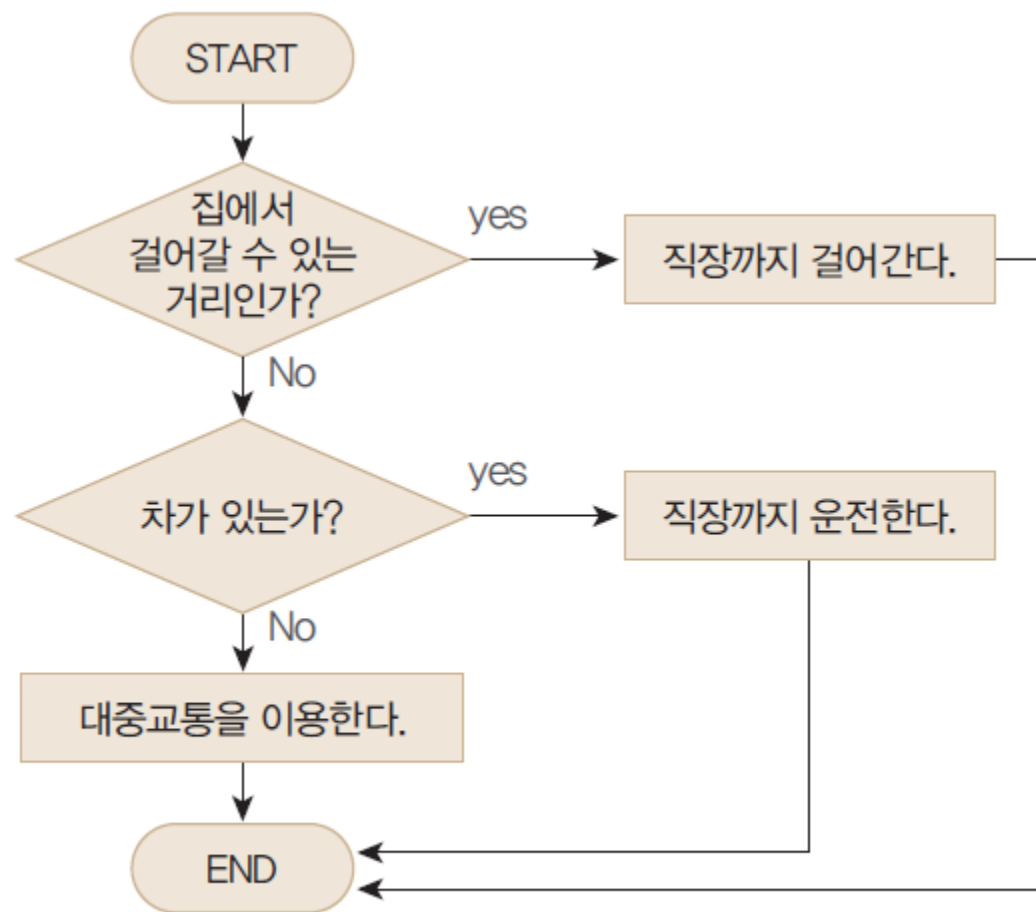


그림 1.9 순서도의 예

중간 점검



중간점검

1. 친구에게 전화를 거는 알고리즘을 순서도로 만들어보라.
2. 세탁기로 세탁하는 알고리즘을 순서도로 만들어보라.



프로그램 개발 과정

- ❶ 텍스트 에디터로 C언어 프로그램을 작성하여 파일로 저장한다.

텍스트 에디터로 작성된 프로그램을 소스 프로그램이라고 하고 이것을 확장자가 ".c"인 파일로 저장한 것을 소스 파일(source file)이라고 한다.



hello.c

- ❷ 소스 파일을 컴파일한다.

컴파일러(compiler)는 소스 파일을 분석하여 컴퓨터에서 실행이 가능하도록 기계어로 변환한다. 컴파일러는 소스 파일의 문장을 분석하여 문법에 맞도록 작성되었는지를 체크한다. 만약 오류가 발견되면 사용자에게 오류를 통보하고 프로그래머는 소스 작성 단계로 되돌아가서 소스 파일을 수정하여야 한다.

```
#include <stdio.h>
int main()
{
    printf("Hello")
    ;
    return 0;
}
```

소스 코드



```
100010101010101
000100101010111
011111100110000
001011001101010
010111011100011
011111001111000
00011001110101
010010010101000
```

실행 가능한 코드



hello.exe

- ❸ 프로그램을 실행한다.

컴파일이 성공적으로 수행되면 실행 가능한 파일이 만들어진다. 예를 들어서 소스 파일이 hello.c였다면 hello.exe 파일이 생성된다. 이 실행 파일을 가리키는 아이콘을 더블클릭하거나 통합 개발 환경 안에서 실행 메뉴를 선택하면 프로그램이 실행된다.

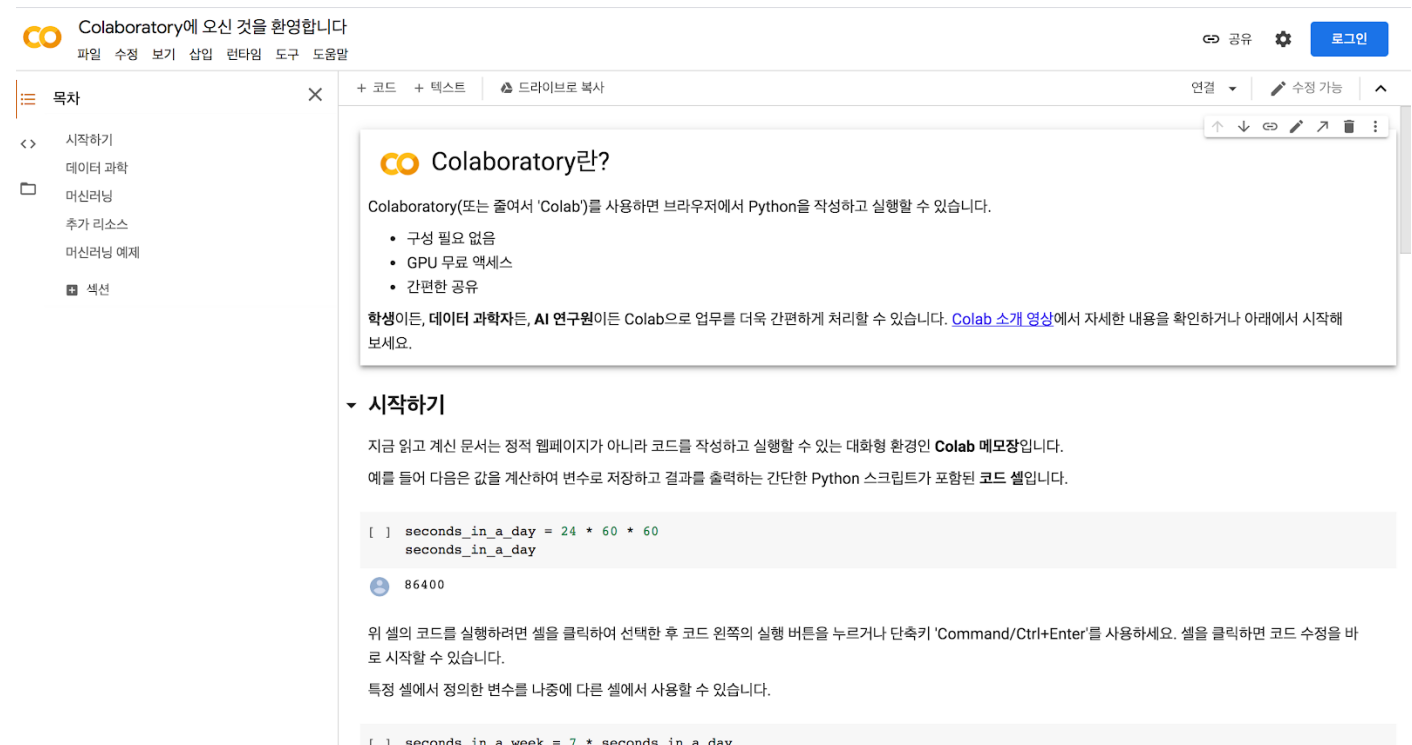


cloud 기반 c언어 실행 – google colab

- chrome 브라우저 다운로드 (기존 사용자는 최신버전 업데이트 권장)
- colab 접속하기

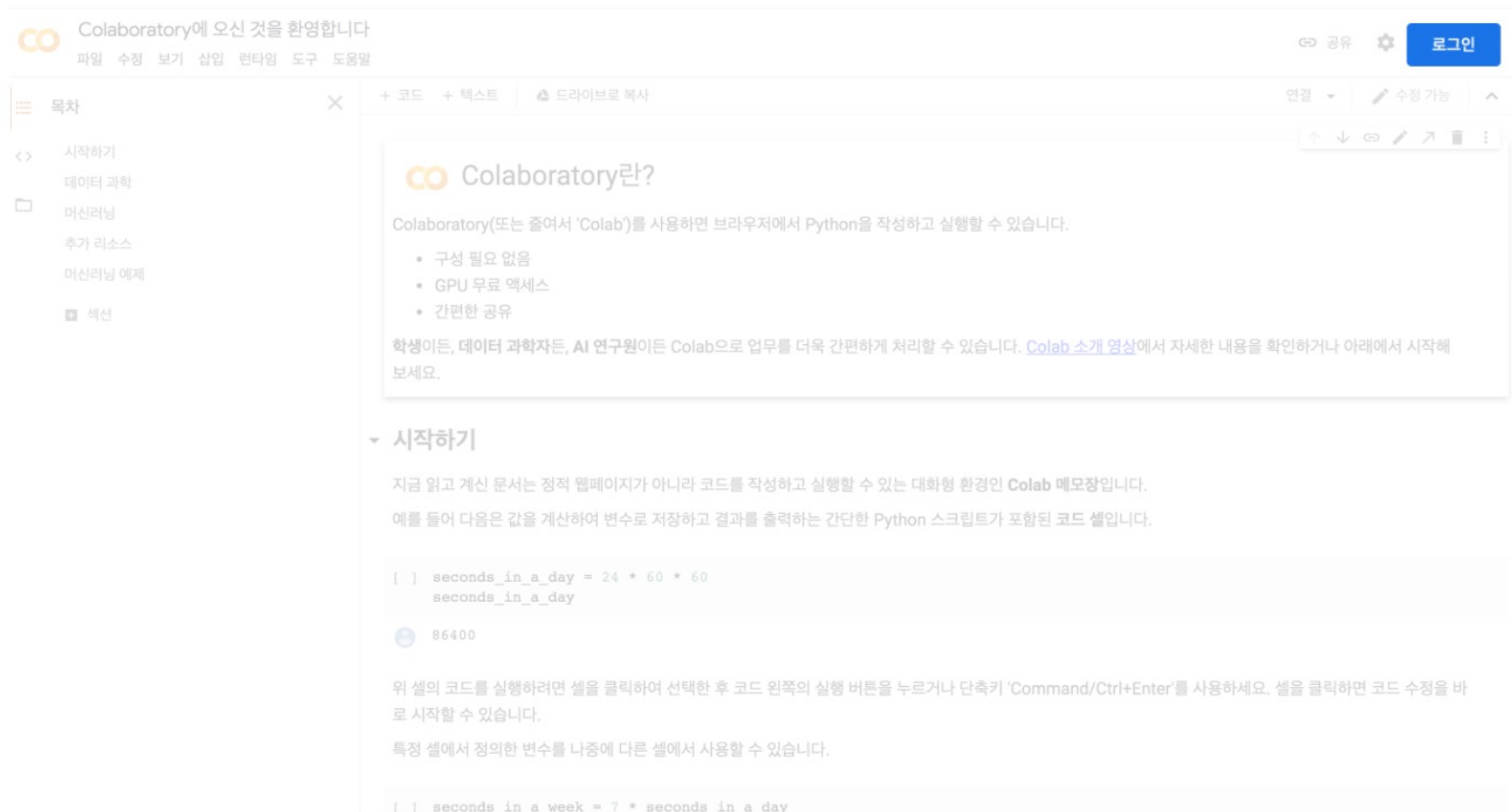
아래 주소로 접속합니다.

<https://colab.research.google.com/notebooks/intro.ipynb>



로그인 하기

구글 계정으로 로그인합니다.



The image shows the Colaboratory web interface. At the top, there's a header with the Colaboratory logo and the text "Colaboratory에 오신 것을 환영합니다". Below this, there's a navigation bar with links like "파일", "수정", "보기", "실행", "런타임", "도구", and "도움말". On the right side of the header, there are icons for "공유" (Share) and "로그인" (Login). The main content area is divided into two sections. The top section is a tutorial titled "Colaboratory란?" (What is Colaboratory?). It explains that Colaboratory (or JupyterLab) allows users to write and execute Python code in a browser. It lists three benefits: "구성 필요 없음" (No setup required), "GPU 무료 액세스" (Free GPU access), and "간편한 공유" (Easy sharing). It also mentions that students, data scientists, and AI researchers can use Colab to work more conveniently. The bottom section is titled "시작하기" (Getting started). It explains that the current screen is a JupyterLab environment. It provides an example of a Python script that calculates the number of seconds in a day. The script is shown in a code editor, and the output is displayed below it. The output is "86400". The interface also includes a sidebar on the left with a "목차" (Table of contents) and a "시작하기" (Getting started) section.

Colaboratory에 오신 것을 환영합니다

파일 수정 보기 실행 런타임 도구 도움말

로그인

공유

Colaboratory란?

Colaboratory(또는 줄여서 'Colab')를 사용하면 브라우저에서 Python을 작성하고 실행할 수 있습니다.

- 구성 필요 없음
- GPU 무료 액세스
- 간편한 공유

학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 내용을 확인하거나 아래에서 시작해 보세요.

시작하기

지금 읽고 계신 문서는 정적 웹페이지가 아니라 코드를 작성하고 실행할 수 있는 대화형 환경인 Colab 메모장입니다.

예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 코드 셀입니다.

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

위 셀의 코드를 실행하려면 셀을 클릭하여 선택한 후 코드 왼쪽의 실행 버튼을 누르거나 단축키 'Command/Ctrl+Enter'를 사용하세요. 셀을 클릭하면 코드 수정을 바로 시작할 수 있습니다.

특정 셀에서 정의한 변수를 나중에 다른 셀에서 사용할 수 있습니다.

```
[ ] seconds in a week = 7 * seconds in a day
```

새 노트 열기

새 노트를 준비합니다.

 Colaboratory에 오신 것을 환영합니다

파일 수정 보기 삽입 런타임 도구 도움말 변경사항을 저장할 수 없음

새 노트
노트 열기 ⌘/Ctrl+O
노트 업로드
노트 이름 변경
휴지통으로 이동
드라이브에 사본 저장
GitHub Gist로 사본 저장
GitHub에 사본 저장
저장 ⌘/Ctrl+S
버전 저장 및 고정 ⌘/Ctrl+M S
업데이트 기록
.ipynb 다운로드
.py 다운로드
드라이브 미리보기 업데이트
인쇄 ⌘/Ctrl+P

+ 코드 + 텍스트

드라이브로 복사

 Colaboratory란?

Colaboratory(또는 줄여서 'Colab')를 사용하면 브라우저에서 Python을 작성하고 실행할 수 있습니다.

- 구성 필요 없음
- GPU 무료 액세스
- 간편한 공유


학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 보세요.

시작하기

지금 읽고 계신 문서는 정적 웹페이지가 아니라 코드를 작성하고 실행할 수 있는 대화형 환경인 **Colab 메모장**입니다. 예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 코드 셀입니다.


```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
print ("hello world")
```

hello world

 Untitled1.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말 오후 1:54에 마지막으로 저장됨

+ 코드 + 텍스트



셀의 개념 이해하기

구글 colab은 코드를 작성하고 실행할 수 있는 대화형 환경을 가지고 있습니다. 해당 환경은 기본적으로 주피터 노트북과 동일합니다. 사용자는 단순히 코드를 작성 실행하는 것을 넘어 텍스트 편집, 시각화, 구글 드라이브와 동기화, 공유 등 다양한 작업을 수행할 수 있습니다.

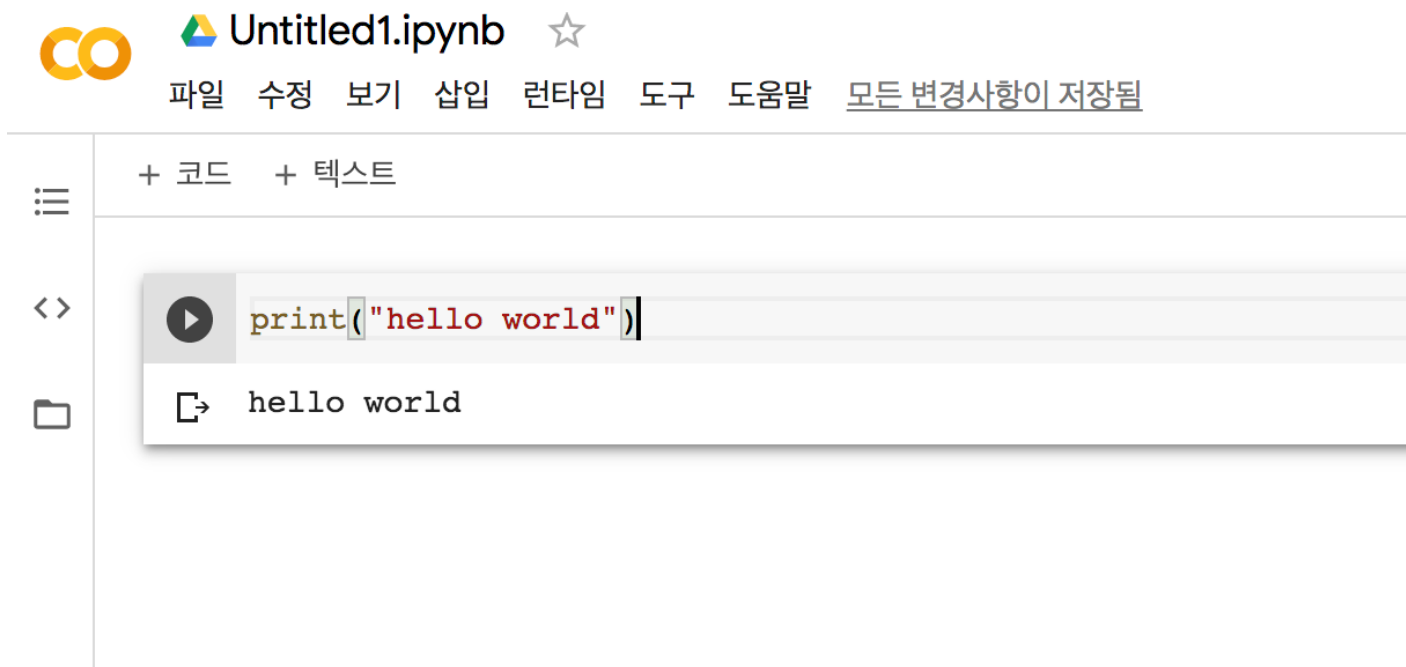
주피터 노트북에 대한 이해는 다음 링크를 참조하세요

<https://jupyter.org/>

코드 셀

코드 셀에서는 (아무런 추가 설정 없이)
python코드를 실행 시킬 수 있습니다.

우리의 c언어는 컴파일러가 필요해요!
(뒤에 방법 나옴)



코드를 작성후 시작 버튼을 누르거나 Command/Ctrl/Shift +Enter 를 누르면 실행됩니다.

텍스트 셀

텍스트 셀은 md(mark down)형식을 이용합니다. md를 이용하면 간단하고 쉽게 텍스트를 표시할 수 있습니다.

▼ This is a heading

This is some text

- this
- is
- a
- list

$$\int_0^{\infty} x^2 dx$$

뿐만 아니라 제목, 문단, 리스트 , 수식 등도 자유롭게 표현할 수 있습니다.

Python으로 직접 코드 작성해보기

셀에 대한 기본적인 이해를 마쳤으니 직접 코드를 작성하고 실행해 봅시다.

새로운 코드 셀을 만들고 `print("Hello World! My name is ○○○")`을 실행시켜 봅시다.

결과 화면은 다음과 같이 나와야합니다.

```
[5] print('Hello World!')
```

```
↳ Hello World!
```

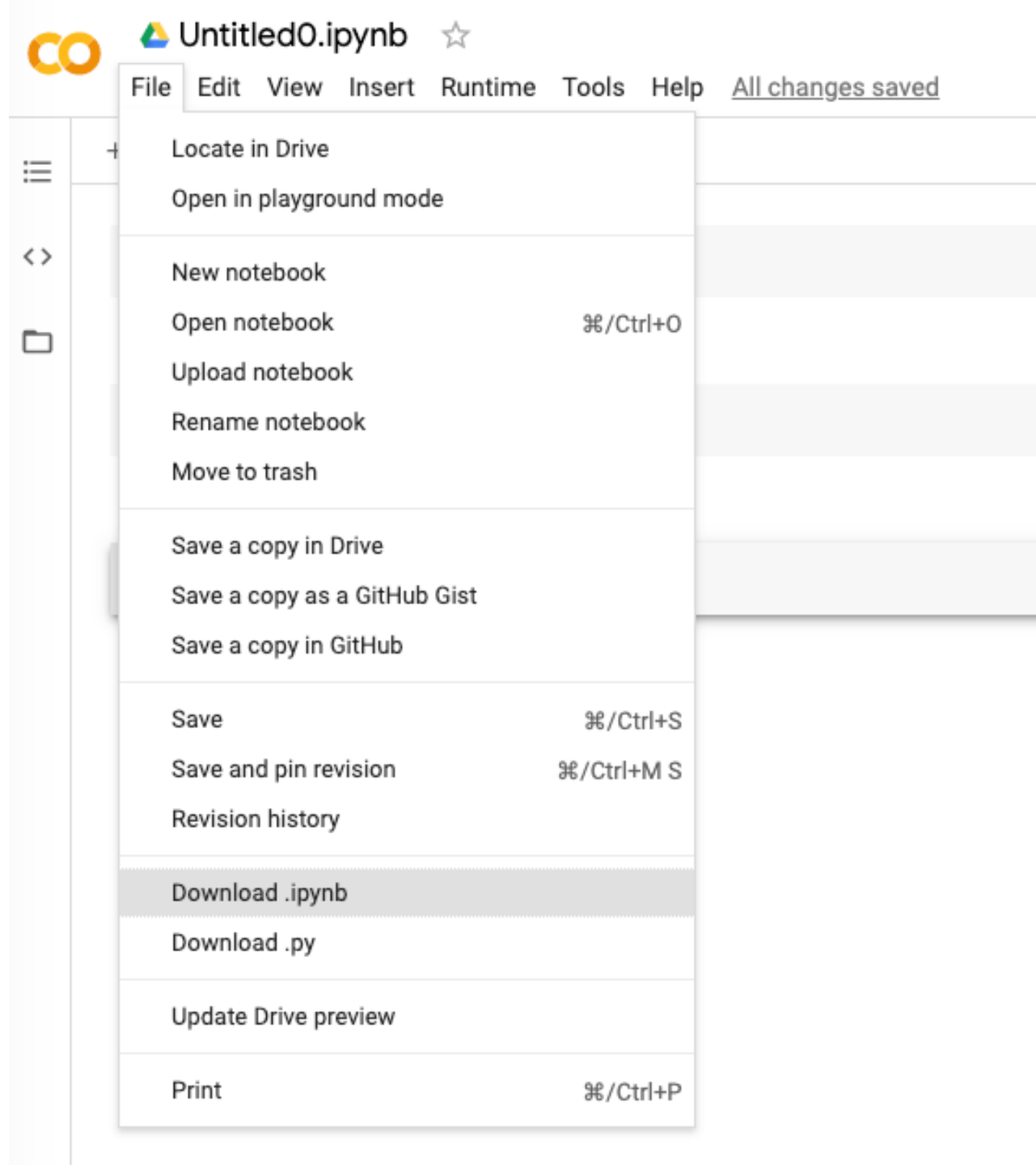
```
[6] print('My name is Minhae Kwon')
```

```
↳ My name is Minhae Kwon
```



코드 저장 방법 (과제 제출시)

과제 제출시에는 .ipynb 파일로 제출합니다.
(Jupyter notebook 타입 확장자)



c언어로 Hello World 출력해보기

```
/* 첫번째 프로그램*/  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!");  
    return 0;  
}
```





+ 코드 + 텍스트

```
[ ] %%writefile hello.c
/* %%writefile filename" command is used to write a file to current directory in colab. */

#include <stdio.h>

int main(void)
{
    printf("Hello World!");

    return 0;
}
```

Overwriting hello.c

```
[ ] ls
```

hello.c output* sample_data/

```
[ ] %%shell

gcc hello.c -o output
./output
```

Hello World!

%%shell magic command invokes linux shell(bash,etc,) to run the entire shell as a shell script.*/*

소스 코드 입력시 주의 사항

- c에서는 대문자와 소문자를 구별한다. 따라서 대문자와 소문자를 정확하게 구별하여 입력하도록 하여야 한다. 즉 main과 Main은 서로 다르다.
- 주어진 소스 코드를 입력할 때 한 글자라도 틀리지 않게 철자에 주의하여야 한다. 하나의 기호만 틀려도 실행이 불가능하다. stdio.h라고 해야 할 것을 stdio,h로 입력하면 안된다. 컴퓨터는 믿을 수 없을 만큼 단순하다.
- 소스 코드에서의 각 명령어 문장은 세미콜론(;) 기호로 끝나야 한다.; 기호는 마침표 역할을 한다.
- 각 문장과 문장 사이에는 공백이 있어도 된다. 또한 문장은 들여 쓸 수 있다. 그러나 일단은 주어진 대로 입력하도록 하도록 하자.

통합 개발 환경

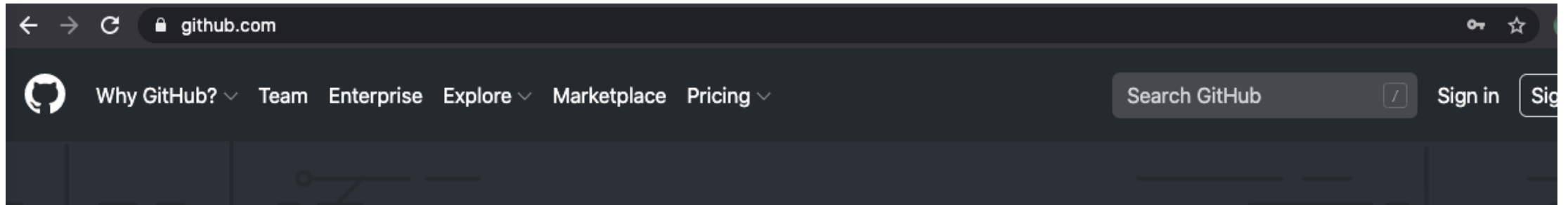
- 통합 개발 환경(IDE: integrated development environment):
 - 에디터 + 컴파일러 + 디버거



통합 개발 환경은 에디터, 컴파일러, 디버거를 하나로 합친 프로그램이다.

MS 비주얼 스튜디오(<http://www.visualstudio.com/ko>)

Github 추천



- 앞으로 4년동안 공부하는 코드들을 github에 관리하기를 추천.
- 과목별/ 프로젝트별 repository를 만들기를 추천
- 본 교과목에서 제출하는 실습코드만 다 모아서 하나의 repository를 만들 수 있음.

To do list

- 교재 ch.1 정독하며 복습합니다.
- 개인적으로 연습문제를 풀어보세요.
- 실습 과제
 - 과제 제출기한은 항상 1주일 입니다
 - Colab에서 하나의 파일에 실습한 코드를 *lab0_ 학번_이름.ipynb* 파일로 제출합니다.