

2020 문화자치위원회 겨울계절 원데이클래스

# LEC 00. PYTHON 사용환경

---

박수빈  
[psbpsu@snu.ac.kr](mailto:psbpsu@snu.ac.kr)

```
.tmux.conf
54 # force a reload of the config file
55 unbind r
56 bind r source-file ~/.tmux.conf
57
58 set -g mouse on
59 set-option -s set-clipboard off
60 bind-key -T copy-mode-vi MouseDragEnd1Pane send-keys -X copy-pipe-and-cancel "xclip -se c -i"
61 bind -T root WheelUpPane if-shell -F -t = "#{{alternate_on}}" "send-keys -M" "select-pane -t =; copy-mode -e; send-keys -M"
62 bind -T root WheelDownPane if-shell -F -t = "#{{alternate_on}}" "send-keys -M" "select-pane -t =; send-keys -M"
63
64 # List of plugins
65 set -g @plugin 'tmux-plugins/tpm'
66 set -g @plugin 'tmux-plugins/tmux-sensible'
67 set -g @plugin 'christoomey/vim-tmux-navigator'
68 set -g @plugin 'tmux-plugins/tmux-yank'
69
70 set-option -g default-command "reattach-to-user-namespace -l $SHELL"
71 # Status bar
72 set -g status on
73 set -g set-titles-string '#{pane_current_command}'
74 set -g status-bg colour15
75 set -g message-command-fg "colour117"
76 set -g status-justify "left"
77 set -g status-left-length "100"
78 set -g pane-base-index 1
79 set -g base-index 1
80 set -g pane-active-border-fg "colour117"
81 set -g message-fg "colour59"
82 set -g message-bg "colour59"
83 set -g status-right-length "100"
84 set -g status-right-attr "none"
85 set -g message-fg "colour117"
86 set -g message-command-bg "colour59"
87 set -g status-attr "none"
88 set -g pane-border-fg "colour59"
89 set -g status-left-attr "none"
90 setw -g window-status-fg "colour231"
91 setw -g window-status-attr "none"
92 setw -g window-status-activity-bg "colour59"
93 setw -g window-status-activity-attr "none"
94 setw -g window-status-activity-fg "colour215"
```

어디 가서 아는 적 할 만한 단어 주워가고  
바쁘신 분은 윈도우는 5쪽부터, 맥은  
설치에 실패하신 분은 최후의 보루로 10쪽  
다 안되신다면 카톡

여기에서 아는 척 할 만한 단어 주워가고 싶으신 분은 다음 쪽부터 천천히,  
바쁘신 분은 원도우는 5쪽부터, 맥은 맥은 10쪽 보시면 됩니다!

설치에 실패하신 분은 최후의 보루로 16쪽을 시도해보시면 됩니다 😞

# 다 안되신다면 카톡 주세요



# 시작하기 전에!

# IDE란?



## 통합개발환경(Integrated Development Environment, IDE)!

### 왜 IDE를 사용할까?

기본적으로 프로그래밍을 이용해 컴퓨터로 작업을 수행하는 일은 크게 세 가지 단계로 이루어져 있습니다.

1. 명령을 작성한다. (흔히 말하는 소스 코드 Source Code를 작성하는 일이죠!)
2. 컴퓨터가 알아들을 수 있는 언어(기계 언어 Machine Language)로 변환한다.
3. 컴퓨터가 작성한 프로그램을 실행한다.

그런데 이 작업들을 따로따로 수행하려면 과정이 너무 복잡해집니다 😊

1번을 위한 창을 띄워서 프로그램을 작성하고, 2번 작업과 3번 작업을 또 다른 창에서 하려면... 😰

그런 귀찮음을 줄여준 것이 바로 IDE, 즉 통합 개발 환경인데요, IDE에서는 1~3의 과정을 따로따로 수행하지 않고

창의 일부분에서는 소스 코드를 수정하고, ▶👉 이런 버튼만 대충 누르면 2~3의 과정은 프로그램이 알아서 진행해주기 때문에 결과를 바로 확인할 수 있고, 그렇기 때문에 많은 프로그래머들이 애용합니다!

요약하자면, IDE를 설치하는게 이 슬라이드의 목표입니다 🚀

### IDE의 예시

운영 체제(Window, OS, Linux 등)마다, 그리고 사용하는 프로그래밍 언어마다 IDE가 다른데,

보편적으로 많이 사용되는 IDE들은 다음과 같습니다.

Windows - Visual Studio(C/C++, Python, Java, Java Script 등 거의 모든 언어)

Mac OS - Xcode(C/C++, Python, Java, Java Script, Swift 등), VS Code(Visual Studio와 유사)

이외에도 Jet Brain 사에서 만든 언어별 IDE : CLion(C/C++), IntelliJ(Java), PyCharm(Python) 등이 유명합니다!

하지만 저희는 초심자가 사용하기 가장 편한 **Jupyter Lab/Jupyter Notebook**이라는 프로그램을 사용할겁니다 🐍

# JUPYTER LAB/JUPYTER NOTEBOOK

JUPYTER  
NOTEBOOK

JUPYTER LAB

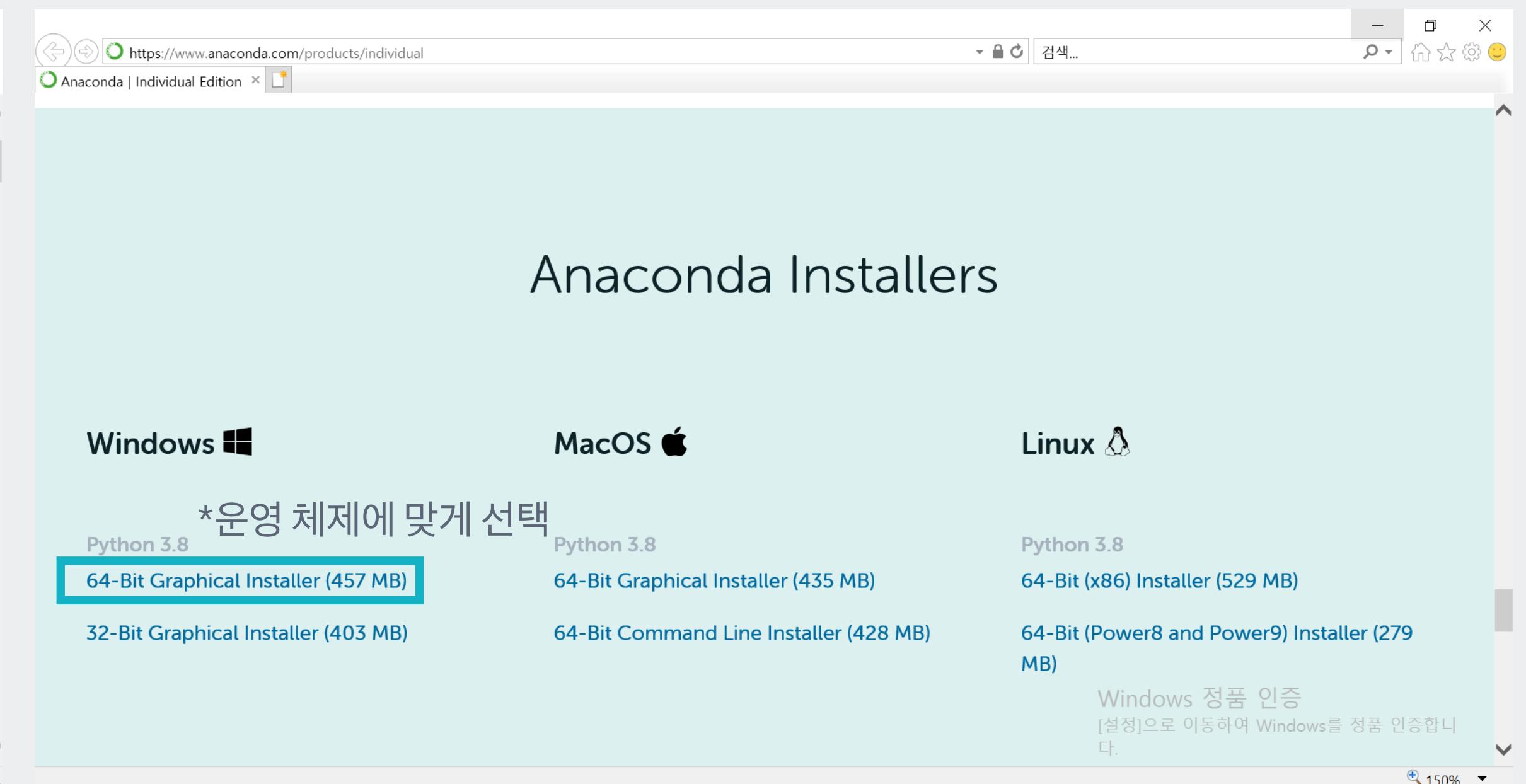
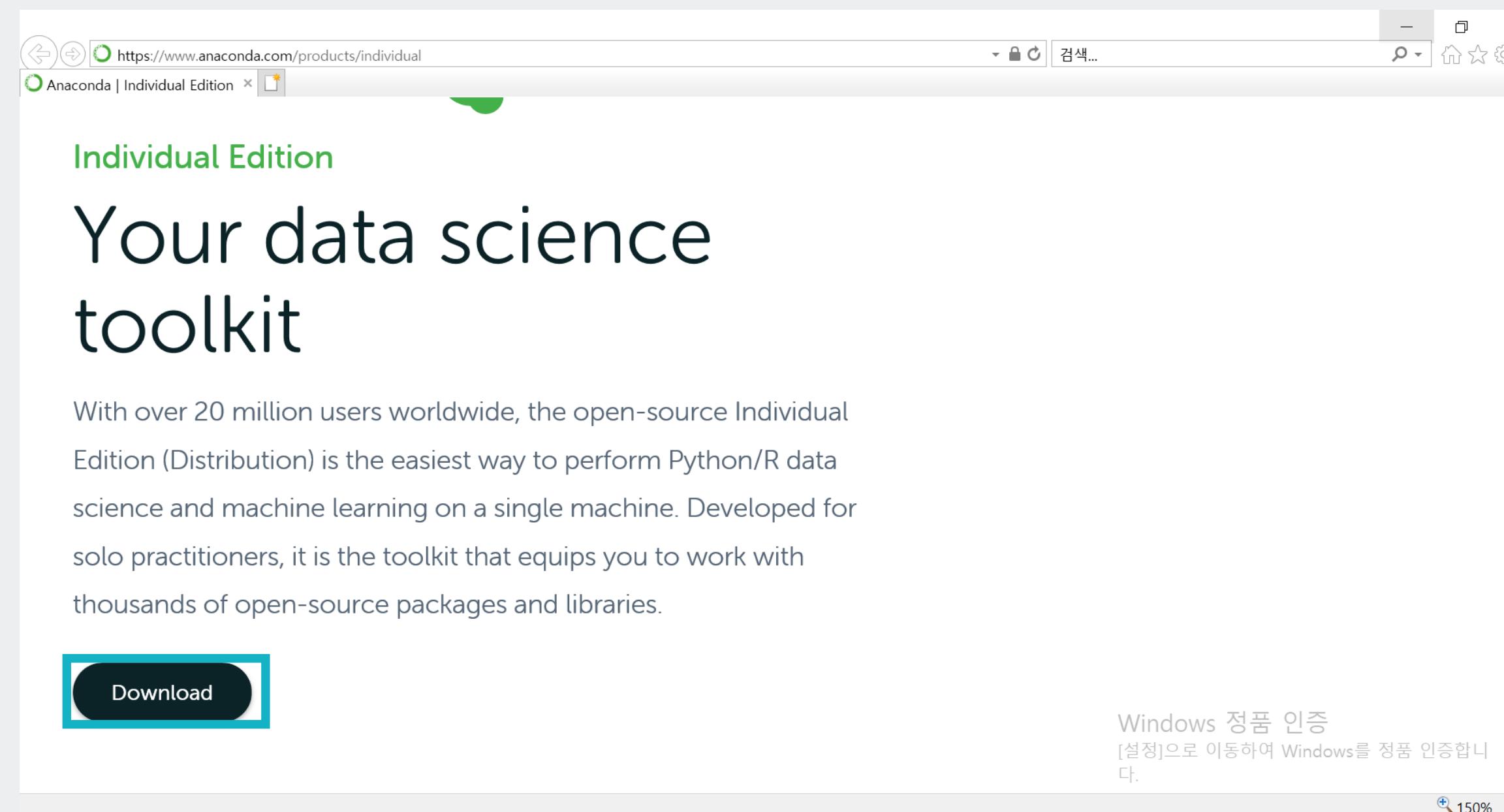
Jupyter Notebook은 Python 사용을 위한 웹 어플리케이션으로, 셀(Cell) 단위로 프로그램을 작성할 수 있다는 게 큰 장점인 프로그램입니다! 프로그램을 300줄을 한꺼번에 작성하고, 틀린 부분을 위에서부터 하나하나 읽으면서 고치는 대신 1번 셀 작성 - 오류 검토 - 2번 셀에 명령 추가와 같이 중간 중간 점검을 하면서 프로그래밍을 할 수 있기 때문에 초심자들에게 아~~주 적합한 프로그램이죠✨  
뿐만 아니라 코드 밑에 부가적인 셀을 추가해서 필기까지 한꺼번에 할 수 있기 때문에 더욱 좋습니다👍

Jupyter Lab은 Jupyter Notebook에 코딩에 용이한 기능들이 부가적으로 더해진 웹 어플리케이션입니다! Jupyter Notebook에서는 되지 않던 다중 창, 파일 디렉토리 확인 등이 가능하기 때문에 다른 코드를 참조하면서 코딩하기에 한결 편리합니다. 뿐만 아니라 Excel, Txt, Png 등 다양한 파일 형식을 Jupyter Lab을 통해 열어볼 수 있기 때문에 앞으로 파이썬을 내가 하는 일에 적극 활용하겠다! 라는 생각을 가지신 분들은 웬만해서는 Jupyter Lab을 설치하시는 것을 추천드립니다😊

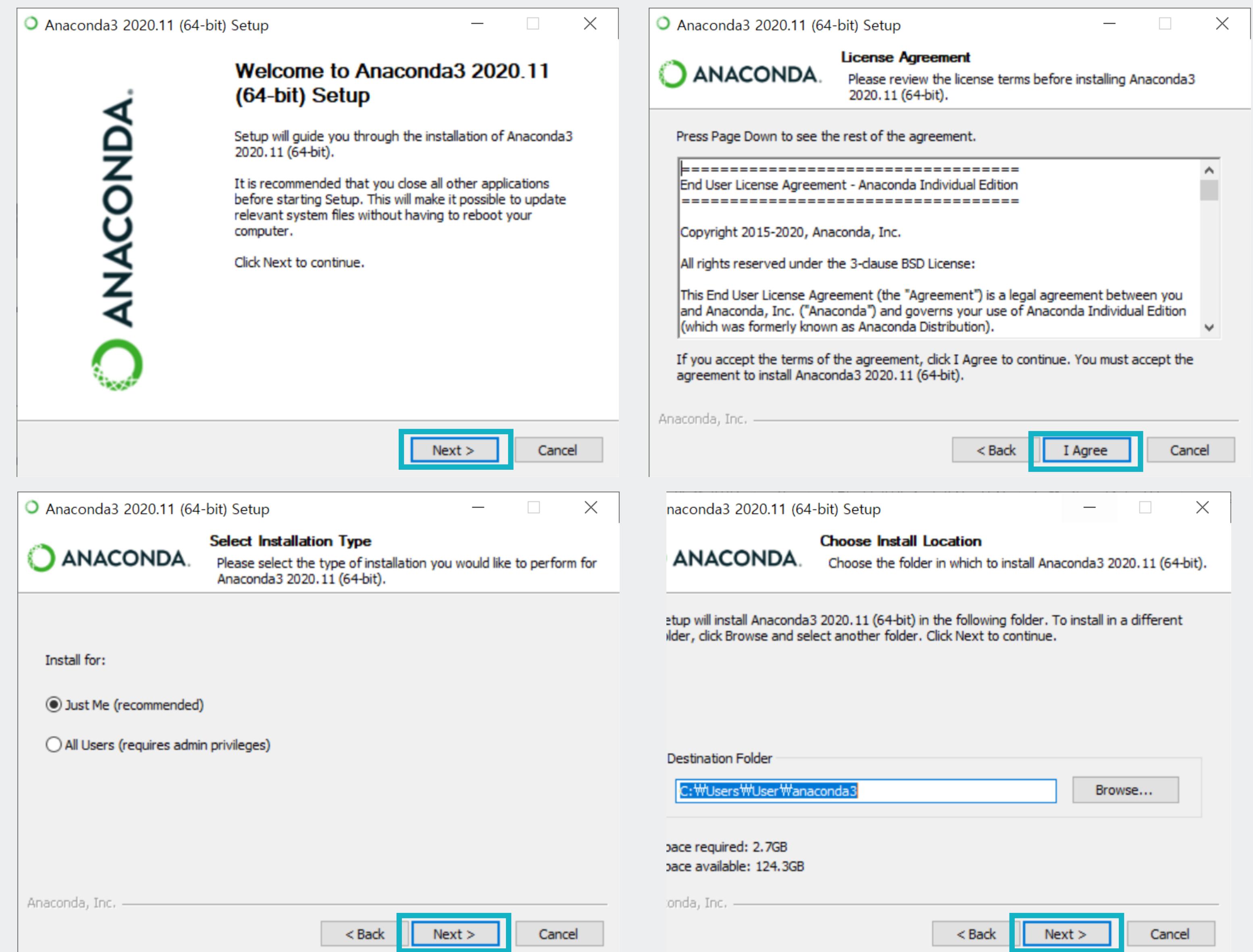


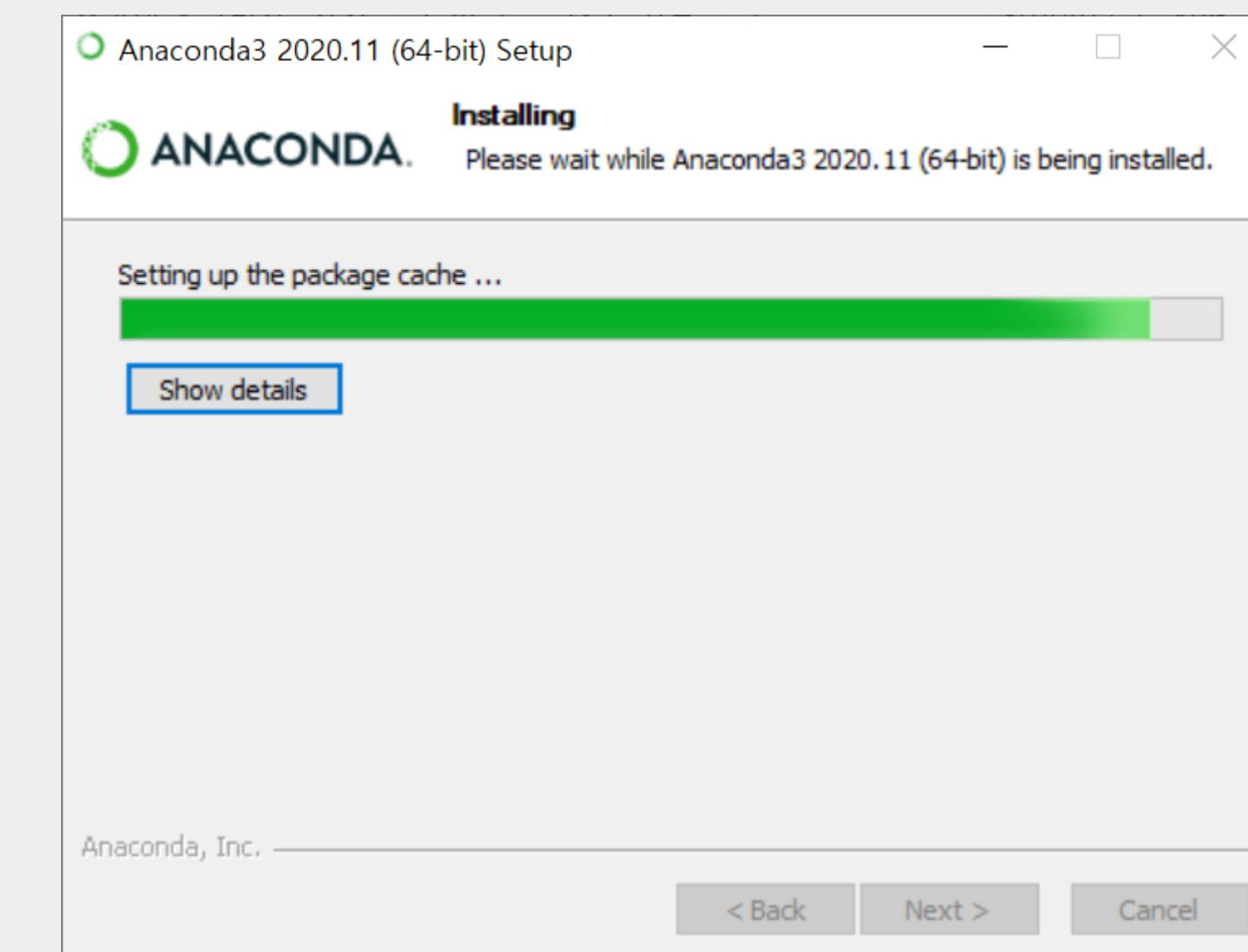
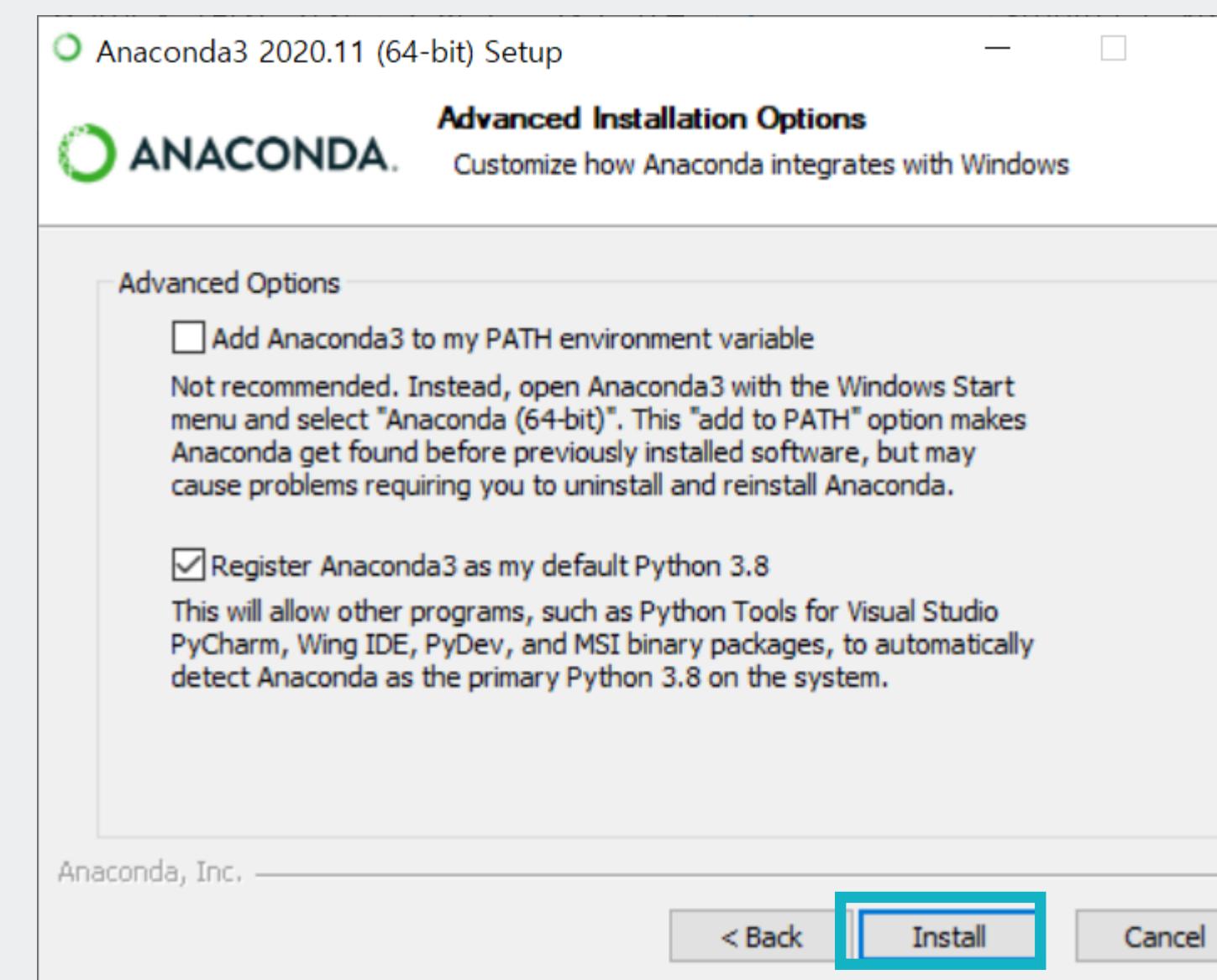
아직 뭐가 좋은지 잘 모르시겠다면, 수업을 하며 차차 알아나가도록 합시다~~

~~Jupyter가 Jupiter보다 의속해지는 그날까지 화이팅~~

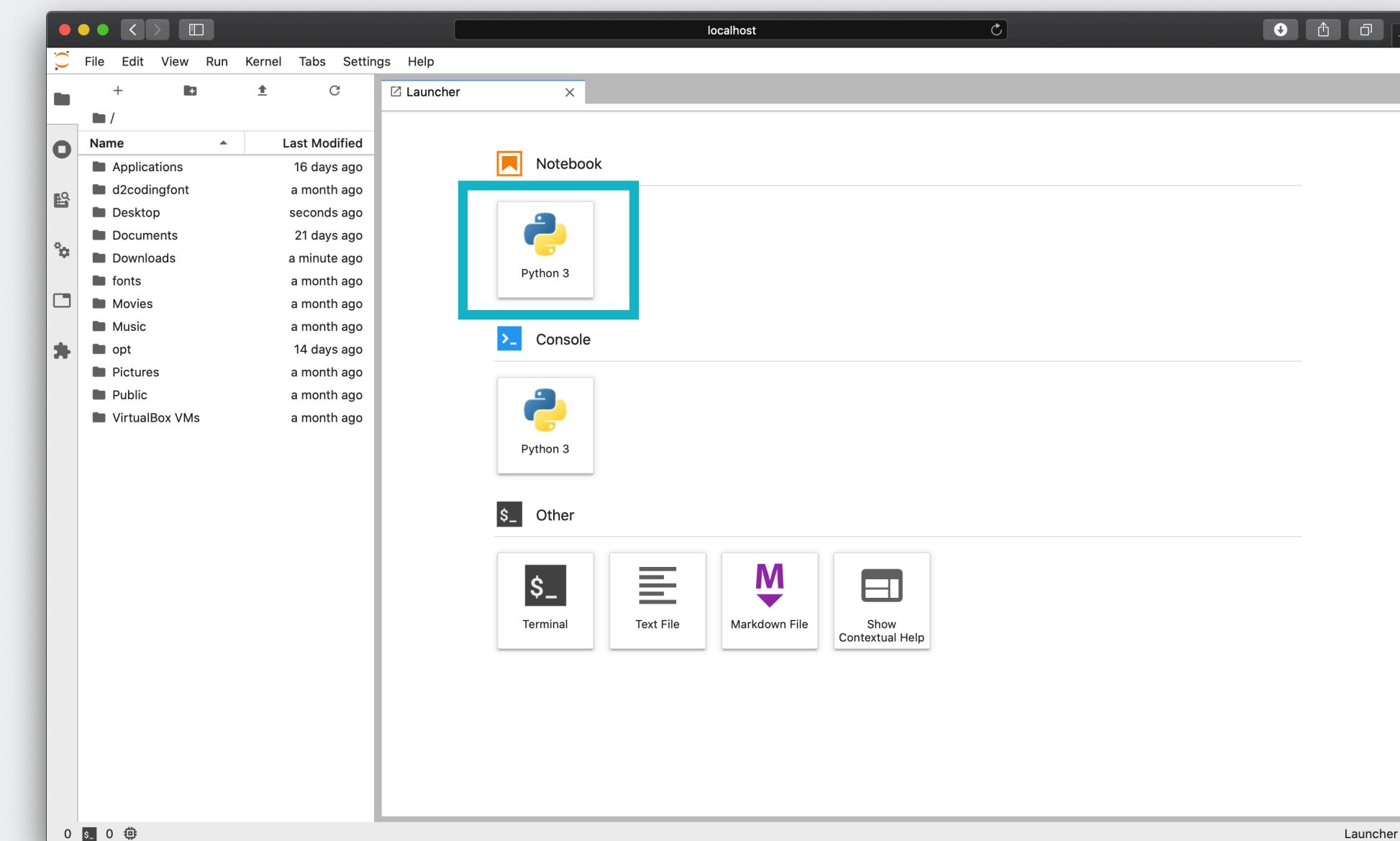
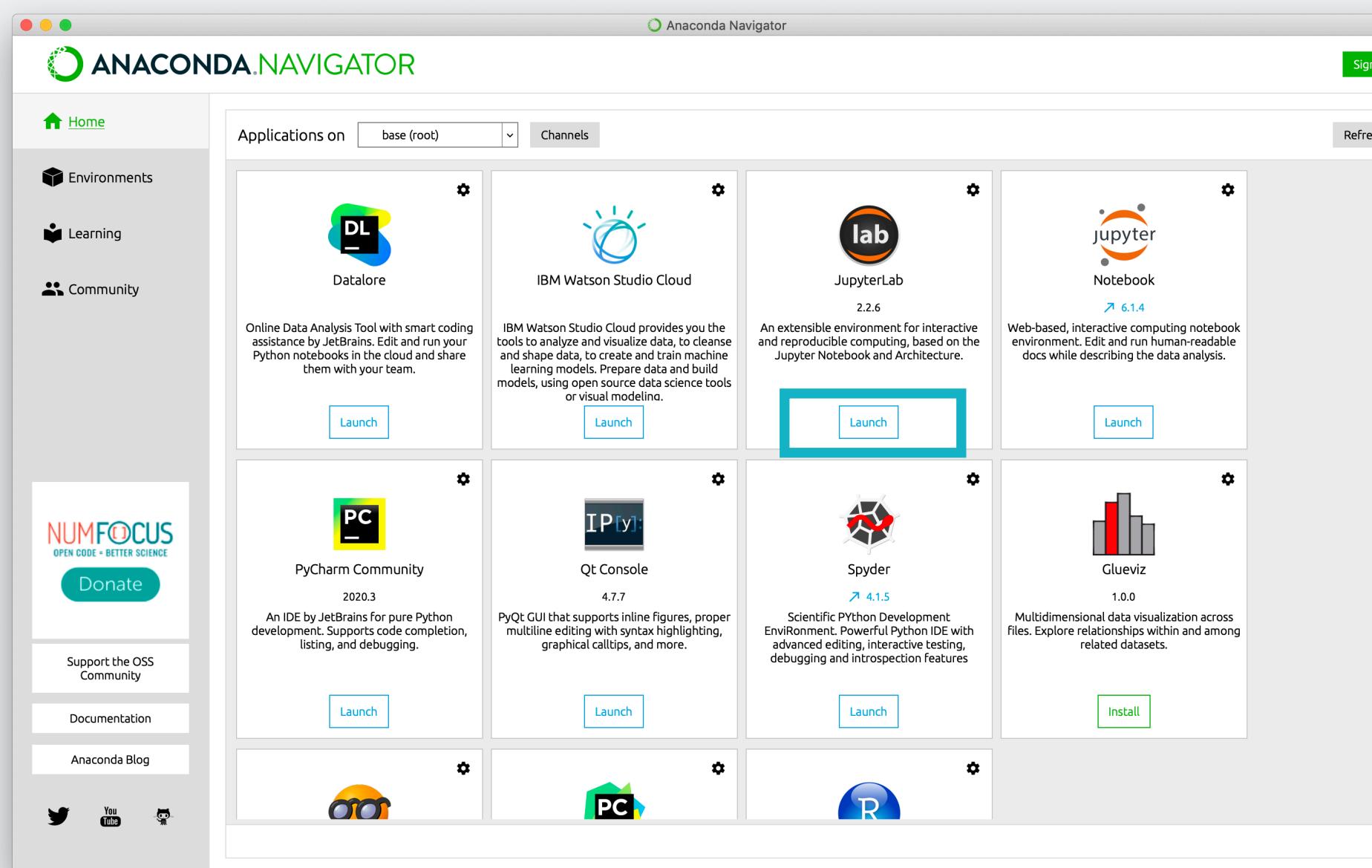


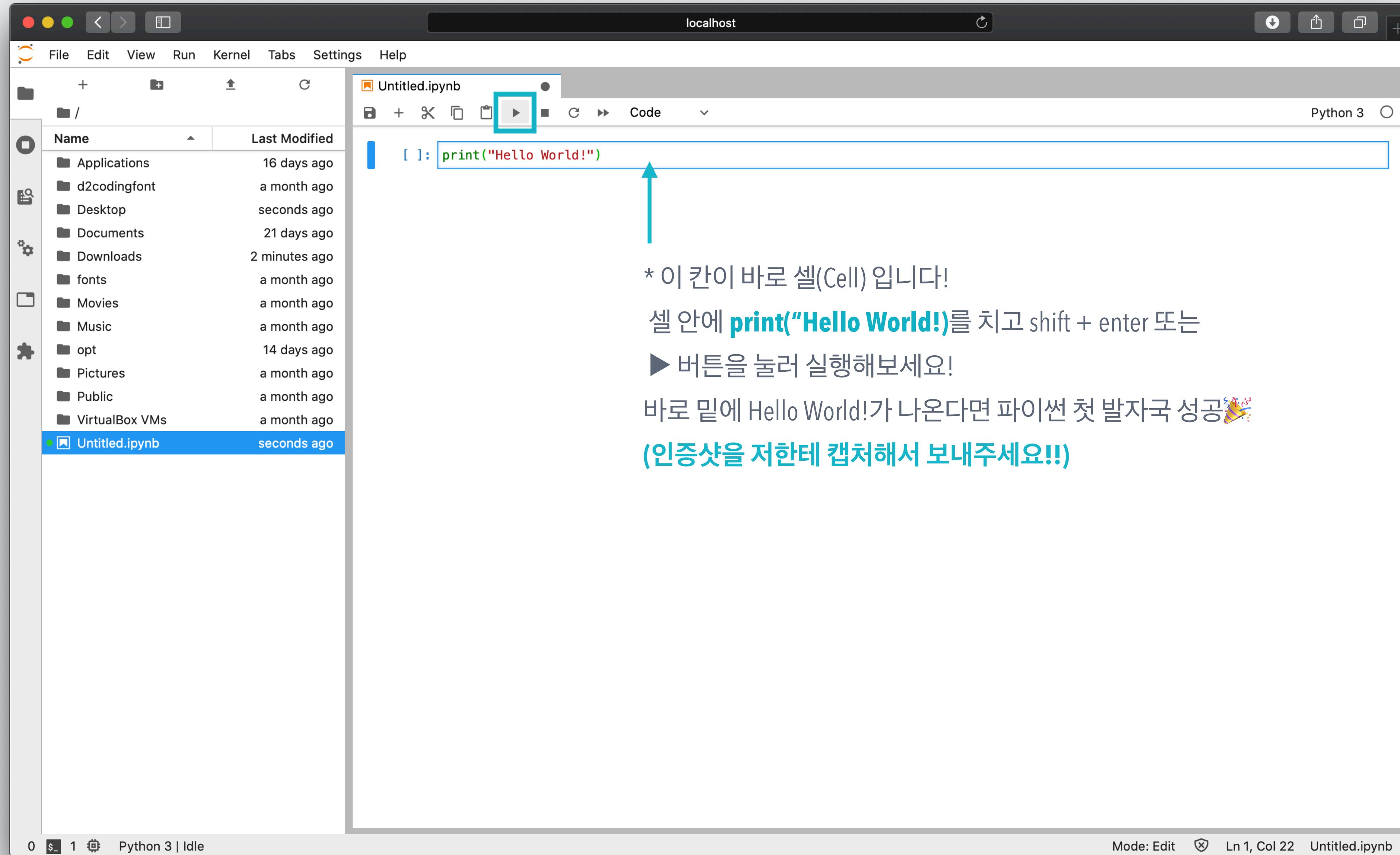
<https://www.anaconda.com/products/individual>





## 프로그램 Anaconda-Navigator(3) 실행



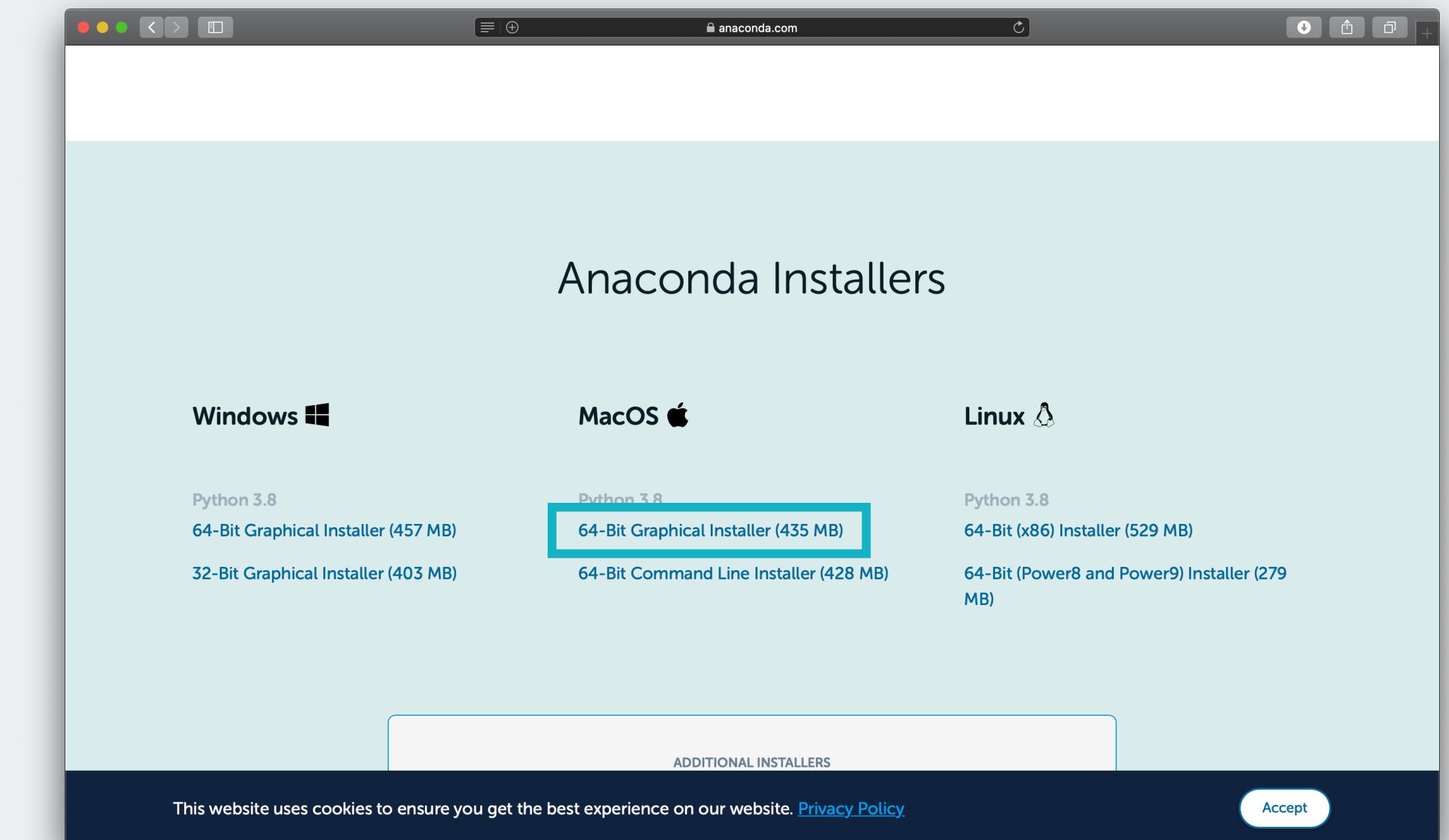
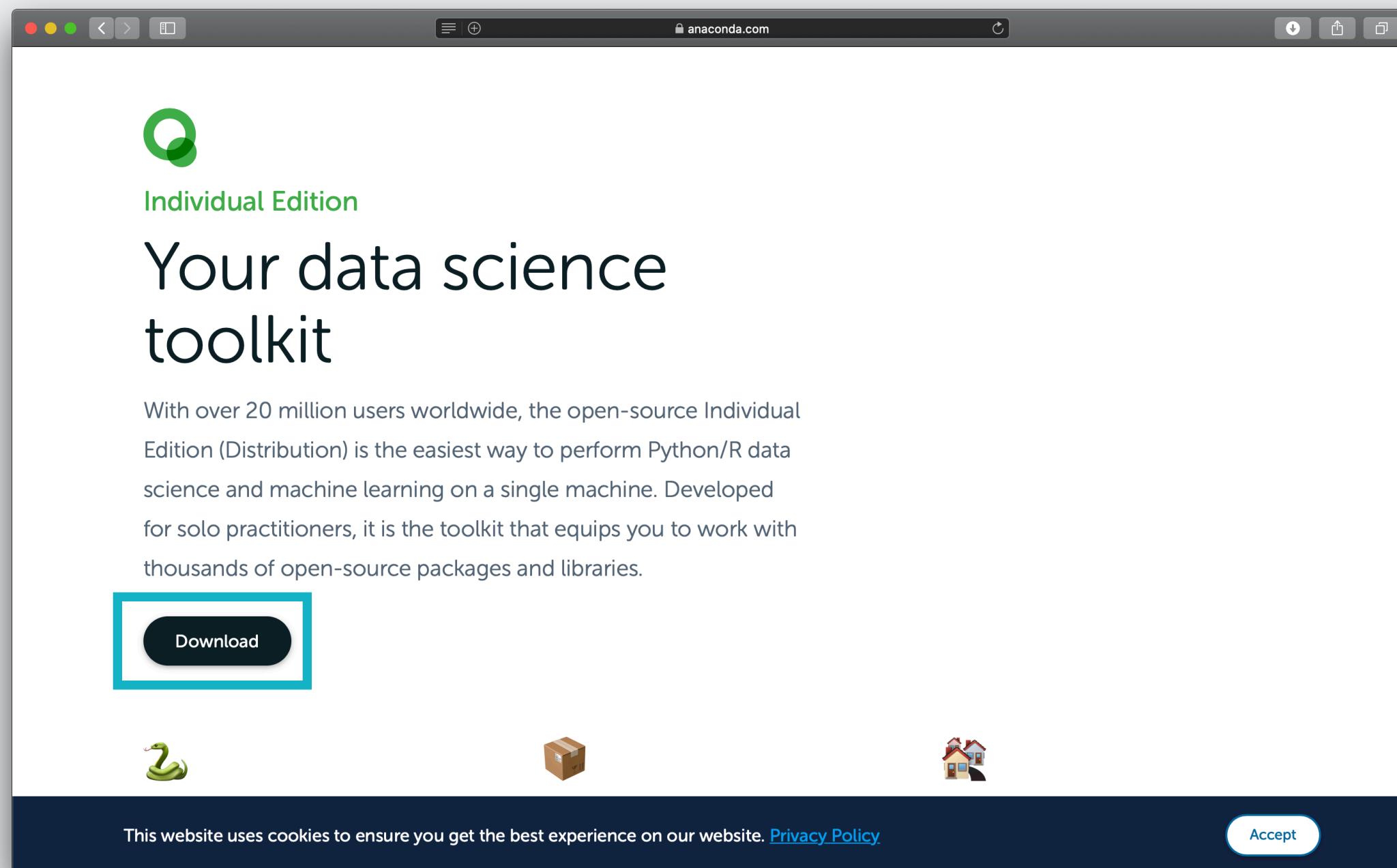


\* 이 칸이 바로 셀(Cell)입니다!

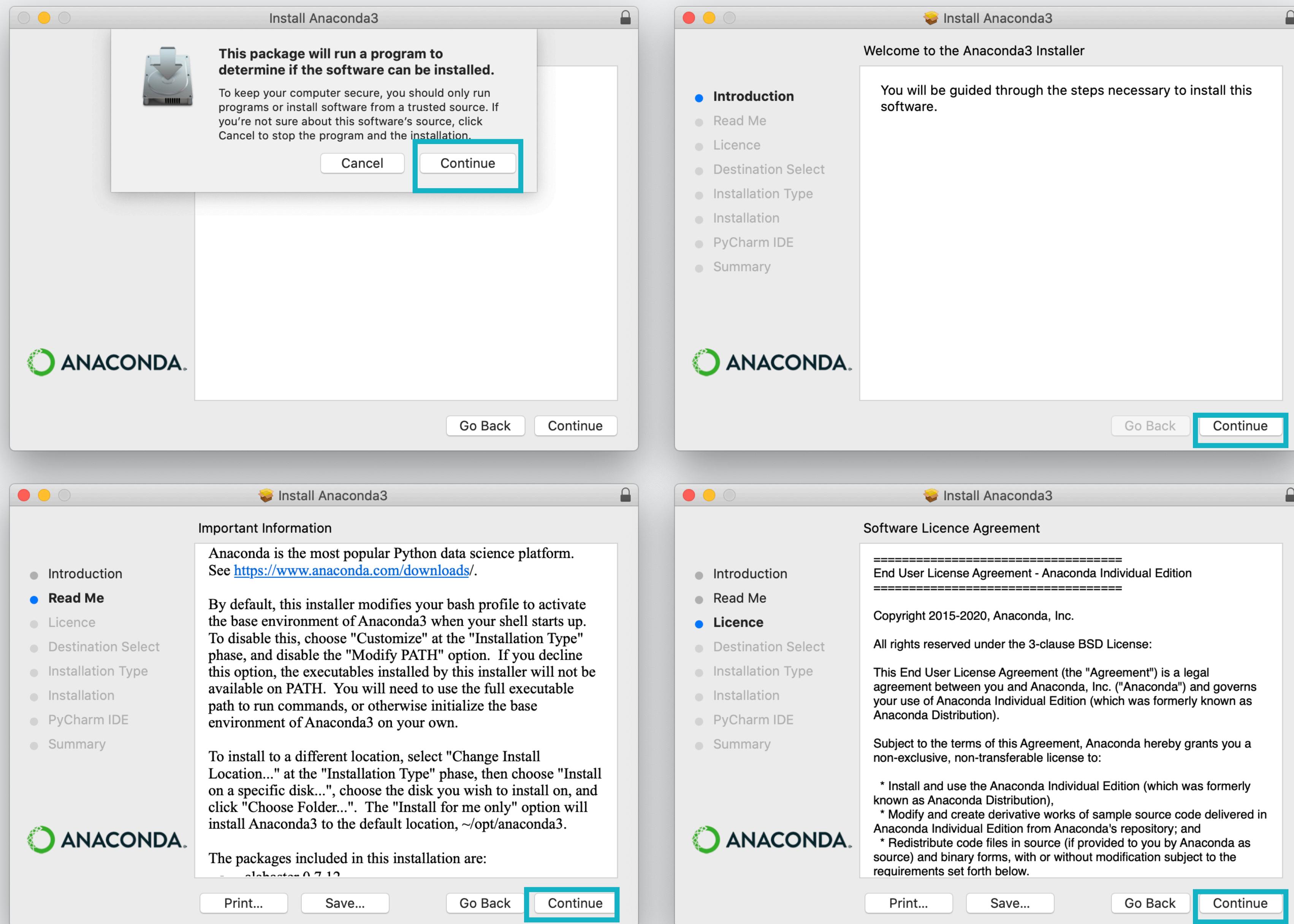
셀 안에 **print("Hello World!")**를 치고 shift + enter 또는 ► 버튼을 눌러 실행해보세요!

바로 밑에 Hello World!가 나온다면 파이썬 첫 발자국 성공🎉  
**(인증샷을 저한테 캡처해서 보내주세요!!)**

```
0 $ 1 Python 3 | Idle Mode: Edit  LN 1, Col 22 Untitled.ipynb
```



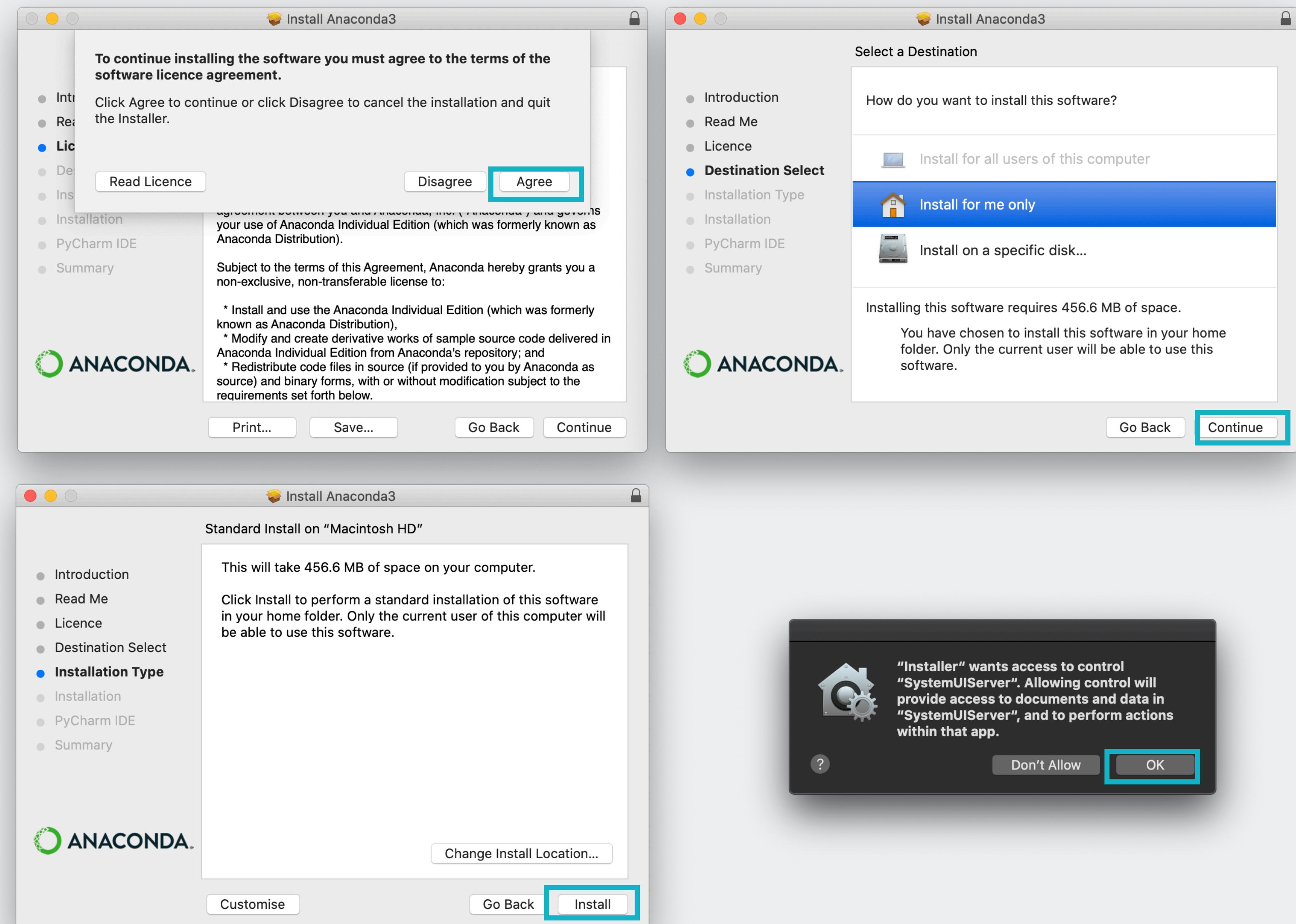
<https://www.anaconda.com/products/individual>



\* 일부 저와 다른 창이

\* 뜨신다면 변경하지 않고

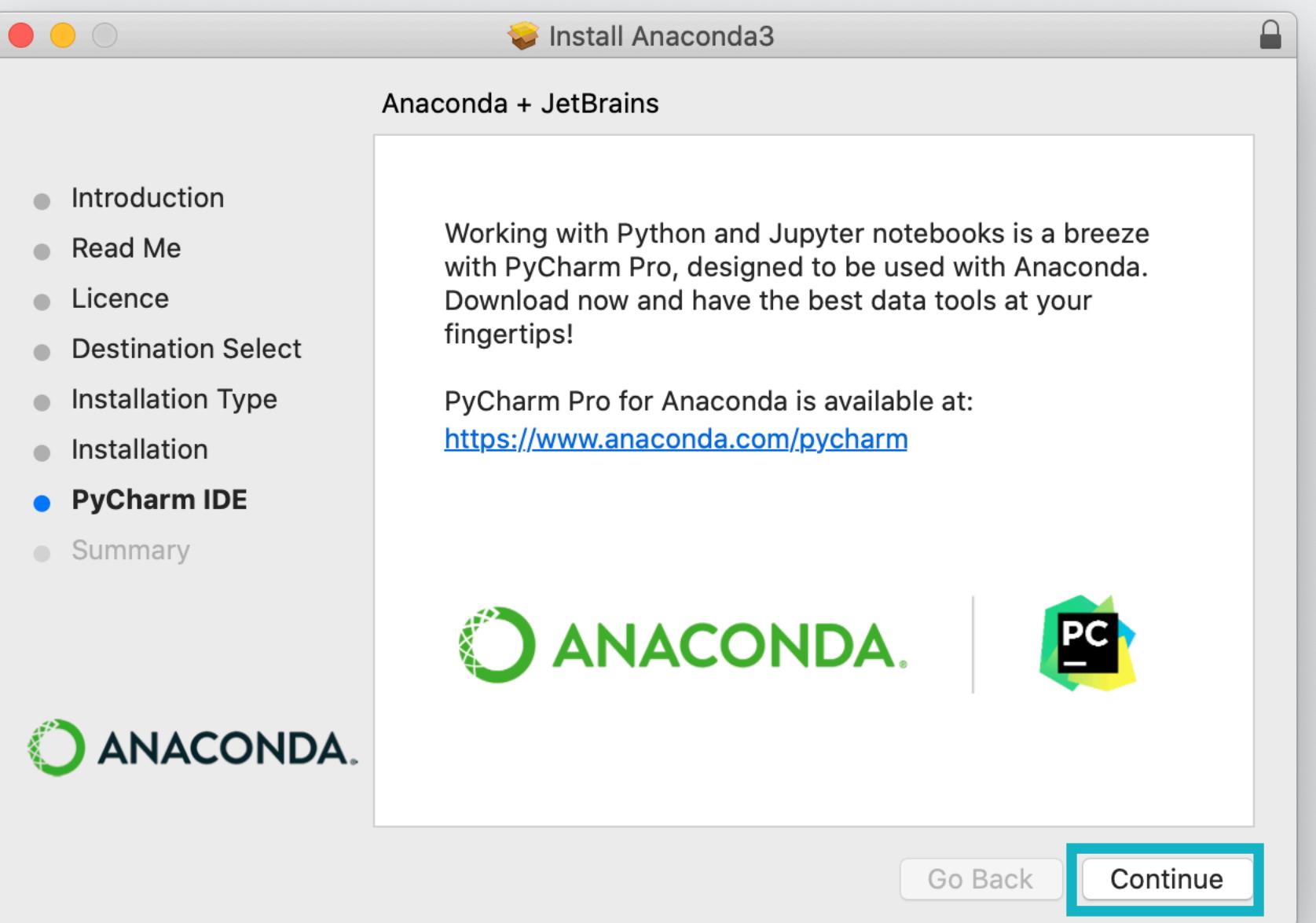
\* 계속을 눌러주시면 됩니다!



\* 일부 저와 다른 창이

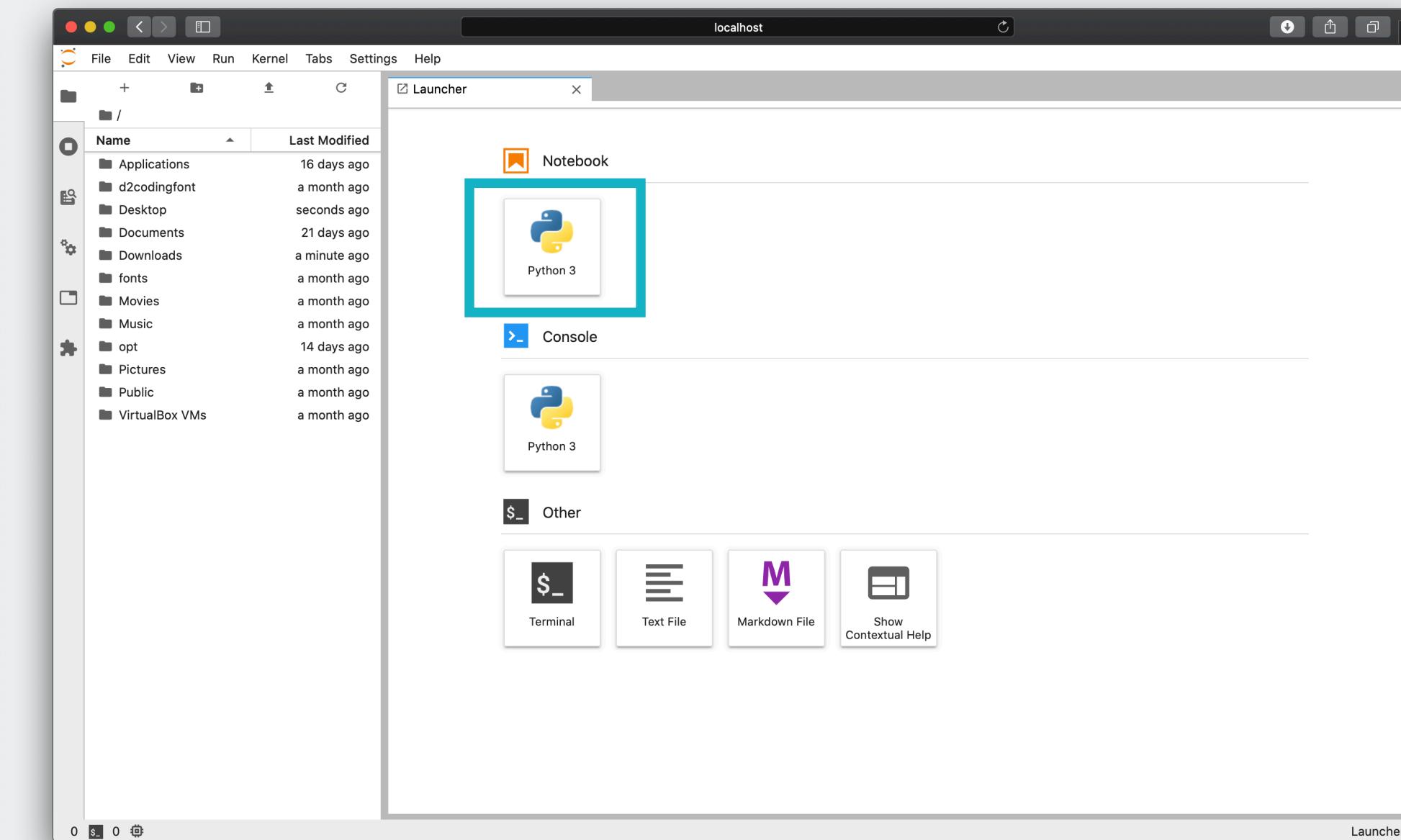
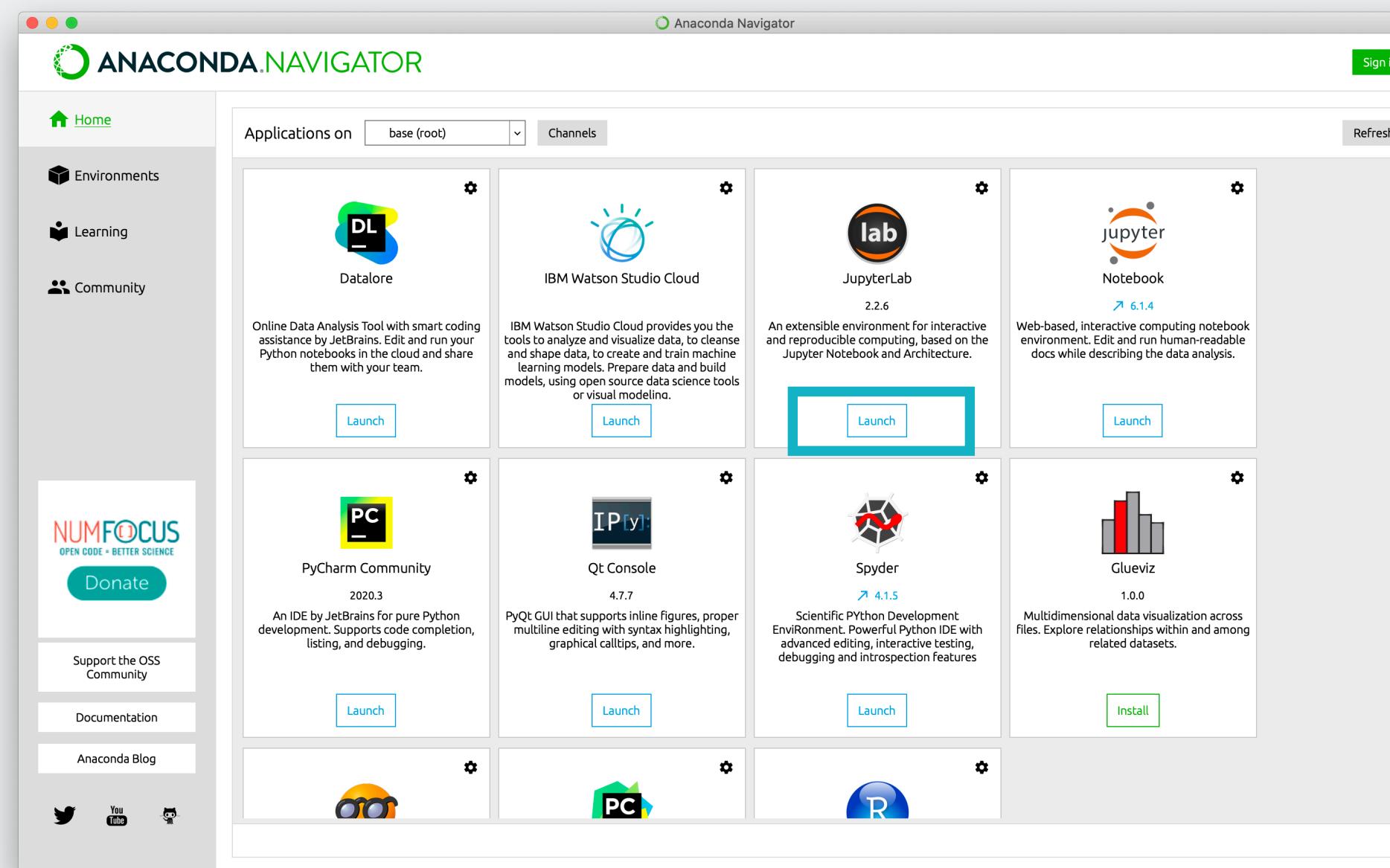
\* 뜨신다면 변경하지 않고

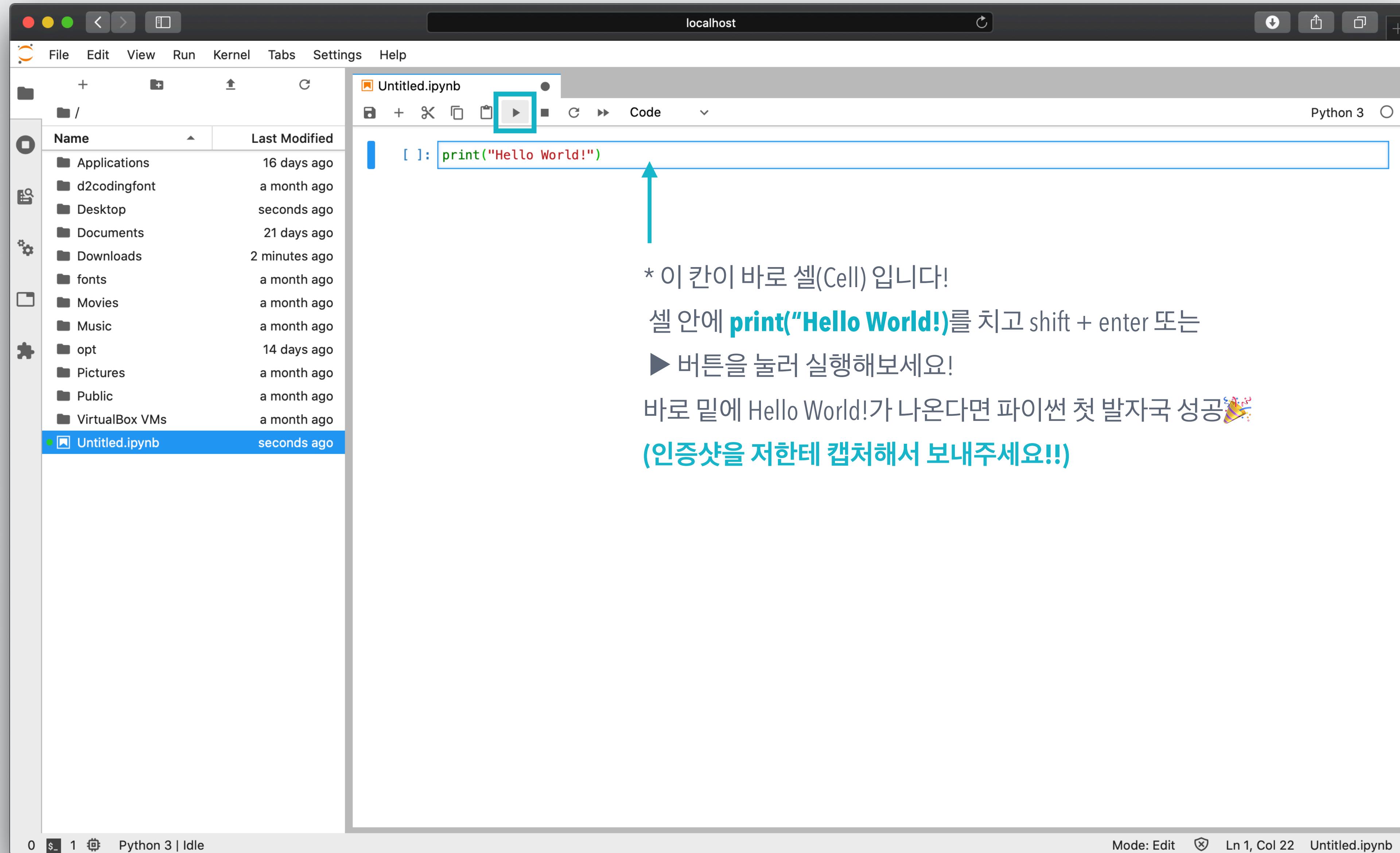
\* 계속을 눌러주시면 됩니다!



- \* 일부 저와 다른 창이
- \* 뜨신다면 변경하지 않고
- \* 계속을 눌러주시면 됩니다!

프로그램 Anaconda-Navigator 실행(못찾으시겠다면 cmd + space 후 검색하세요!)





\* 이 칸이 바로 셀(Cell)입니다!

셀 안에 **print("Hello World!")**를 치고 shift + enter 또는 ► 버튼을 눌러 실행해보세요!

바로 밑에 Hello World!가 나온다면 파이썬 첫 발자국 성공🎉  
**(인증샷을 저한테 캡처해서 보내주세요!!)**

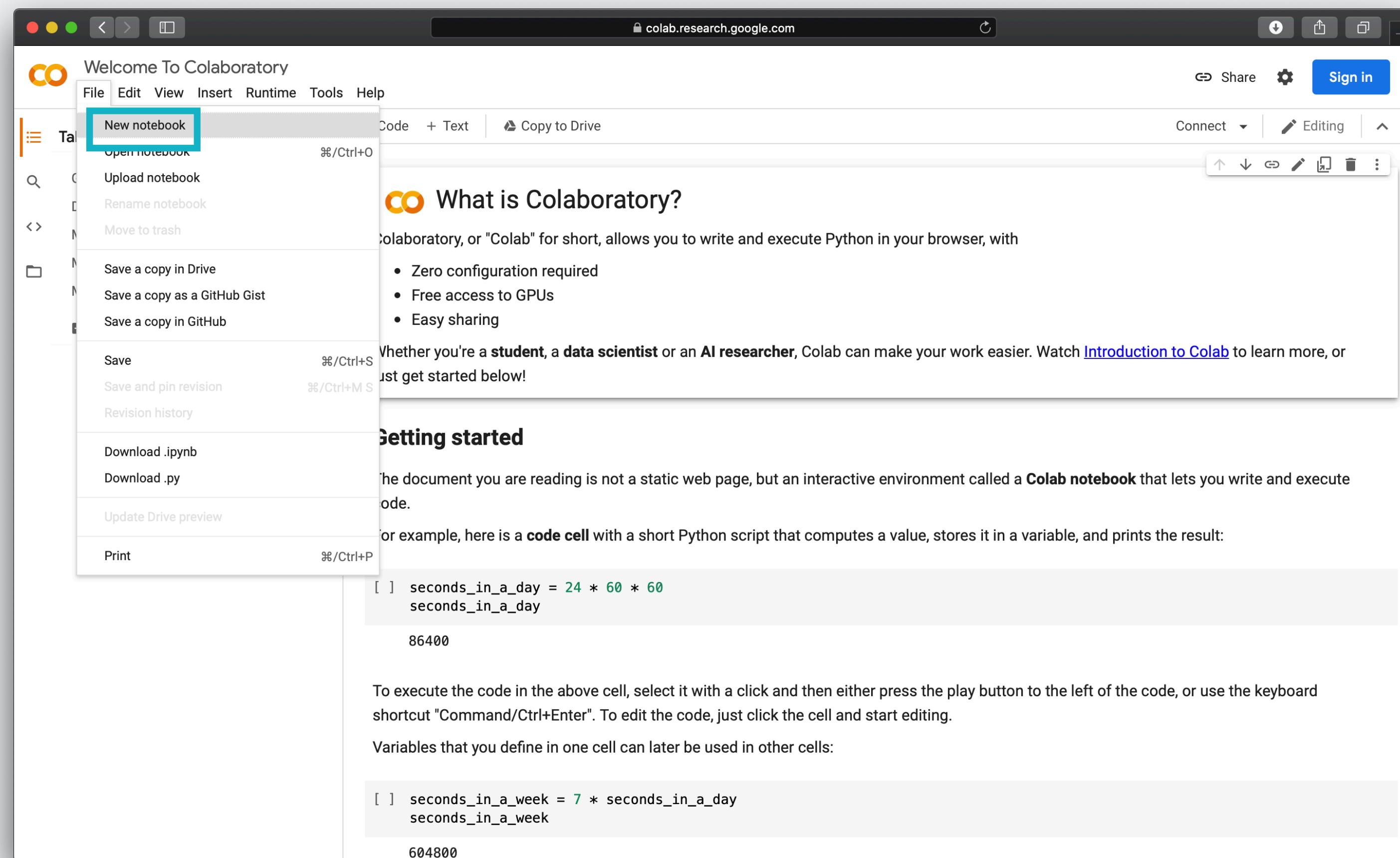
```
0 $ 1 Python 3 | Idle Mode: Edit  LN 1, Col 22 Untitled.ipynb
```

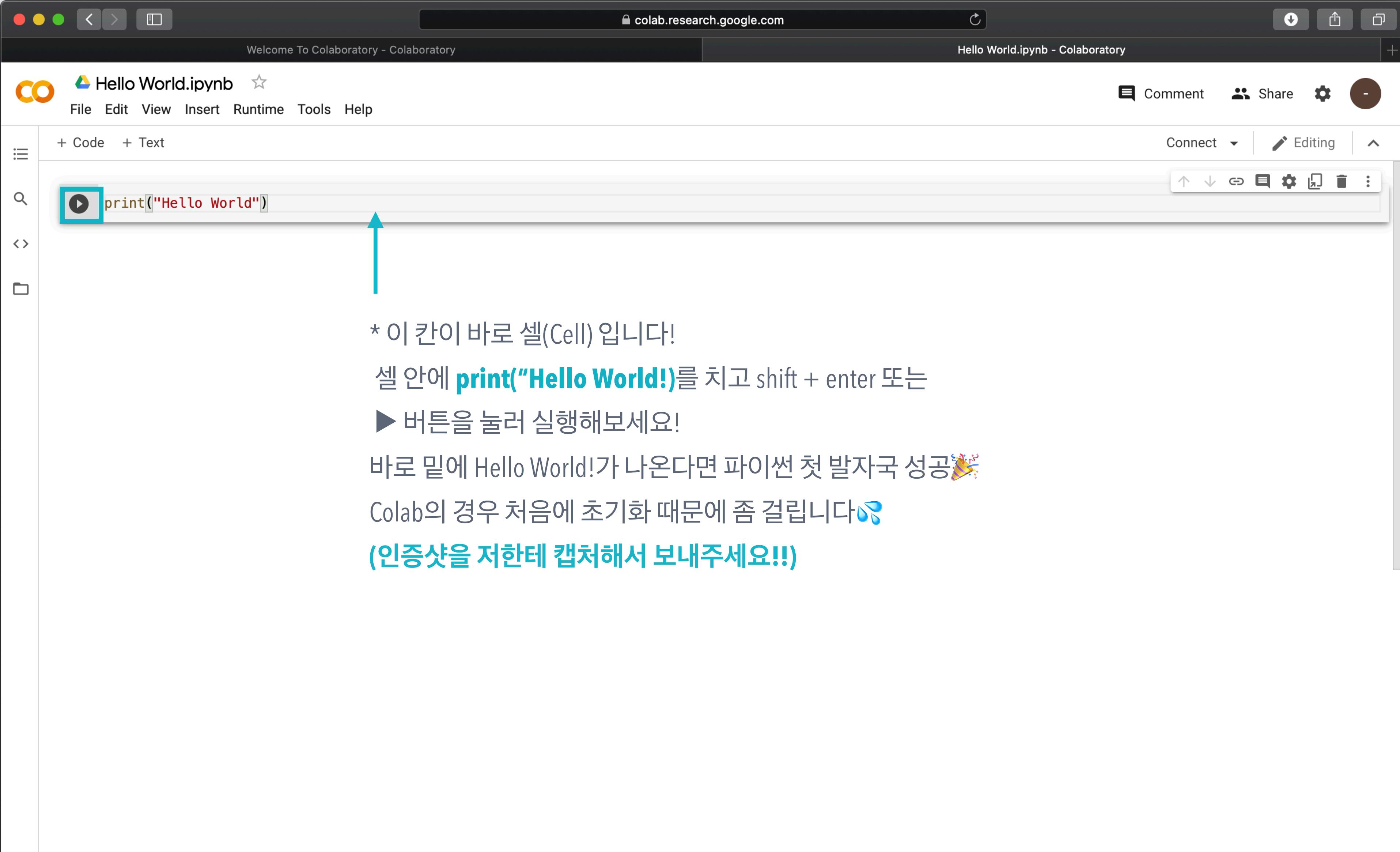
## COLAB

Colab은 구글에서 제공하는 클라우드 기반의 Jupyter Notebook 개발 환경입니다! 컴퓨터에 파이썬이 설치되어 있지 않아도 사용가능 하고, 인터넷만 있다면 사용할 수 있기 때문에 무척 편리합니다. 하지만 프로그램을 실행한지 12시간이 지나거나, 중간에 인터넷이 잠시라도 끊기면 리셋되어 데이터가 날라갈 수 있고, 사용자가 몰리면 느려지기 때문에 개인적으로는 좋아하지 않습니다 하지만 알 수 없는 이유로 프로그램이 설치되지 않거나 용량이 부족하다면 입문자용으로는 Colab도 전혀 무리가 없습니다~~

! 구글 계정이 필요합니다 !

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





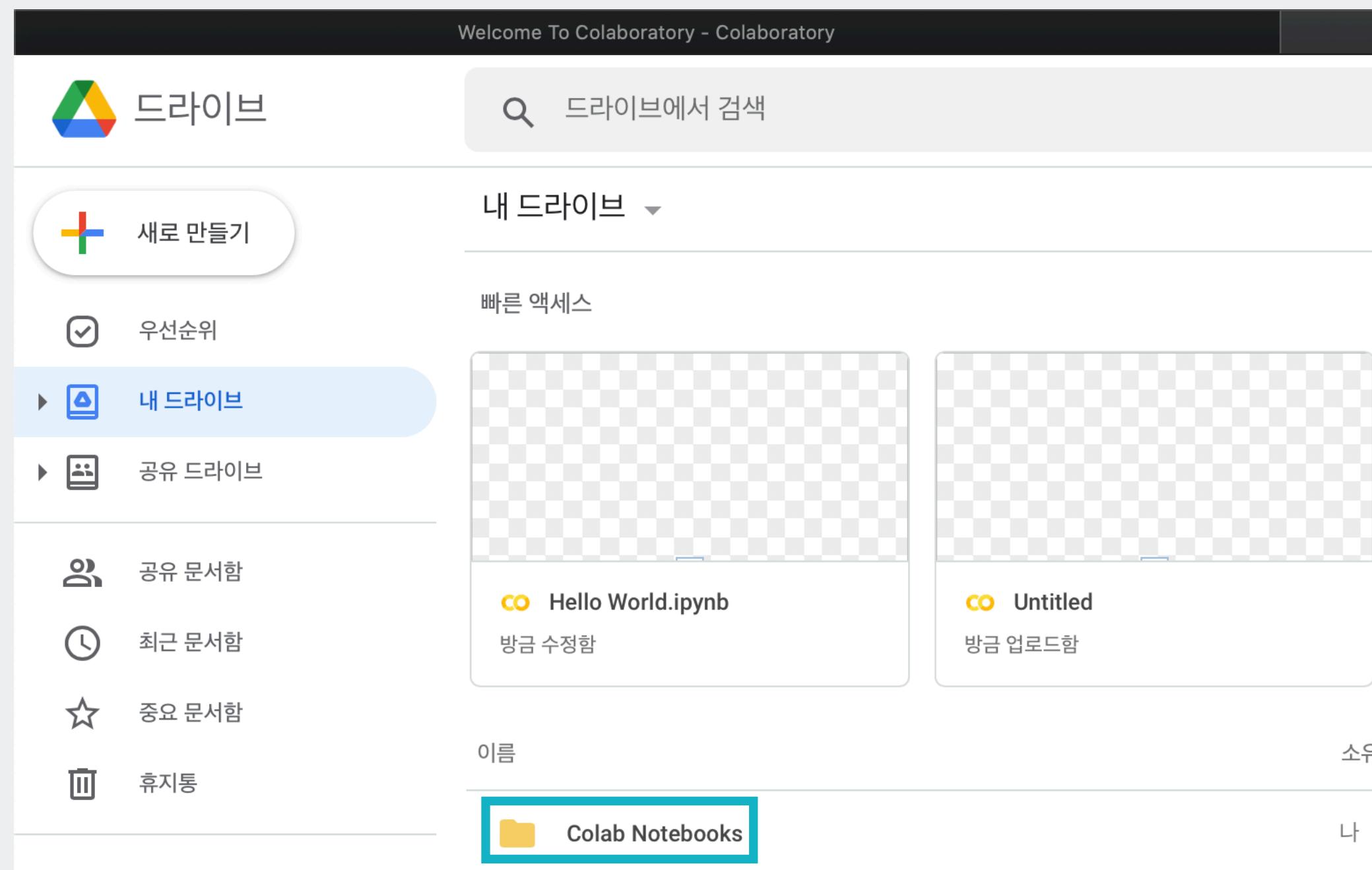
The screenshot shows a Google Colab notebook titled "Hello World.ipynb". A single code cell contains the Python command `print("Hello World")`. A blue arrow points from the explanatory text below to the play button icon in the cell toolbar.

\* 이 칸이 바로 셀(Cell)입니다!

셀 안에 **print("Hello World!")**를 치고 shift + enter 또는  
▶ 버튼을 눌러 실행해보세요!

바로 밑에 Hello World!가 나온다면 파이썬 첫 발자국 성공🎉  
Colab의 경우 처음에 초기화 때문에 좀 걸립니다💦

**(인증샷을 저한테 캡처해서 보내주세요!!)**



+ 생성한 파일은 해당 계정 구글 드라이브에서 확인 가능하고,  
한 번 접속 후에는 구글 드라이브에서 바로 파일을 만드는 것도  
가능합니다!

