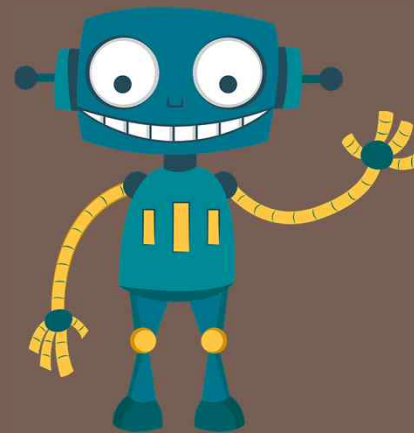


파이썬 익스프레스



1 장 프로그래밍과 파이썬 소개

Q & A

- 언제라도 질문하세요
 1. 강의시간의 채팅창 (수신인: 모두, 수강생모두에게 답변하기에)
 2. 네이버카페 질의응답게시판

- 강사의 1번 선생님은 여러분의 질문

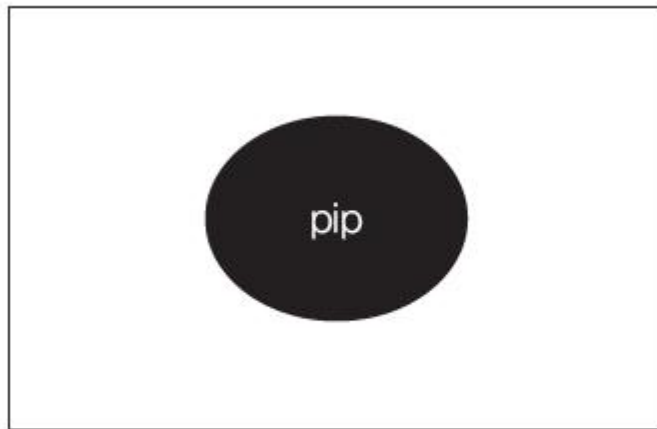


아프로 나는 할 수 있다

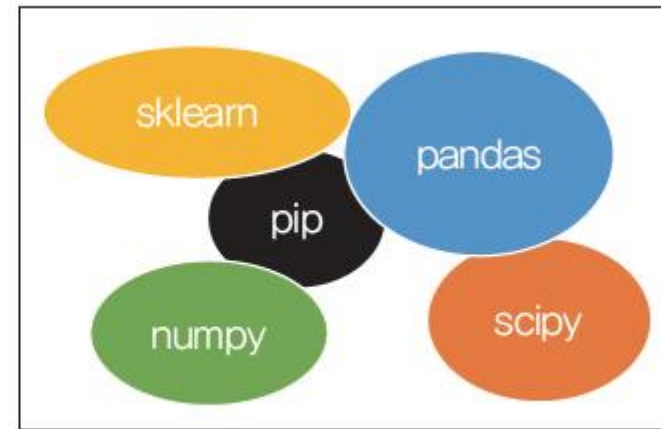
1. 나는 개발환경인 **anaconda**를 내 피씨에 설치할 수 있다
2. 나는 개발환경인 **anaconda**를 사용할 수 있다

아나콘다

- 인기 있는 라이브러리가 거의 모두 포함된 배포판
- python.org 와 anaconda.com



파이썬



아나콘다

아나콘다 다운로드 (교재와 동일한 버전으로 강의)

- 네이버카페에서 다운로드 : `anaconda3 2019.10(64bit)`

최신 아나콘다 다운로드

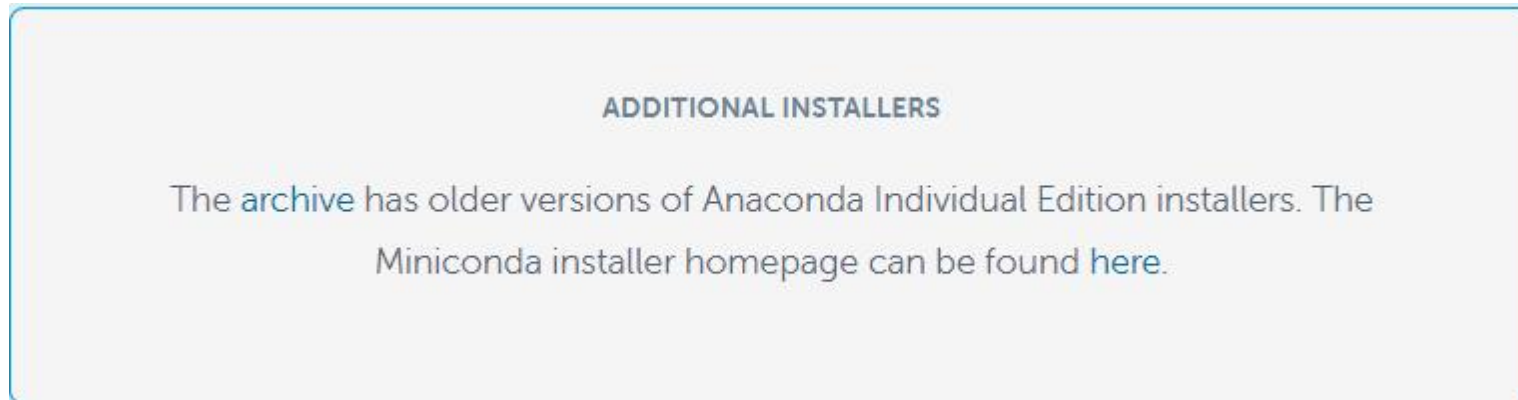
- <https://www.anaconda.com/products/individual>

Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (457 MB)	64-Bit Graphical Installer (435 MB)	64-Bit (x86) Installer (529 MB)
32-Bit Graphical Installer (403 MB)	64-Bit Command Line Installer (428 MB)	64-Bit (Power8 and Power9) Installer (279 MB)

옛날 아나콘다 다운로드

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

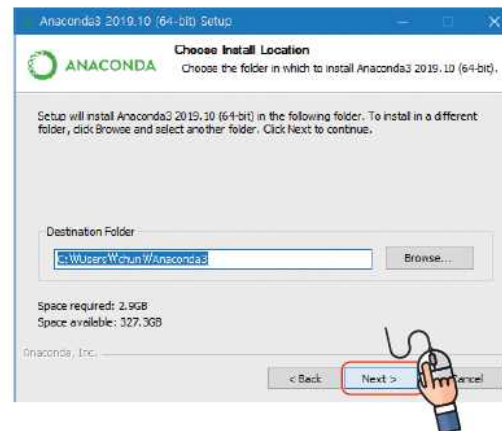
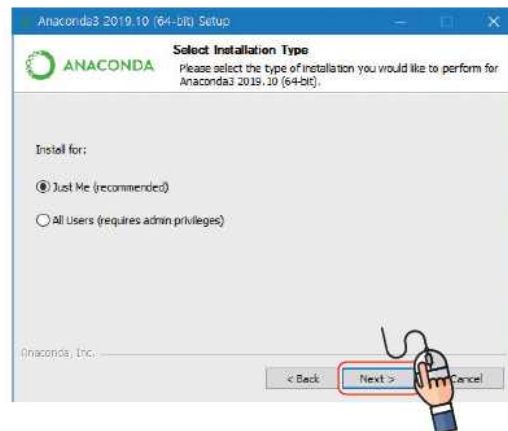
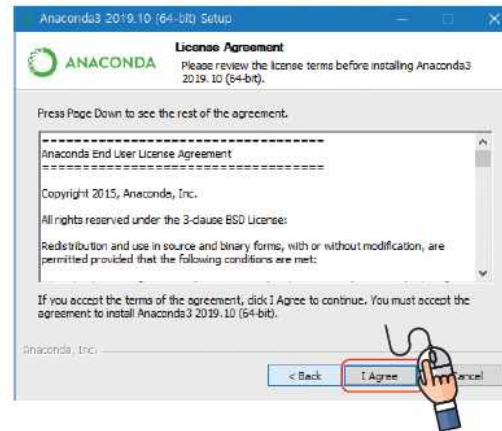
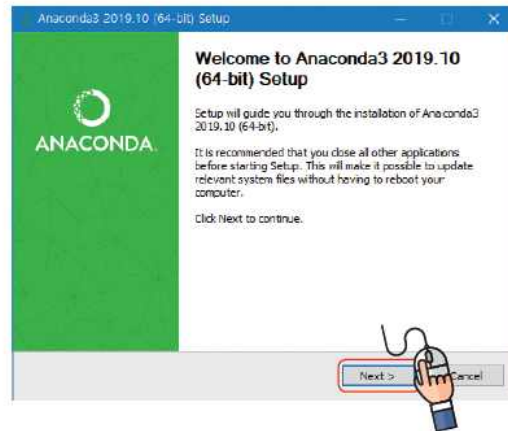


- <https://repo.anaconda.com/archive/>

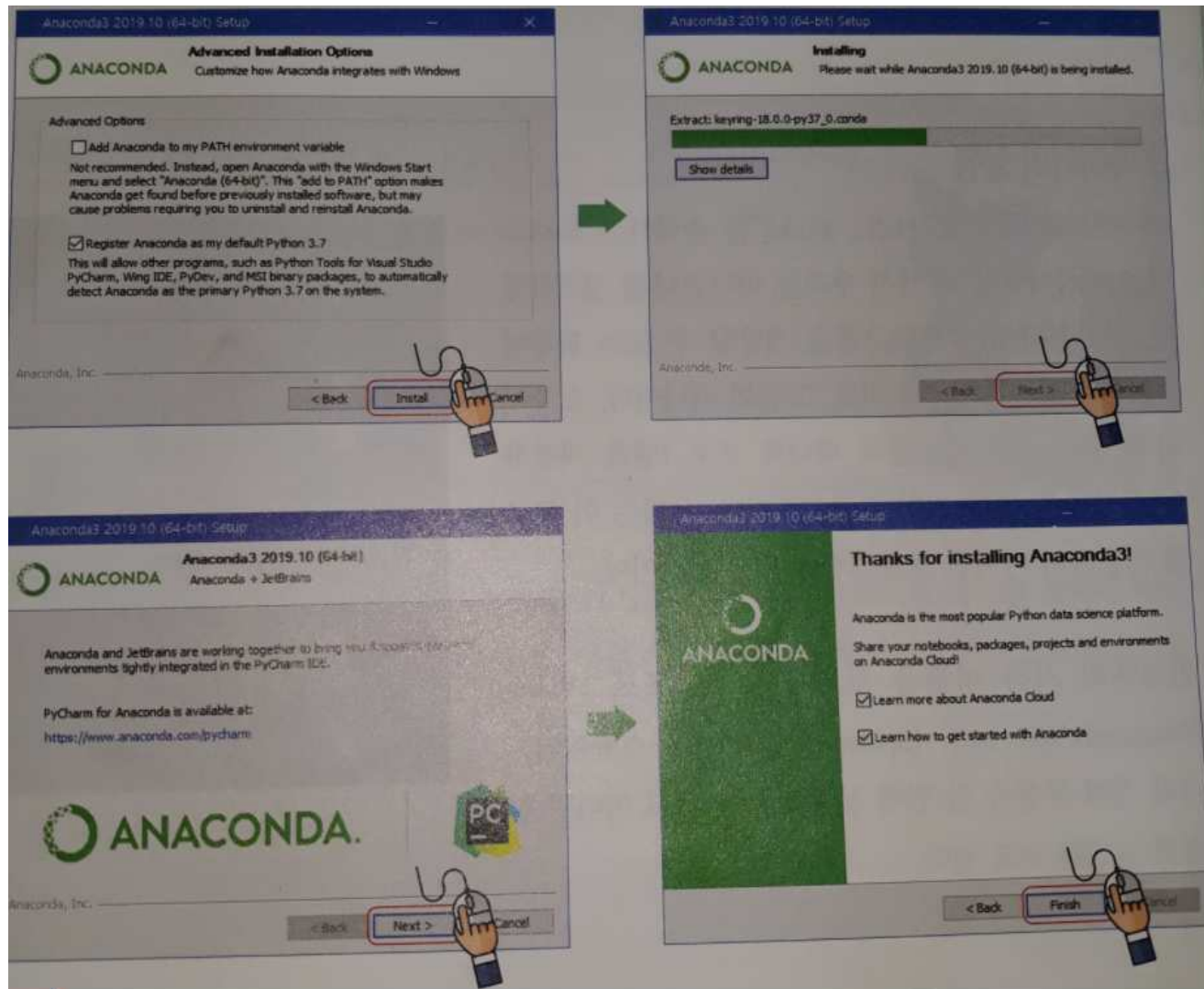
A screenshot of a web browser window showing the index of files in the Anaconda repository. The browser's address bar shows the URL "repo.anaconda.com". The page title is "Index of /". The table lists various Anaconda installer files with their sizes.

Anaconda3-2019.07-MacOSX-x86_64.sh	435.4M
Anaconda3-2019.07-Windows-x86.exe	418.4M
Anaconda3-2019.07-Windows-x86_64.exe	485.8M
Anaconda3-2019.10-Linux-ppc64le.sh	320.3M
Anaconda3-2019.10-Linux-x86_64.sh	505.7M
Anaconda3-2019.10-MacOSX-x86_64.pkg	653.5M
Anaconda3-2019.10-MacOSX-x86_64.sh	424.2M
Anaconda3-2019.10-Windows-x86.exe	409.6M
Anaconda3-2019.10-Windows-x86_64.exe	461.5M

아나콘다 설치 - 1 (옵션 변경 없이 설치)



아나콘다 설치-2 (옵션 변경없이 설치)

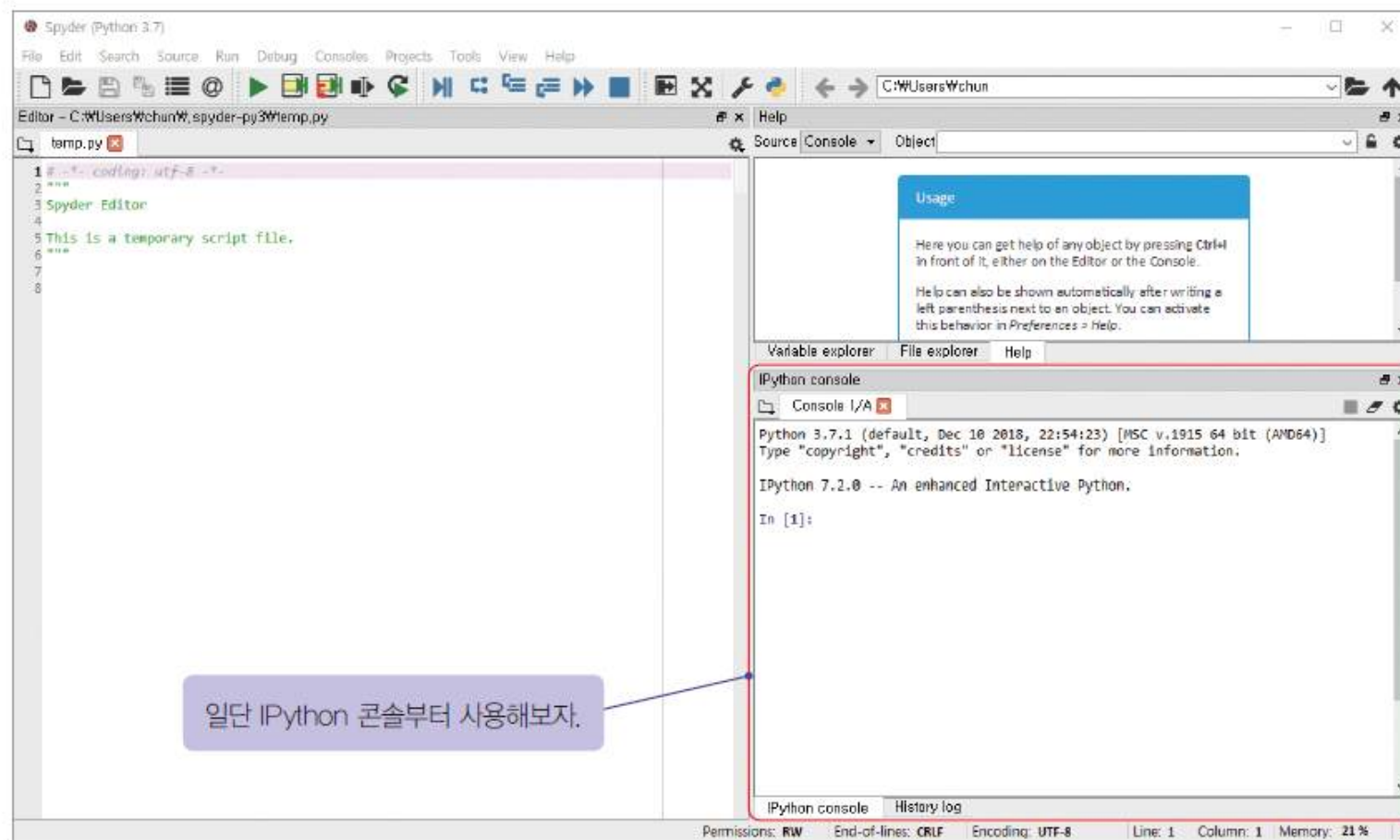


스파이더

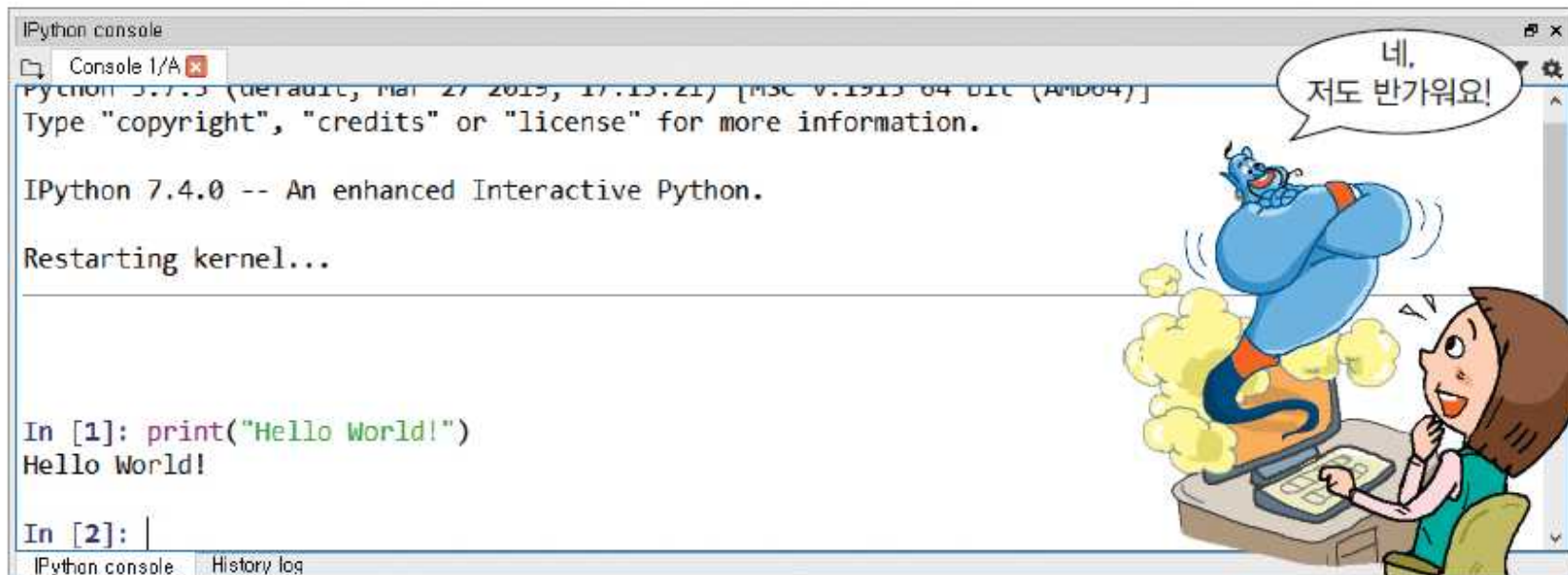
- 스파이더는 파이썬으로 작성된 파이썬 개발 도구

 Editor 함수 / 클래스 브라우저, 코드 분석 도구, 자동 코드 완성, 수평 / 수직 분할을 사용하여 효율적으로 작업이 가능하다.	 IPython Console GUI 인터페이스 안에서 코드를 라인별로 실행하거나 인라인으로 차트를 그릴 수 있다.	 Variable Explorer 변수와 상호작용하거나 변수를 변경할 수 있다. 히스토그램을 그리거나 데이터 프레임을 편집하고 컬렉션을 정렬할 수도 있다.
 Profiler 코드에서 가장 시간을 잡아먹는 부분을 바로 알아내서 제거할 수 있다.	 Debugger 디버거를 사용하여 코드를 한 줄씩 실행할 수 있다.	 Help 클래스에 관한 도움말을 즉시 볼 수 있다.

ipython



ipython



IPython console

Console 1/A

Python 3.7.5 (default, Mar 27 2019, 17:13:21) [MSC v.1913 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

Restarting kernel...

In [1]: `print("Hello World!")`
Hello World!

In [2]:

Python console History log

네, 저도 반가워요!

대화형 모드와 스크립트 모드

- 대화형 모드(interactive mode): 콘솔에서 문장을 한 줄씩 입력하여 실행
- 스크립트 모드(script mode): 파일을 만들어서 저장한 후에 파이썬 인터프리터가 이 파일을 읽어서 한 번에 전부 실행

대화형 모드	스크립트 모드
<pre>>>> print("Hello World!") Hello World! >>> print(10+20) 30</pre>	<pre>print("Hello World!") print(10+20)</pre> <p>실행결과</p> <pre>Hello World! 30</pre>

문장이 즉시 실행되고 실행결과가 바로 나타난다.

▶ 버튼을 눌러야 문장들이 실행되고 실행결과는 콘솔에 나타난다.

스크립트 모드

- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

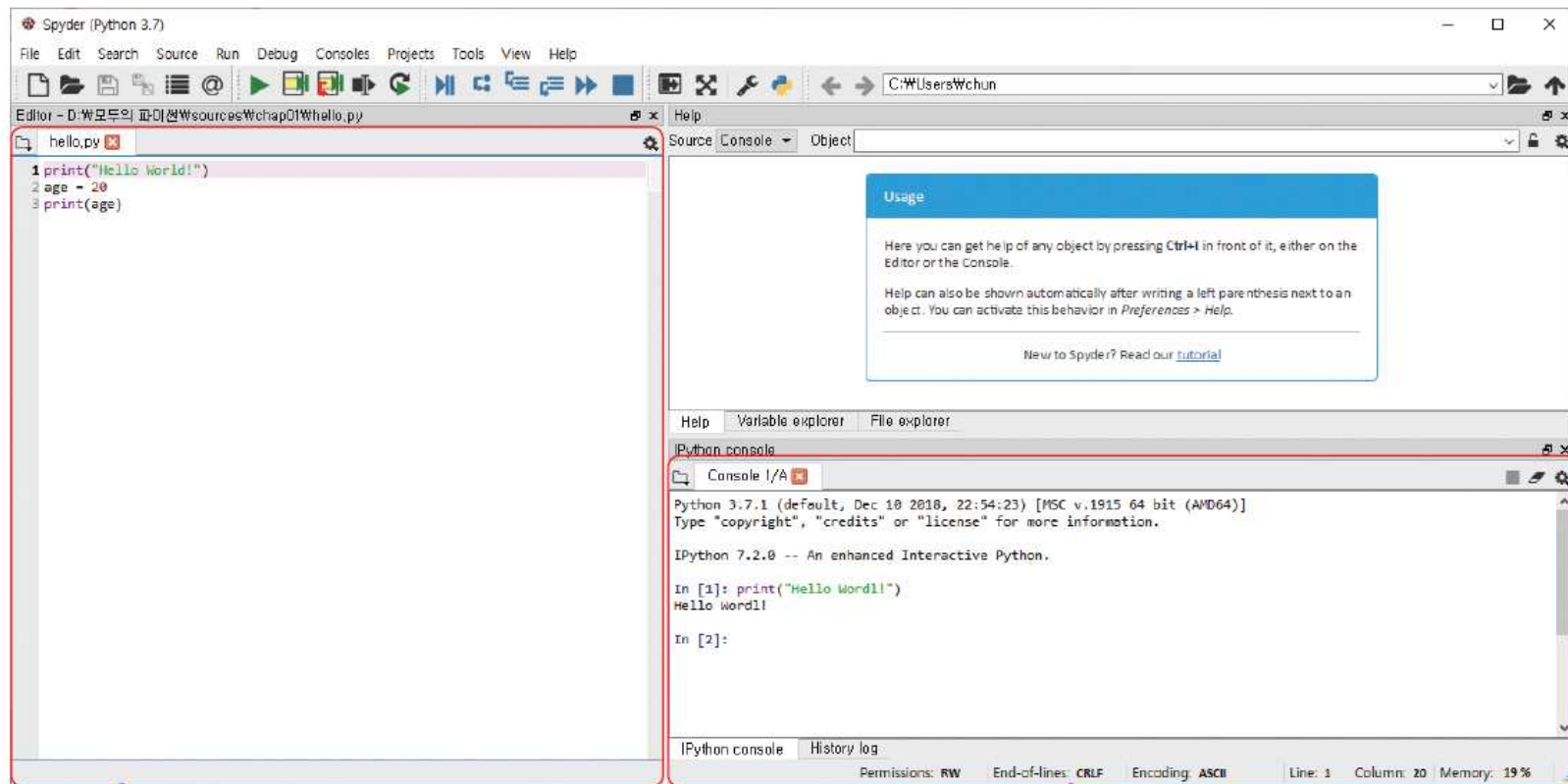
이걸 한 줄씩 입력하라고?



```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print ' %s (label="%s' % (nodename, label)
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s';' % ast[1]
        else:
            print '='
    else:
        print '[]';
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename
        for n, child in enumerate(children):
            print '%s' % child,
```

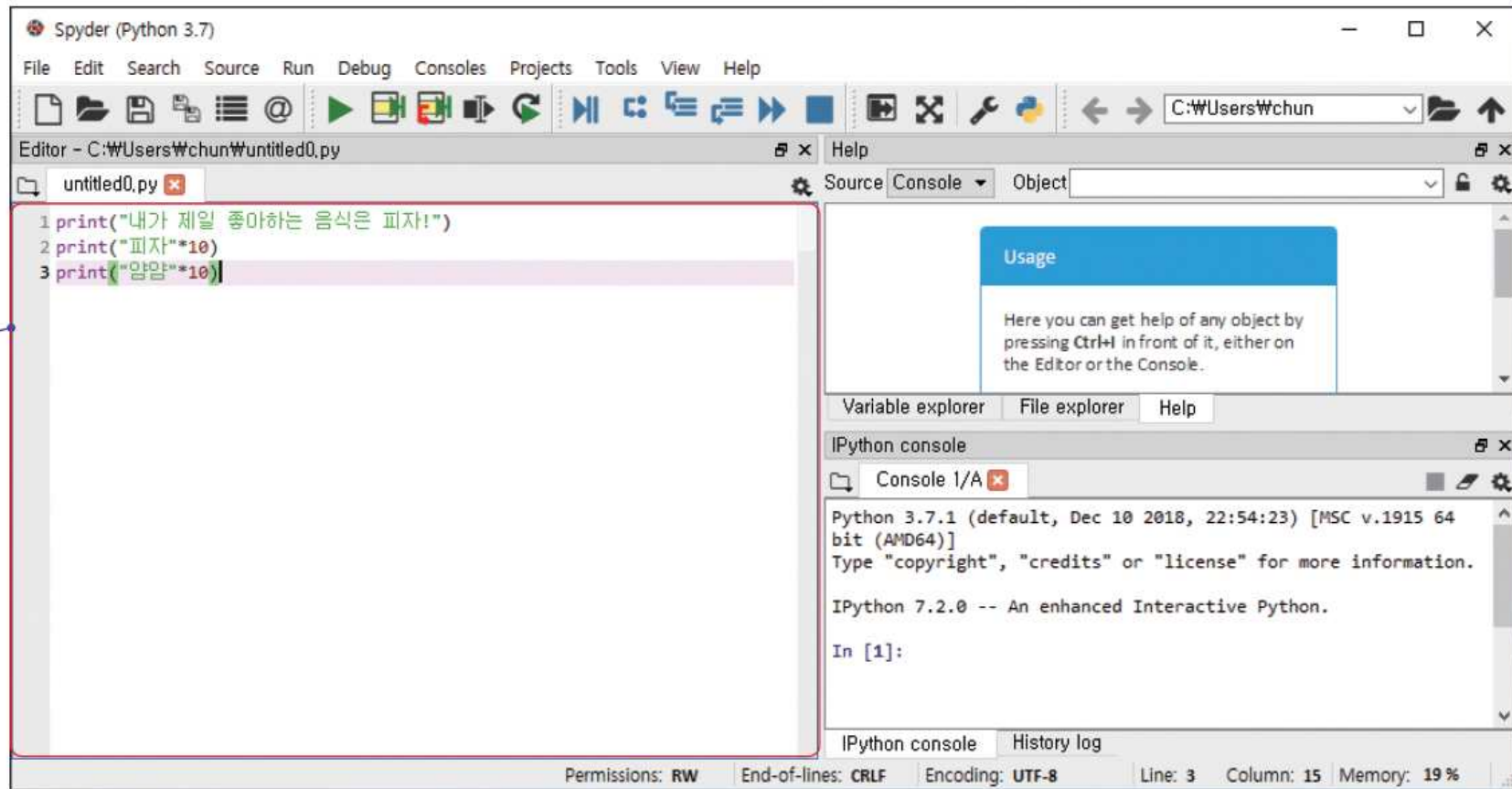
스파이더에서 대화형 모드와 스크립트 모드



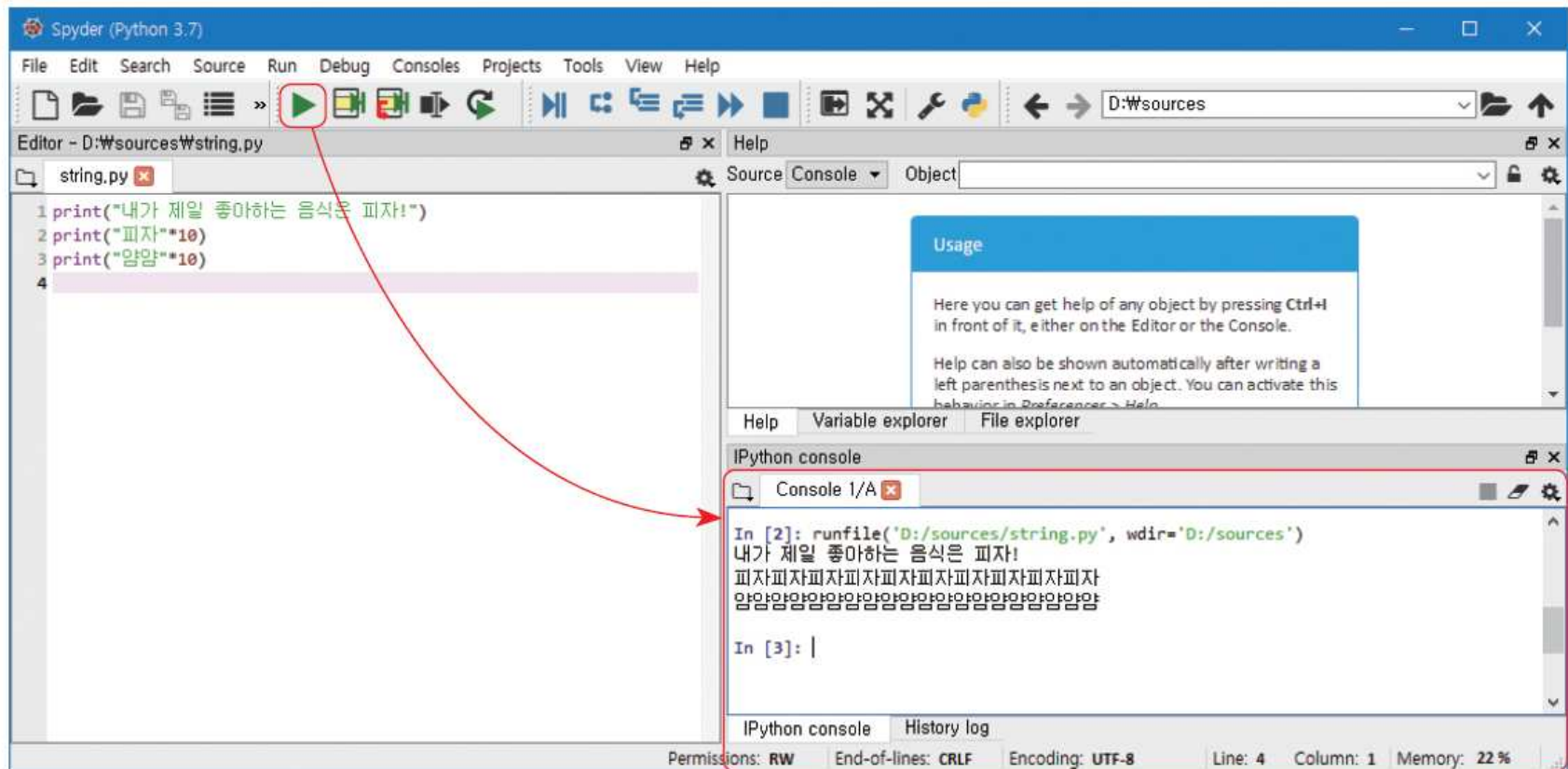
여기다 입력하면 스크립트
모드로 실행된다.

여기다 입력하면 대화형
모드로 실행된다.

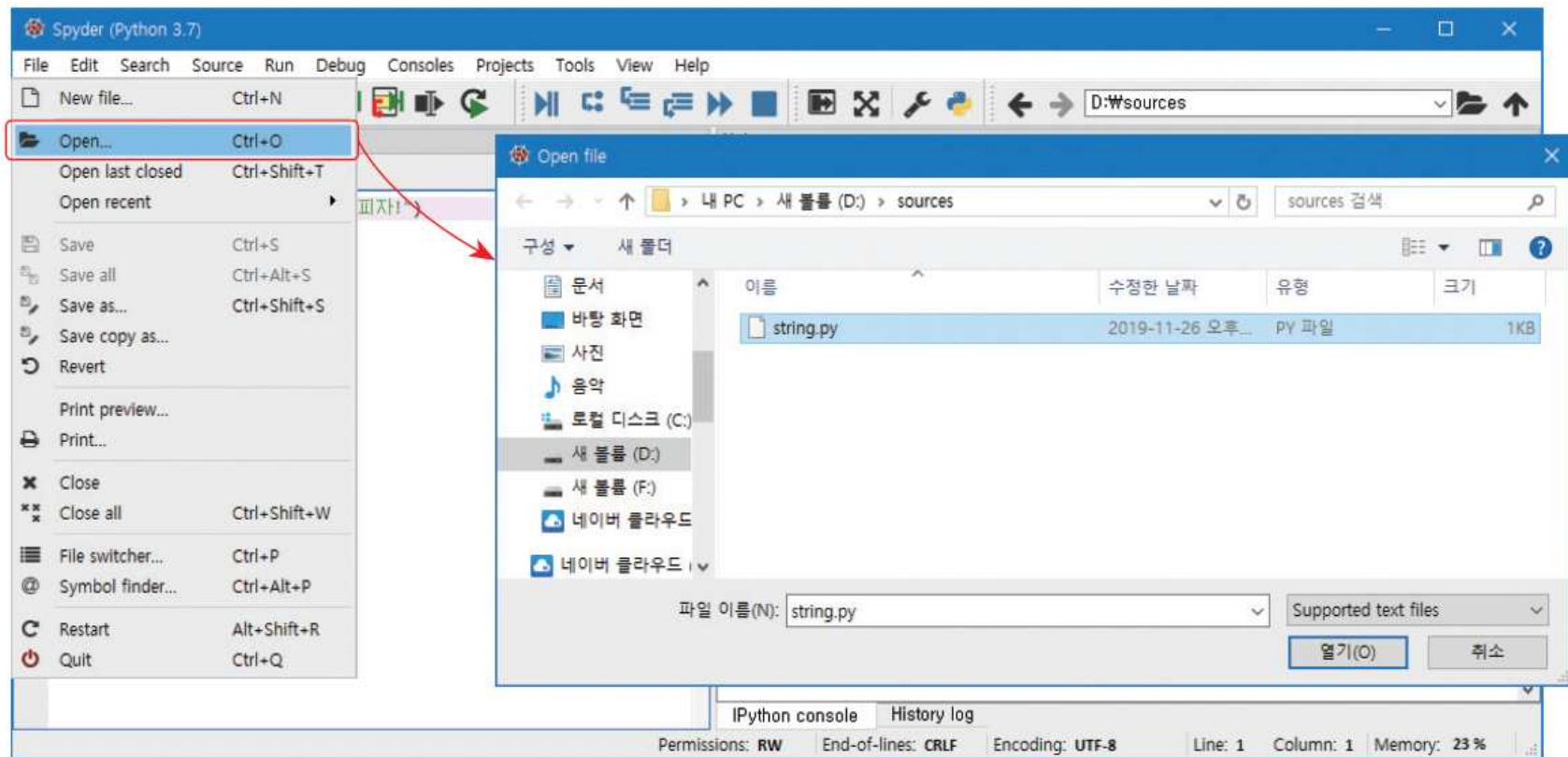
스크립트 모드



스크립트 모드



소스 파일 다시 열기



Lab: print() 함수 실습 (스크립트 모드)

파이썬에 오신 것을 환영합니다.

파이썬은 쉽습니다.

파이썬으로 비데이터, 인공지능 프로그램을 작성할 수 있습니다.

Sol: print() 함수 실습 (lab1.py)

```
print("파이썬에 오신 것을 환영합니다.")  
print("파이썬은 쉽습니다.")  
print("파이썬으로 빅데이터, 인공지능 프로그램을 작성할 수 있습니다.")
```

디버깅

- 디버깅에는 compile time debugging과 run time debugging 만 있음

< 심규현강사 입장에서 >

- ASIC 디버깅이 제일 어려움. Pin 신호만 볼 수 있음
- FPGA는 동작 못시키는 것은 이해가 되지 않음
무료인 SignalTap과 ILA로 내부 신호를 모두 볼 수 있음
- 심규현강사는 디버깅 어려운 ASIC보다 디버깅이 쉬운 FPGA 좋아함
IDEC MPW를 많이 만들어 보아야함
- 언어 중에서는 SystemVerilog가 제일 어려움
객체지향언어이고 병렬언어. 회사에서는 검증시 SystemVerilog사용
<http://www.idec.or.kr/edu/apply/view/?no=1365>
- C언어는 Visual Studio, Java는 Eclipse가 있기에 디버깅 쉬움
- Bare metal 환경의 경우 Jtag 디버거 사용

Lab: compile time 오류를 처리해보자 (bug.py)

=> 00shim.py에서 수정

```
print(안녕하세요?)  
Print("이번 코드에는 많은 오류가 있다네요")  
print("제가 다 고쳐 보겠습니다.")
```

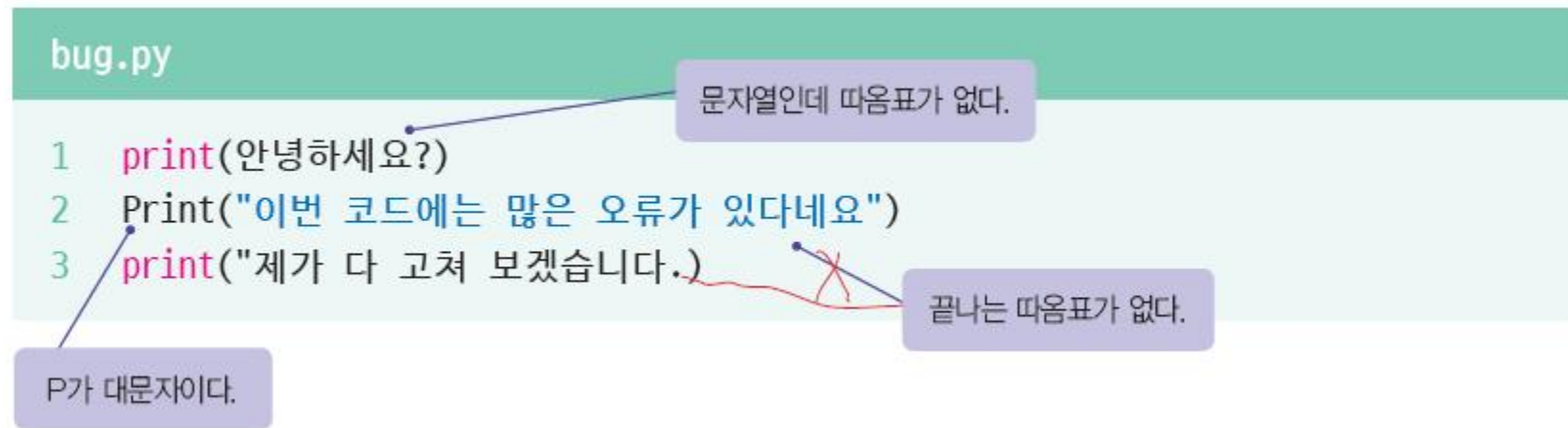
```
File "D:/모두의 파이썬/sources/chap01/hello.py", line 1  
    print(안녕하세요?)  
          ^  
SyntaxError: invalid syntax
```

```
File "C:/idec/sources/chap01/chap1.py", line 3  
    print("제가 다 고쳐 보겠습니다.")  
          ^  
SyntaxError: EOL while scanning string literal
```

```
File "C:/idec/sources/chap01/chap1.py", line 2, in <module>  
    Print("이번 코드에는 많은 오류가 있다네요")  
NameError: name 'Print' is not defined
```

- 에러 메시지 보아도 오류원인 파악이 어려움

Sol: compile time 오류를 처리해보자.



```
print("안녕하세요?")  
print("이번 코드에는 많은 오류가 있다네요")  
print("제가 다 고쳐 보겠습니다.")
```

6장 Lab: 리스트 변경 함수 (run time 오류를 처리해보자)

- 어떤 회사에서 리스트에 직원들의 월급을 저장하고 있다. 회사에서 일괄적으로 30%의 월급 인상을 하기로 하였다. 리스트의 모든 요소들을 30% 증가시키는 함수 `modify()`를 작성하고 테스트 해보자.

인상전 [200, 250, 300, 280, 500]

인상후 [260.0, 325.0, 390.0, 364.0, 650.0]



Solution: (6장 salary_.py)

(run time 오류를 처리해보자)

```
salaries = [200, 250, 300, 280, 500]
```

```
def modify(values, factor) :  
    for i in range(len(values)) :  
        values[i] = values[i] * factor
```

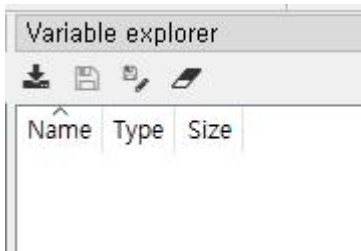
```
print("인상전", salaries)  
modify(salaries, 1.3)  
print("인상후", salaries)
```

디버깅 (run time 오류를 처리해보자)

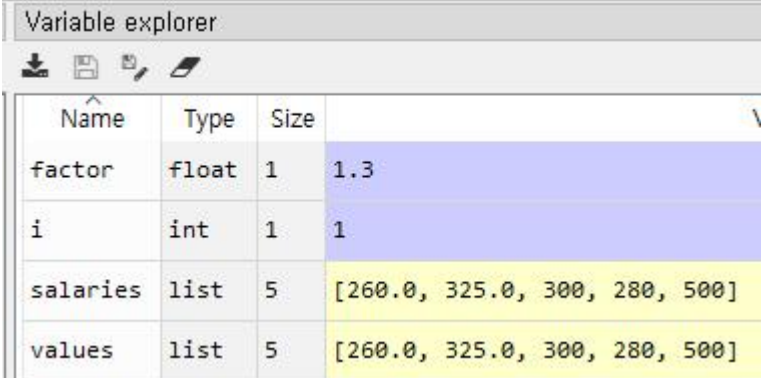

- 디버깅은 몇번째 줄에서 **error**가 발생했는지 찾고 수정하는 과정
- 디버깅의 원리 : 라인단위로 변수의 예측값과 실측값 비교

1단계: breakpoint 설정과 variable explorer 클리어

```
1 salaries = [200, 250, 300, 280, 500]
2
3 def modify(values, factor) :
4     for i in range(len(values)) :
5         values[i] = values[i] * factor
6
7 print("인상전", salaries)
8 modify(salaries, 1.3)
9 print("인상후", salaries)
```



2단계: 1줄씩 수행하면서 변수값을 예측값과 비교



Name	Type	Size	
factor	float	1	1.3
i	int	1	1
salaries	list	5	[260.0, 325.0, 300, 280, 500]
values	list	5	[260.0, 325.0, 300, 280, 500]

3단계: Stop debugging과 BP제거 후 코드 변경

Run time 디버깅의 특징

- run time 오류는 검색어를 작성할 수 없어서 구글링이나 stackoverflow.com에서 답을 구할 수 없음. Compile time 오류는 error 메시지로 구글링
(검색어 예: 화면에 아무것도 나오지 않아요?)
- 디버깅은 오류가 있어야 해볼 수 있기에 친구 에러 발생시 적극적으로 도와주면서 본인의 디버깅 실력을 향상할 수 있음
대학강의에서는 심규현강사의 디버깅 실력이 가장 많이 향상됨

이제 나는 할 수 있다

1. 나는 개발환경인 **anaconda**를 내 피씨에 설치할 수 있다
2. 나는 개발환경인 **anaconda**를 사용할 수 있다

HW1장

- HW13장-1, HW13장-2 게시판에 업로드