

대학3부 파이썬 스터디

2018.01.06 ~ 2018.02.24

Lecture 1 설치 및 기본 구조

INTRO..

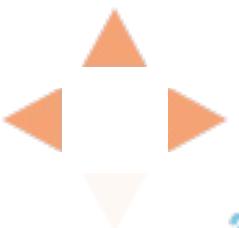
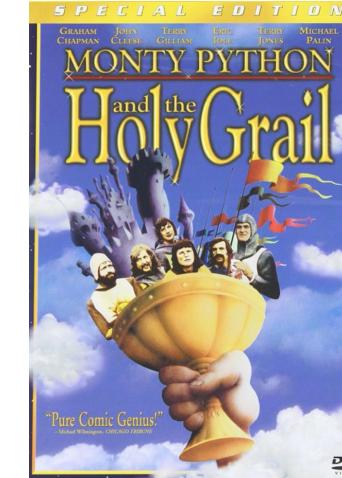
PYTHON 이란?



PYTHON 이란?

- ⦿ 파이썬: Guido Van Rossum이 개발한 컴퓨터 언어.
- ⦿ 이름은 즐겨 보던 코미디 쇼 "몬티 파이썬의 날아다니는 서커스"에서 착안.

GUIDO VAN ROSSUM



PYTHON 이란?



WHY PYTHON?

- ④ 간결하며 가독성이 좋아 입문이 어렵지 않다

WHY PYTHON?

- ◎ 간결하며 가독성이 좋아 입문이 어렵지 않다

int x; char *y; x = 10; printf("%d", x) y = "Hello World" printf("%s", y)	x = 10 print x x = "Hello World" print x
--	---



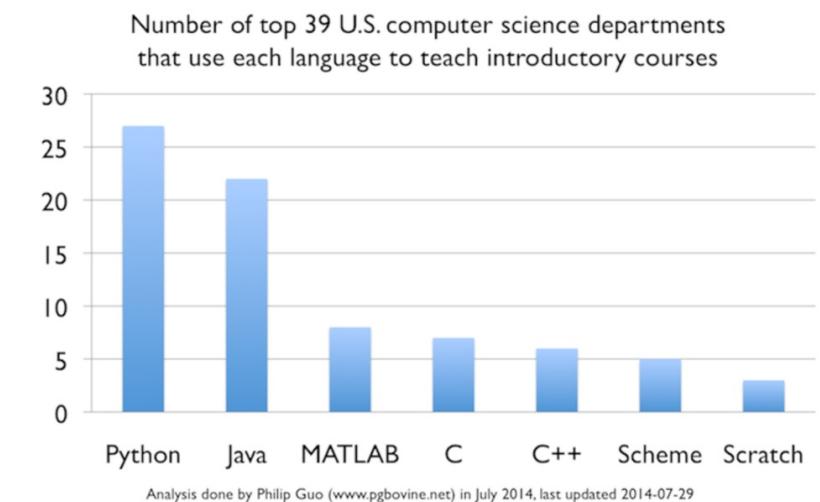
WHY PYTHON?

- ④ 컴퓨터 언어를 이해하고 배우기에 가장 적합한 언어로써 미국 상위 39개 대학의 컴퓨터공학 개론 수업에서 가장 많이 쓰이는 언어



WHY PYTHON?

- ⦿ 컴퓨터 언어를 이해하고 배우기에 가장 적합한 언어로써 미국 상위 39개 대학의 컴퓨터공학 개론 수업에서 가장 많이 쓰이는 언어



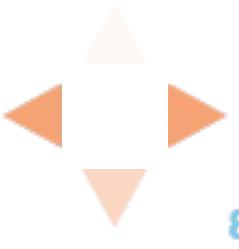
WHY PYTHON?

- ④ 파이썬은 무료

FREE

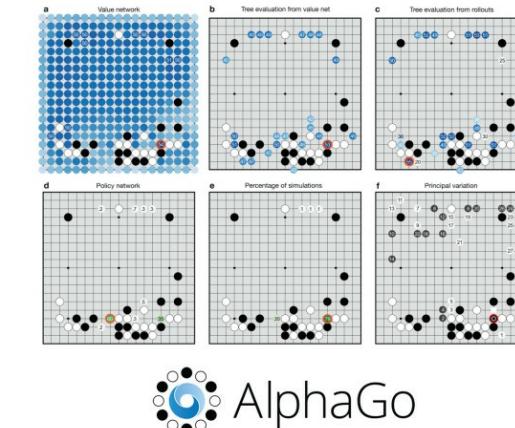
WHY PYTHON?

- ⦿ 파이썬은 Facebook 과 Google 의 인공지능 알고리즘의 핵심 언어이며 Yahoo, EventBrite, Reddit, Disqus, Nasa 등 다양한 기업이 활발하게 사용하고 있는 언어



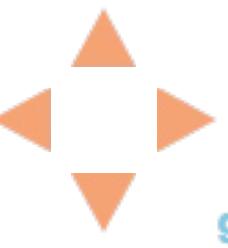
WHY PYTHON?

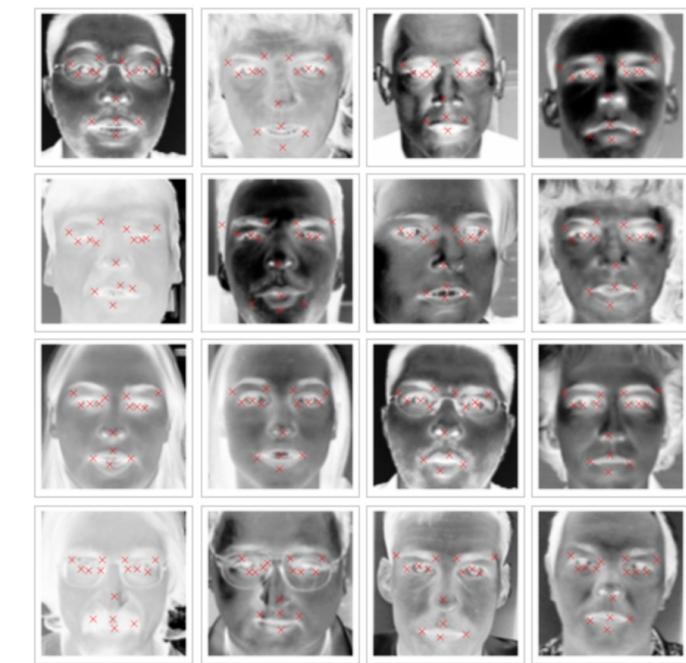
- ⦿ 파이썬은 Facebook 과 Google 의 인공지능 알고리즘의 핵심 언어이며 Yahoo, EventBrite, Reddit, Disqus, Nasa 등 다양한 기업이 활발하게 사용하고 있는 언어



In []:

```
net1 = NeuralNet(  
    layers=[ # three layers: one hidden layer  
        ('input', layers.InputLayer),  
        ('hidden', layers.DenseLayer),  
        ('output', layers.DenseLayer),  
    ],  
    # layer parameters:  
    input_shape=(None, 9216), # 96x96 input pixels per batch  
    hidden_num_units=100, # number of units in hidden layer  
    output_nonlinearity=None, # output layer uses identity function  
    output_num_units=30, # 30 target values  
  
    # optimization method:  
    update=nesterov_momentum,  
    update_learning_rate=0.01,  
    update_momentum=0.9,  
  
    regression=True, # flag to indicate we're dealing with regression problem  
    max_epochs=100, # we want to train this many epochs  
    verbose=1,  
)  
net1.fit(X, y)
```





WHY PYTHON?

- 파이썬은 주식과 같은 데이터 분석에도 활용되고 있다.

```
In [27]: import pandas as pd  
from pandas import Series, DataFrame  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from pandas.io.data import DataReader  
from datetime import datetime  
from __future__ import division  
tech_list = ['AAPL', 'GOOG', 'MSFT', 'AMZN']  
end = datetime.now()  
start = datetime(end.year-1, end.month, end.day)  
for stock in tech_list:  
    globals()[stock] = DataReader(stock, 'yahoo', start, end)
```



In [28]: # 애플 사의 지난 1년간 주가 정보

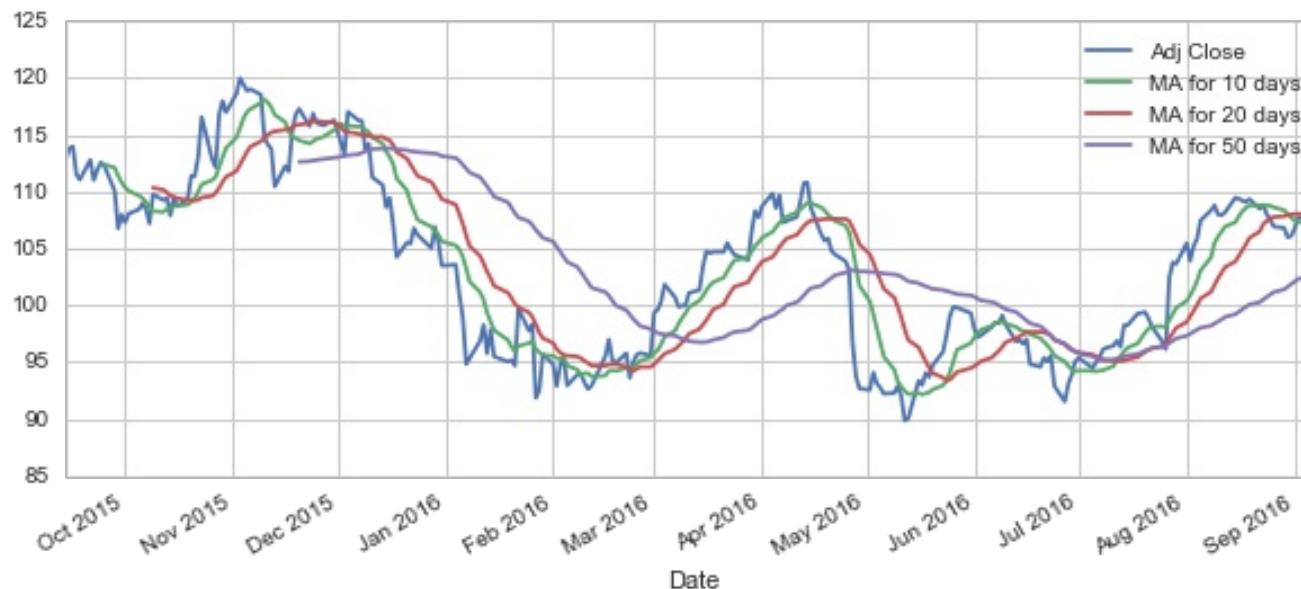
AAPL.head()

Out[28]:

	Open	High	Low	Close	Volume	Adj Close
Date						
2015-09-14	116.580002	116.889999	114.860001	115.309998	58363400	112.896168
2015-09-15	115.930000	116.529999	114.419998	116.279999	43341200	113.845864
2015-09-16	116.250000	116.540001	115.440002	116.410004	37173500	113.973148
2015-09-17	115.660004	116.489998	113.720001	113.919998	64112600	111.535266
2015-09-18	112.209999	114.300003	111.870003	113.449997	74285300	111.075104

```
In [29]: ma_day = [10,20,50]
for ma in ma_day:
    column_name = "MA for %s days" %(str(ma))
    AAPL[column_name] = pd.rolling_mean(AAPL['Adj Close'], ma)
AAPL[['Adj Close', 'MA for 10 days',
       'MA for 20 days', 'MA for 50 days']].plot(subplots = False, figsize = (10,4))
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x110e49150>



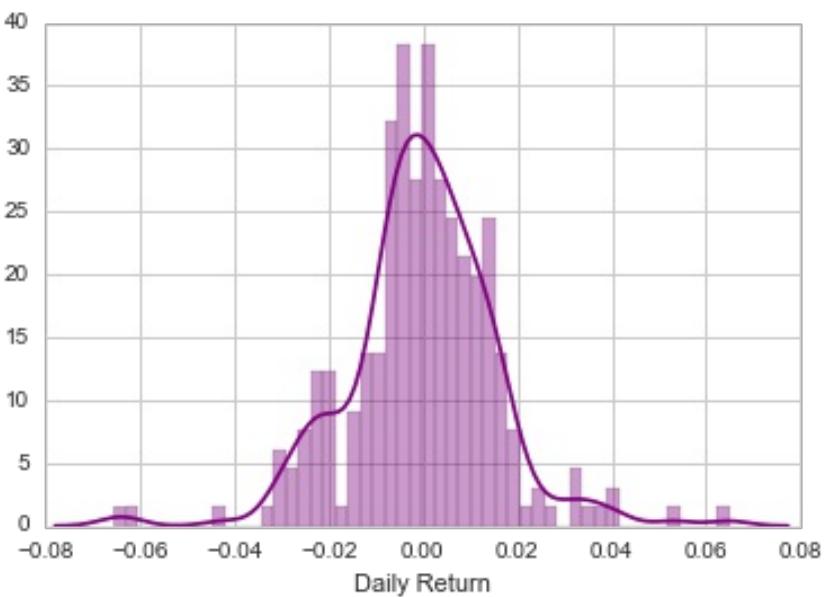
In [32]:

일별 수익을 계산하고 분포를 그린다.

```
AAPL['Daily Return'] = AAPL['Adj Close'].pct_change()  
sns.distplot(AAPL['Daily Return'].dropna(),  
             bins = 50, color = 'purple')
```

Out[32]:

<matplotlib.axes._subplots.AxesSubplot at 0x110c0a6d0>



CURRICULUM

1. 파이썬 소개 및 설치. 변수의 종류와 연산
2. 조건문과 리스트 그리고 반복문
3. 변수와 함수
4. 구글 페이지랭크 및 검색엔진 구축
5. 자료구조의 중요성 1 Dictionary
6. 프로그램의 속도 (해쉬와 재귀함수)
7. 자료구조의 중요성 2 Numpy 와 Pandas



PYTHON 2 VS PYTHON 3

파이썬의 종류



출처: <http://learntocodewith.me/programming/python/python-2-vs-python-3/>

특징:

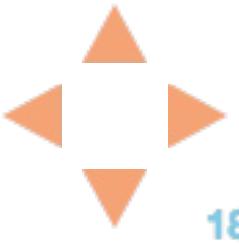
- ④ Python 2: 더 많은 사용자와 그로 인한 다양한 패키지
- ④ python 3: 더 나은 논리구조로 인한 빠른 신기술 적용

In [7]: `### 헌실:`

```
print 'python2 has no bracket'  
print ('python3 has a bracket')
```

python2 has no bracket

python3 has a bracket



결론:

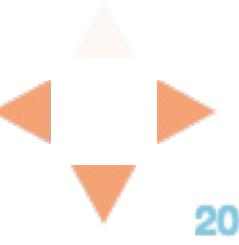
현 상황에서는 다양한 패키지를 제공하는 파이썬2를 사용하는 것이 더 좋은 옵션.



파이썬의 설치

ANACONDA?

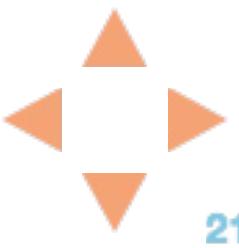
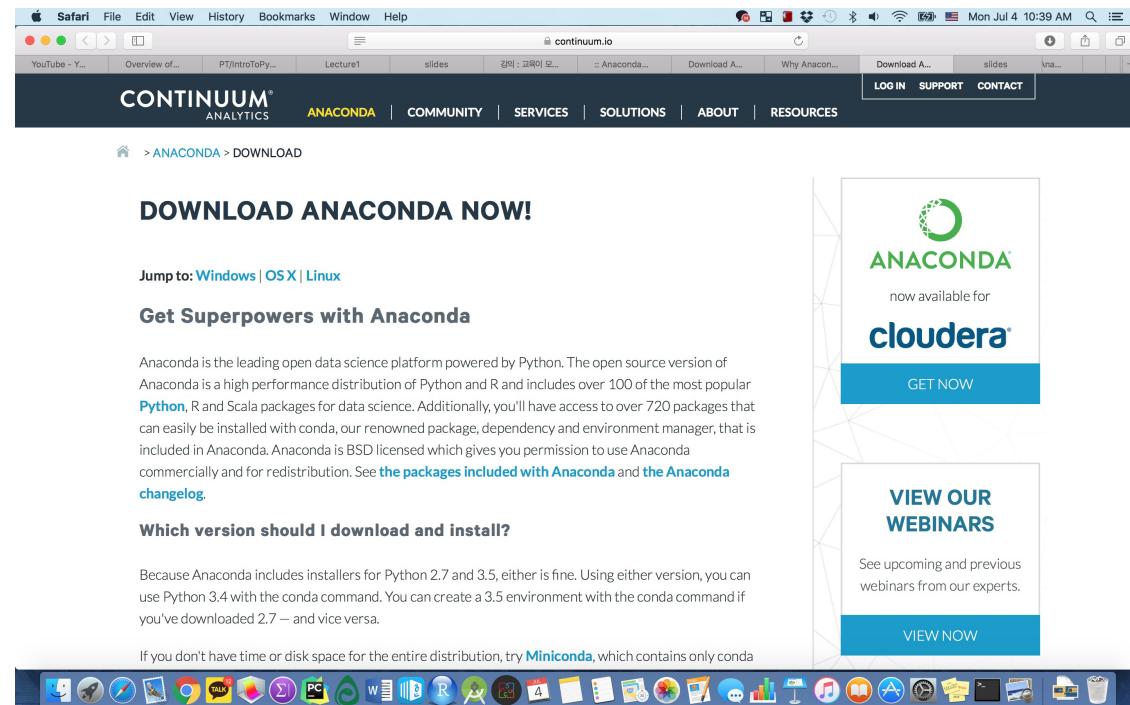
- ① 아나콘다란.. (파이썬 + 사람들이 자주 쓰는 패키지)
- ② 아나콘다란.. 패키지를 쉽게 추가 할 수 있는 패키지 관리 프로그램



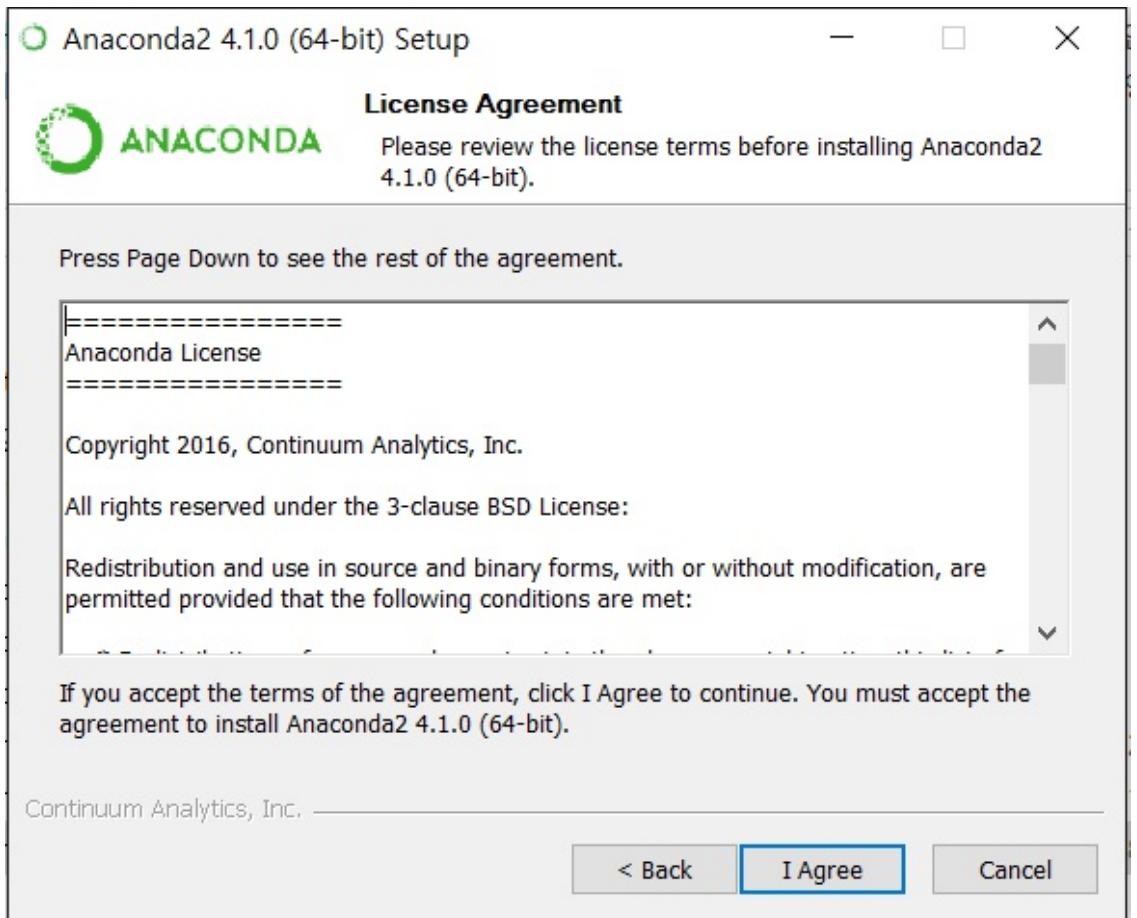
파이썬의 설치

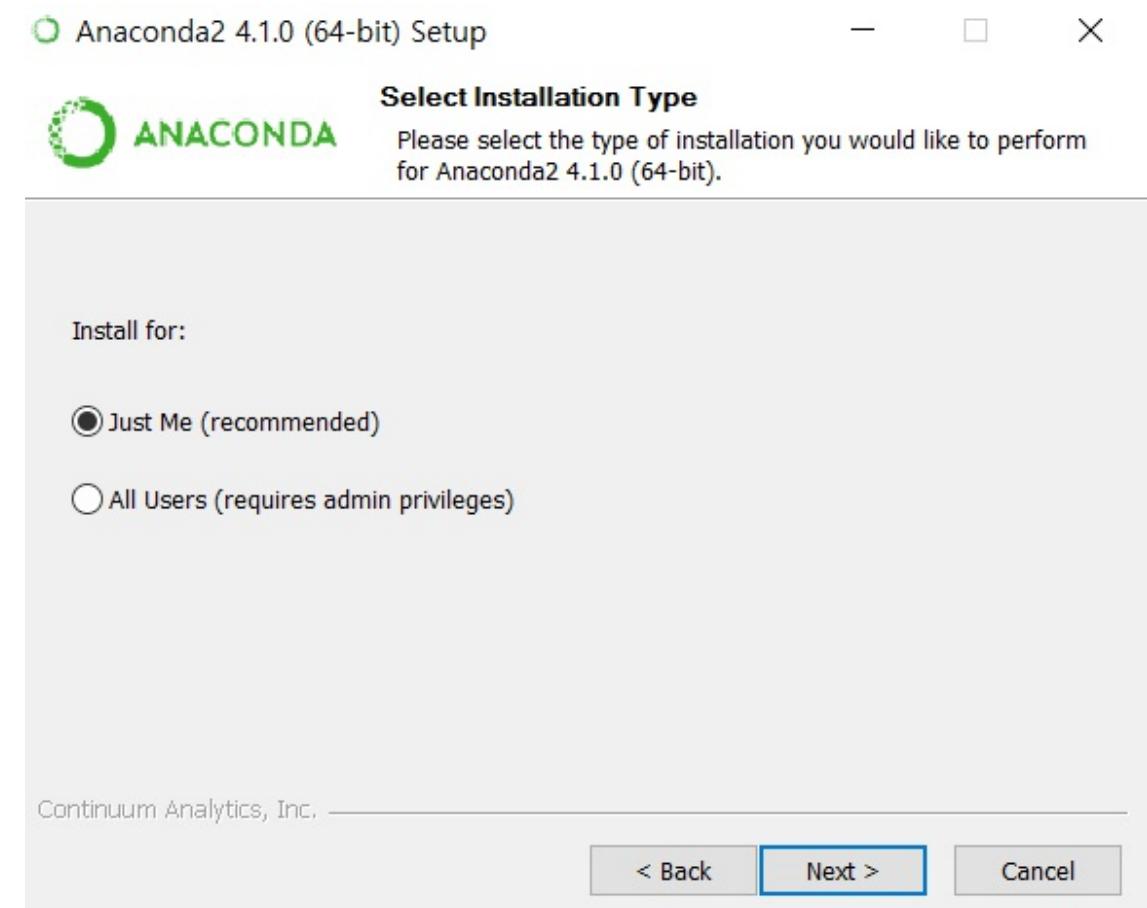
ANACONDA?

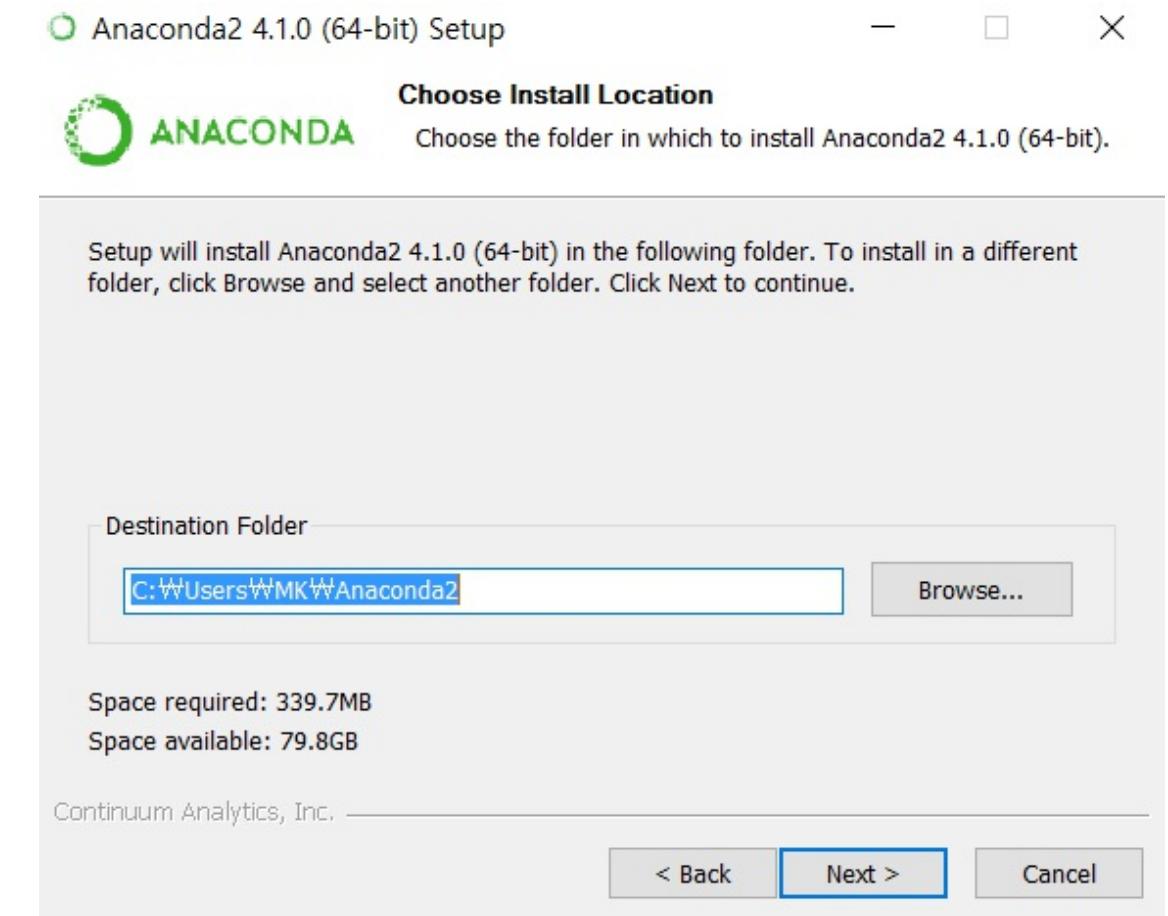
- ④ 구글에 'anaconda' 라고 검색
- ④ 혹은 <https://www.continuum.io/downloads>

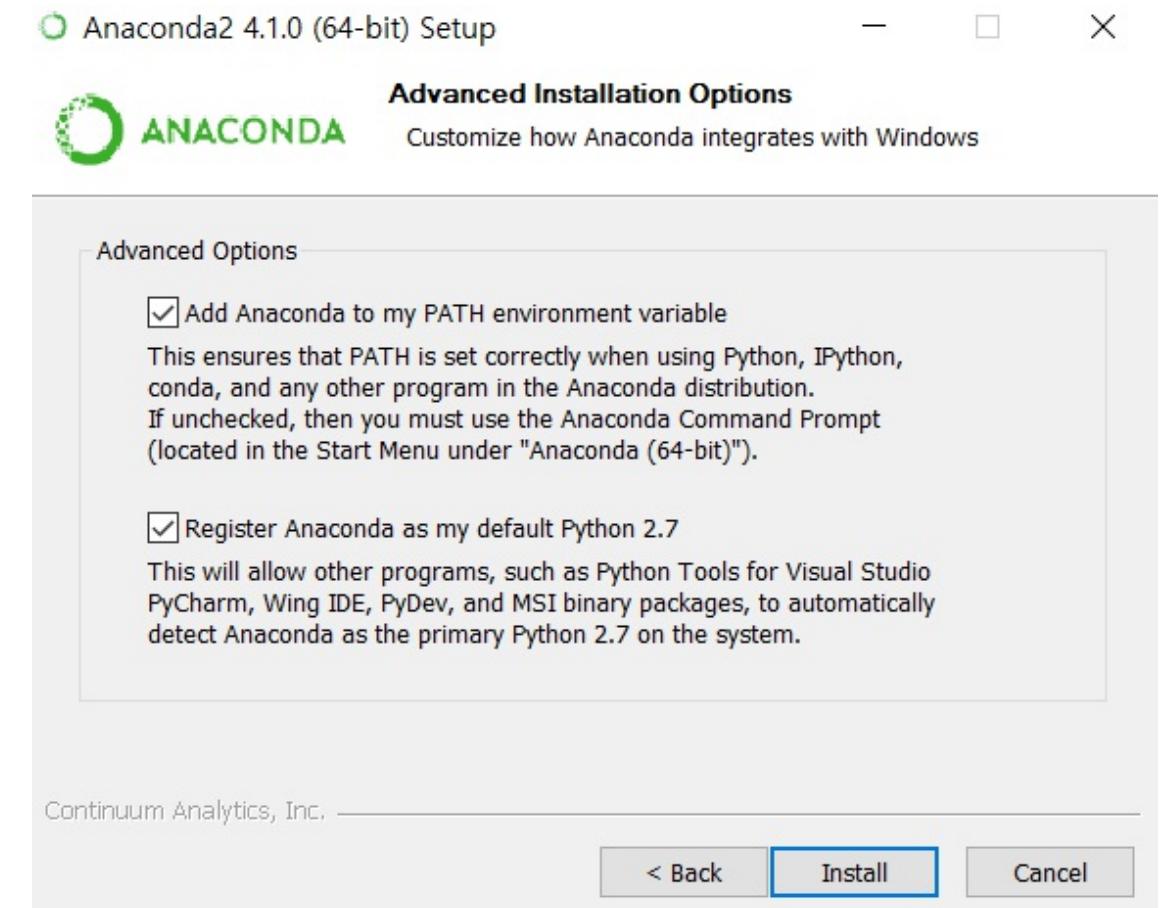


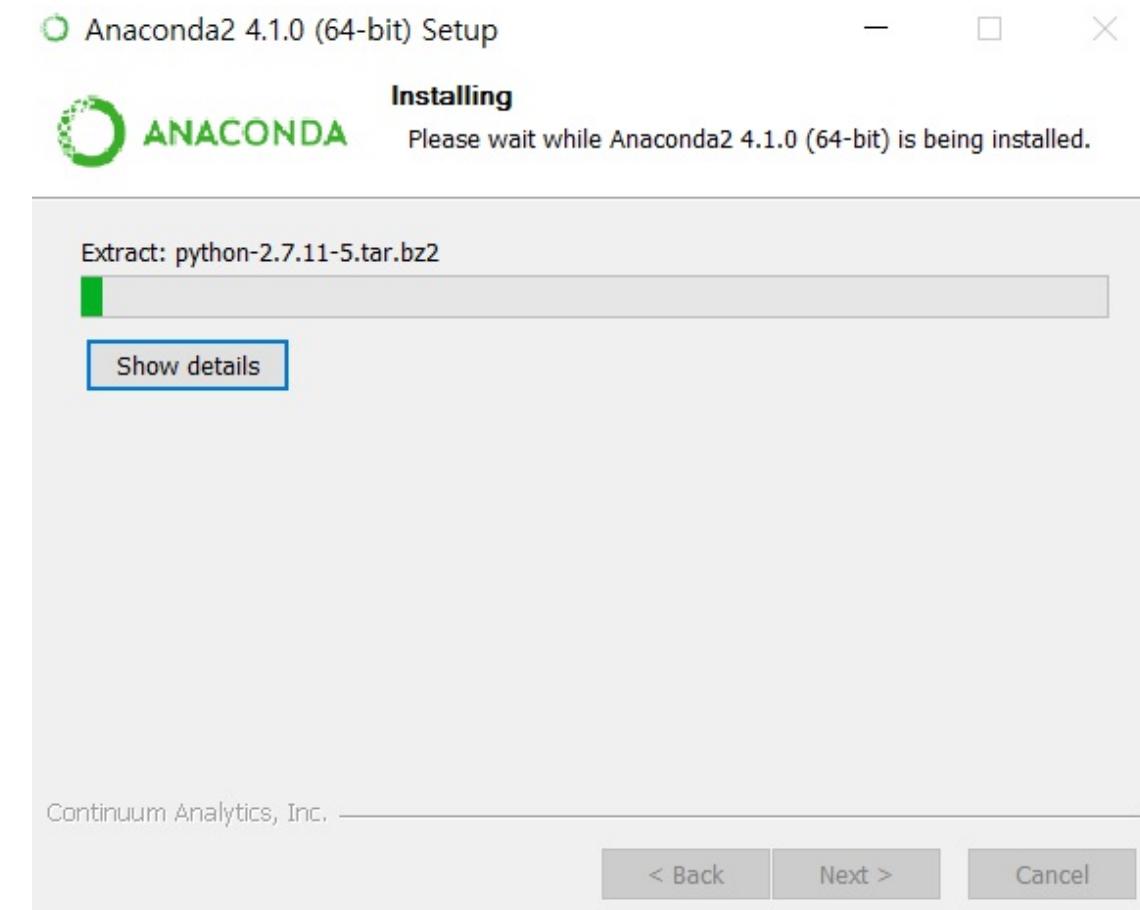












파이썬 실행

WINDOWS:

1. 실행 창에 'cmd'라고 입력하고 enter 를 누른다.
2. 명령 프롬프트창이 뜨면 'python' 이라고 입력하고 enter 를 누른다.

MAC:

1. terminal 을 열어서 'python' 이라고 입력하고 enter를 누른다.



파이썬 IDE

IDE 란:

- ① Integrated development environment의 약자.
- ② 프로그래밍을 할 때 편의를 위해 사용하는 강력한 노트패드

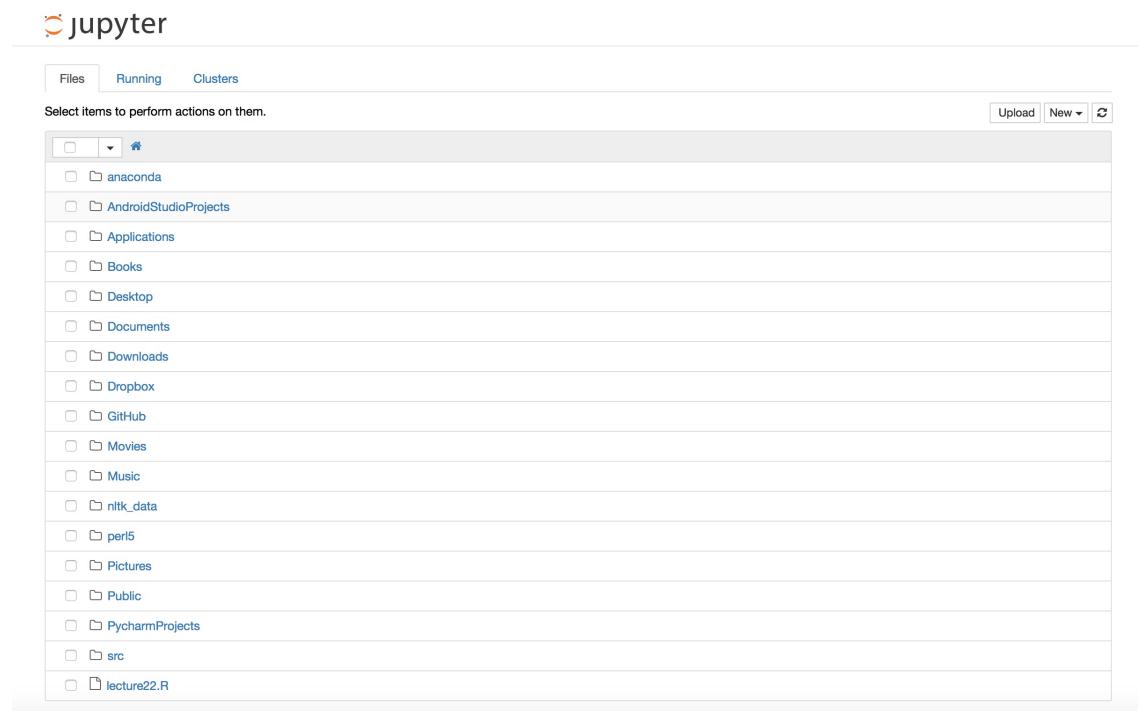
파이썬을 위한 대표적인 IDE:

1. Jupyter notebook
2. PyCharm
3. Sublime Text



파이썬 IDE

JUPYTER NOTEBOOK:



JUPYTER NOTEBOOK 실행

WINDOWS:

1. 실행 창에 'cmd'라고 입력하고 enter 를 누른다.
2. 명령 프롬프트 창이 뜨면 'jupyter notebook'이라고 입력하고 enter 를 누른다.

MAC:

1. terminal 을 열어서 jupyter notebook 이라고 입력하고 enter를 누른다.



HELLO WORLD!!

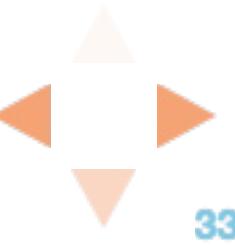
HELLO WORLD!!

```
In [9]: print 'hello world'
```

```
hello world
```

변수의 종류

1. numerical (숫자)
2. string (문자)



변수의 종류

1. numerical (숫자)
2. string (문자)

In [26]:

```
print 3
print 'string'
print 323454545
print '내 이름은 파이썬이다.'
```

```
3
string
323454545
내 이름은 파이썬이다.
```



NUMERICAL 변수

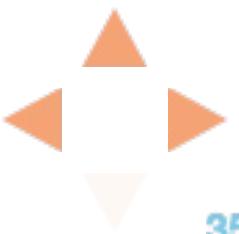
NUMERICAL 변수의 연산

1. + 덧셈
2. - 뺄셈
3. * 곱셈
4. ** 지수
5. % 모듈로

In [27]:

```
print 3+3  
print 3-3  
print 3*4  
print 4**2  
print 13 % 4
```

6
0
12
16
1



NUMERICAL 변수

NUMERICAL 변수의 비교

1. == 같다
2. != 다르다
3. >= 크거나 같다
4. <= 작거나 같다
5. > 크다
6. < 작다

```
In [31]: print 3 == 3  
print 3 != 3  
print 3 <= 4  
print 3 >= 4  
print 3 > 10  
print 3 < 10
```

True

False

True

False

False

True



연습문제:

1. 일주일이 총 몇 시간인지 파이썬으로 구해보라.

In [37]: $7 * 24$

Out[37]: 168

변수의 저장

1. '변수이름 = 저장 될 내용' 의 형태를 갖춘다.
2. 여러개의 변수를 한번에 지정할 수 있다.
3. 변수의 이름은 숫자나 문자로 구성해야 하며 숫자로 시작할 수 없다.
4. 변수는 대소문자가 다르면 다른 이름이다.



In [34]:

```
x = 3
```

```
print x
```

```
y = 34034
```

```
print y
```

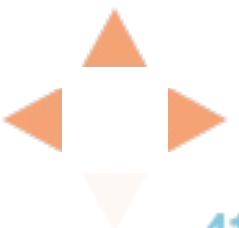
```
x,y = 5,10
```

```
print x,y
```

3

34034

5 10



연습문제:

1. 강의실에 a,b,c,d,e 라는 학생이 있다. a의 몸무게는 50, b는 40, c는 60, d는 80, e는 70 이다. 5명의 총 몸무게를 a,b,c,d,e라는 변수에 저장하여 구하라.

In [38]:

```
a,b,c,d,e = 50,40,60,80,70
```



연습문제:

x의 출력값은 예상해보시오.

In [41]:

```
x=3
```

```
x = x + 1
```

```
x = x
```

In [42]:

```
x=3
```

```
x = x + 1
```

```
x = x
```

```
print x
```

4



연습문제:

x의 출력값은 예상해보시오.

In [43]:

#4

x = 3

a = 4

a,x = x,a

a,x = x,a

In [46]:

```
#4  
x = 3  
a = 4  
a,x = x,a  
a,x = x,a  
print x,a
```

3 4



연습문제:

다음 코드를 실행했을 때 seconds 의 값은?

In []:

```
# minutes = minutes + 1  
# seconds = minutes * 60
```

In [40]:

```
minutes = minutes + 1  
seconds = minutes * 60
```

NameError Traceback (most recent call last)
<ipython-input-40-f69563a137bf> in <module>()
----> 1 minutes = minutes + 1
 2 seconds = minutes * 60

NameError: name 'minutes' is not defined

연습문제:

다음 코드를 실행하였을 때 변수 x의 값은?

In [47]:

```
x = 3  
a = x  
a = a + 1
```

In [51]:

```
x = 3  
a = x  
a = a + 1  
print x
```

3



연습문제:

다음 코드를 실행하였을 때 변수 x의 값은?

In [56]:

```
a = 3  
x = a  
a = 4
```

In [53]:

```
a = 3  
x = a  
a = 4  
print x
```

3



STRING 변수

STRING 변수의 규칙

- ① 모든 문자는 큰 따옴표나 작은 따옴표로 감싸준다.
- ② 중간에 따옴표가 온다면 감싸주는 따옴표는 다른 종류의 따옴표를 사용한다.
- ③ 띄어쓰기도 문자에 포함이 된다.
- ④ 여러줄의 string은 작은따옴표 3개로 묶어줄 수 있다.
- ⑤ 틀린 예: 'sdfs", 'He doesn't like her' , hello



In [41]:

```
print "hello"
print 'hi'
print '"hey!"'
```

hello
hi
"hey!"



연습문제:

- ◉ 다음 중 실행 할 수 없는 형태의 string 변수는?

연습문제:

- ◎ 다음 중 실행 할 수 없는 형태의 string 변수는?

In [42]:

```
#1  
'!!!!'  
#2  
''''''''  
#3  
'''  
#4  
"한글"  
#5  
"string"
```

Out[42]:

```
''''''''
```



STRING

STRING의 저장

1. 숫자변수와 동일한 규칙
2. '변수이름 = 저장 될 내용' 의 형태를 갖춘다.
3. 여러개의 변수를 한번에 지정할 수 있다.
4. 변수의 이름은 숫자나 문자로 구성해야 하며 숫자로 시작할 수 없다.
5. 변수는 대소문자가 다르면 다른 이름이다.



In [1]:

```
#예)  
name1 = 'Iron man'  
name2 = 'Hulk'  
name3 = 'Black Widow'  
name4 = '헐크'  
name5 = '아이언맨'  
print name1
```

Iron man



STRING 변수

STRING 변수의 연산

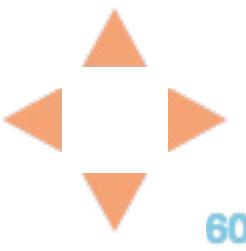
- ④ + (0|어주기)
- ④ * (반복)

In [54]:

```
##예)  
name1 = '남상대'  
name2 = '민하은'  
name3 = '이연주'  
name4 = '윤병길'  
name5 = '김민철'  
print name1 + name2 + name3 + name4 + name5  
  
print name2*3
```

남상대민하은이연주윤병길김민철

민하은민하은민하은



In [53]: `##예)`

`name1 + 9`

`name1 + '9'`

`TypeError`

Traceback (most recent call last)

`<ipython-input-53-ff4fb504c8cf> in <module>()`

`1 ##예)`

`----> 2 name1 + 9`

`3`

`4 name1 + '9'`

`TypeError: cannot concatenate 'str' and 'int' objects`



STRING 변수

STRING 의 INDEX

STRING 변수

STRING 의 INDEX

In [59]:

```
name = 'George'  
print name[0]  
print name[1:3]  
print name[:]  
print name[:3]  
print name[3:]
```

G

eo

George

Geo

rge



STRING 변수

STRING 내 찾기 (FIND)

- ① find 함수는 string변수 뒤에 따옴표를 붙이고 사용해준다.
- ② ex) name.find('G') - name이라는 변수가 저장하고 있는 string 안에서 'G'의 위치를 검색
- ③ 검색에 실패하면 '-1'을 출력한다.



In [35]:

```
text = "a program that searches for and identifies items in a database that  
correspond to keywords or characters specified by the user,  
used especially for finding particular sites on the World Wide Web."
```

```
print text.find('a')  
print text.find('program')  
print text.find('words')  
print text.find('gesdf')
```

```
0  
2  
86  
-1
```

In [54]: #처음 in 0/ 나오는 위치:

```
text = "a program that searches for and identifies items in a database that  
correspond to keywords or characters specified by the user,  
used especially for finding particular sites on the World Wide Web."
```

```
first = text.find('in', 0)  
print first
```

49



연습문제:

In []:

아래 글에서 *in*이 두번째, 세번째나오는 위치를 찾아보라.

text = "a program that searches for and identifies items in a database that correspond to keywords or characters specified by the user, used especially for finding particular sites on the World Wide Web."



In [55]:

아래 글에서 in 이 두번째, 세번째나오는 위치를 찾아보라.

```
text = "a program that searches for and identifies items in a database that  
correspond to keywords or characters specified by the user,  
used especially for finding particular sites on the World Wide Web."
```

```
first = text.find('in', 0)
```

```
print first
```

```
second = text.find('in', first+1 )
```

```
print second
```

```
third = text.find('in',second+1)
```

```
print third
```

49

151

154



STRING 변수

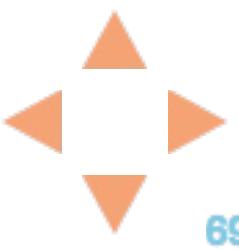
find 를 이용하여 url을 찾아보자.

url은 웹 페이지 주소를 말하며 <a href= 다음에 따옴표 사이에 나오는 형식으로 되어있다.

In [2]:

```
#이것은 웹페이지의 소스코드이다.  
page = "<body>  
    <div id='topContainer'>  
        <div id='topLeft'>  
            <ul>  
                <li><a href='http://www.facebook.com'>What If?</a></li>  
                <li><a href='http://www.google.com'>Blag</a></li>  
                <li><a href='http://www.twitter.com/'>Store</a></li>  
                <li><a rel='author' href='/about'>About</a></li>  
            </ul>"  
  
# 첫번째 URL을 뽑아내는 과정이다.  
start_link = page.find('<a href=',0)  
start_url = page.find("'", start_link + 1)  
end_url = page.find("'", start_url + 1)  
print page[start_url:end_url]
```

"http://www.facebook.com



In [3]:

```
# 두번째 URL을 뽑아내는 과정이다.  
second_link = page.find('<a href=',start_link +1)  
start_url = page.find(" ", second_link + 1)  
end_url = page.find(" ", start_url + 1)  
print page[start_url:end_url]
```

"http://www.google.com



In [1]:

주석 (*comment*)

주석은 컴퓨터가 읽지 않고 사람만 읽기 위해 쓰는 글이다.

샵을 이용하여 작성한다.

2.1 BOOLEAN 값

TRUE & FALSE

불리안은 참이나 거짓 중 하나를 값으로 가지는 데이터.

In [2]:

```
a = True
```

```
b = False
```

```
c = "True"
```

```
print "a 는 ", a
```

```
print "b 는 ", b
```

```
print "c 는 ", c
```

a 는 True

b 는 False

c 는 True



```
In [3]: print "a 의 타입: ", type(a)  
print "c 의 타입: ", type(c)
```

a 의 타입: <type 'bool'>
c 의 타입: <type 'str'>



비교 연산자

1. `==` 같다
2. `!=` 다르다
3. `>=` 크거나 같다
4. `<=` 작거나 같다
5. `>` 크다
6. `<` 작다

In [4]:

```
# 예제: 다음 코드에서 a의 결과값은 어떤 것이 되겠는가?  
# * 변수를 지정해주는 = 연산자가 가장 마지막에 처리된다.  
a = 1 == 2
```

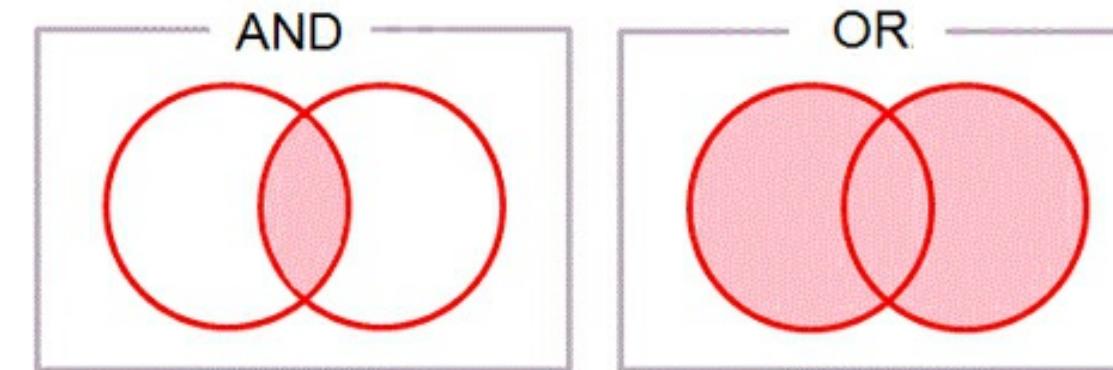
```
In [5]: a = 1 == 2  
       print a
```

False



불리언 연산자

1. and : 두 구문 모두 참일 경우 True, 아니면 False
2. or : 두 구문 중 하나라도 참인 경우 True, 둘 다 거짓이면 False
3. not : 구문이 참이면 거짓, 거짓이면 참 으로 값을 뒤집는다.



In [6]: *### 다음 예를 살펴보자.*

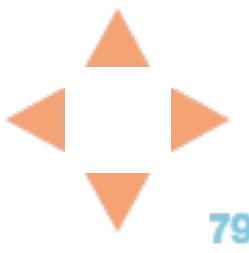
```
# and 연산자  
print True and True  
print True and False  
print False and True  
print False and False
```

True

False

False

False



In [7]: *# or 연산자*

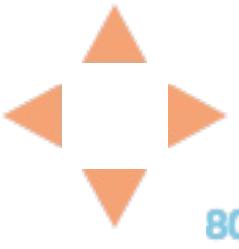
```
print True or True  
print True or False  
print False or True  
print False or False
```

True

True

True

False

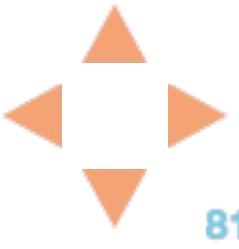


In [8]: *# not 연산자*

```
print not True  
print not False
```

False

True



In [9]:

```
# 다음 예를 풀어보시오.  
#  $37 > 29$   
#  $37 < 40$   
#  $37 > 29$  and  $37 < 40$   
#  $37 > 29$  or  $30 < 40$   
# not  $37 > 29$ 
```

In [10]:

```
# 다음 예를 풀어보시오.  
print 37 > 29  
print 37 < 40  
print 37 > 29 and 37 < 40  
print 37 > 29 or 30 < 40  
print not 37 > 29
```

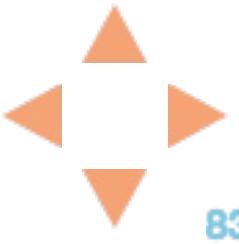
True

True

True

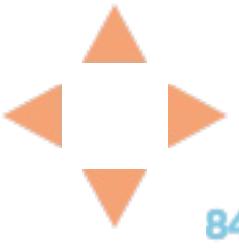
True

False



불리안 연산자의 순서

1. not
2. and
3. or



In [11]: #예를 들어

True and False or not False

Out[11]: True

제일 먼저 `not False` 가 평가되어 `True` 가 되고

다음으로 맨 앞의 `True and False` 를 평가하여 `False`가 되고

그리고 이 값을 `True` 와 `or` 로 연결하여 `False or True`, 즉 `True` 가 된다.



In [12]:

다음 문제를 풀어보시오

#1.

#True or False and not False

#2.

#not False or True and not True



```
In [13]: print True or False and not False  
print not False or True and not True
```

True

True



2.2 조건문

if 구문은 불리언 표현식을 계산하여 값이 True 일 경우 하위 코드를 수행한다.

In [14]: #예

```
if 37 < 40:  
    print '서른 일곱은 마흔 보다 작다.'
```

서른 일곱은 마흔 보다 작다.

여기서 중요한 것은

1. if 와 : 사이에 평가 될 조건이 온다.
2. : 뒤에 들여 쓴 코드가 온다. 이를 코드 블록 이라고 한다.

In [15]: **if** 37 < 40:

print '첫번째 코드블록'

if 40 > 30:

print '들여쓴 첫번째 코드블록'

if 50 > 20:

print '들여쓴 두번째 코드블록'

if 50 > 30:

print '두번째 코드블록'

첫번째 코드블록

들여쓴 첫번째 코드블록

들여쓴 두번째 코드블록

두번째 코드블록

```
In [16]: #들여쓰기 예제  
if 37 < 40:  
    print 'hello'
```

```
File "<ipython-input-16-2607a0dc2ed5>", line 3  
    print 'hello'  
          ^
```

IndentationError: expected an indented block



IF & ELSE

if구문은 else와 함께 사용할 수 있다.

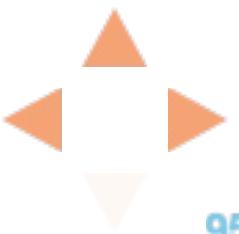
Else 아래에서 들여써진 코드는 if 구문이 거짓일 때 실행된다.

다음 예를 살펴보자.

```
In [17]: age = 30
if age >= 35:
    print "대통령이 되기에 충분한 나이입니다."
else:
    print "대통령이 되기에 충분한 나이가 아닙니다"
print 'have a good day'
```

대통령이 되기에 충분한 나이가 아닙니다

have a good day



IF & ELSE & ELSEIF

"아니라면 만약"이라는 의미의 else if 를 줄인 elif를 사용하면 여러개의 조건문을 사용할 수 있다. 다음의 예를 살펴보자

In [18]:

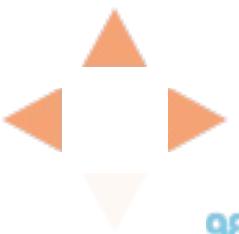
```
age = 31
if age >= 35:
    print '상의원이나 대통령이 되기에 충분한 나이입니다'
elif age >= 30:
    print '상원 의원이 되기에 충분한 나이입니다'
else:
    print "상원의원이나 대통령이 되기에 충분한 나이가 아닙니다."
```

상원 의원이 되기에 충분한 나이입니다

In [19]: *## if 나 elif 의 어느 조건문이라도 참으로 평가되면 남은 elif나 else 블록은 아무것도 실행되지 않는다.*

```
age = 99
if age >= 35:
    print '상원의원이나 대변인 혹은 대통령이 되기에 충분한 나이입니다.'
elif age >= 30:
    print '상원의원이 되기에 충분한 나이입니다.'
elif age >= 25:
    print '대변인이 되기에 충분한 나이입니다'
else:
    print "상원의원이나 대변인 혹은 대통령이 되기에 충분한 나이가 아닙니다."
```

상원의원이나 대변인 혹은 대통령이 되기에 충분한 나이입니다.

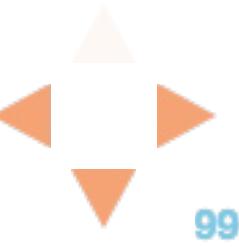


연습문제

사용자에게 얼마나 멀리 여행하고 싶은지 물어보는 프로그램을 만들어라. 3마일 이하로 대답한다면 '걸으세요' 라고 말하고 3마일보다 멀고 300 마일보다 작다면 '차로 가세요' 라고 말해야 한다. 만일 300마일 이상 간다면 "비행기로 가세요" 라고 대답해야 한다.

힌트: `input` 함수를 이용하면 사용자에게 입력을 요구 할 수 있다.

```
distance = input('몇 마일 떨어진 곳으로 여행하고 싶은가요?')
```

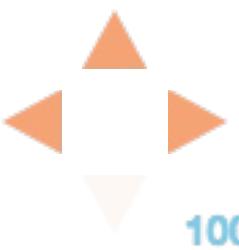


In [21]:

```
#거리 물어보기
distance = input("몇 마일 떨어진 곳으로 여행하고 싶은가요?")
#거리를 정수로 변환
distance = int(distance)
#어떻게 이동할지를 결정
if distance <3:
    mode_of_transport = '도보'
elif distance < 300:
    mode_of_transport = "차"
else:
    mode_of_transport = "비행기"
print "목적지까지", mode_of_transport, '로 가는 것이 좋겠어요.'
```

몇 마일 떨어진 곳으로 여행하고 싶은가요?30

목적지까지 차로 가는 것이 좋겠어요.



숙제 1

아래 코드를 완성하여 url을 출력해보라.

숙제 1

아래 코드를 완성하여 url을 출력해보라.

```
In [ ]: page = "<div id='top_bin'> <div id='top_content' class='width960'>
          <div class='udacity float-left'> <a href='http://naver.com'>What If?</a></li>""

start_link =
star_url =
end_url =
print page[start_url:end_url]
```



숙제 2

아래 코드에서 3번째 url 즉 www.twitter.com 의 위치를 찾아서 출력해보라.

숙제 2

아래 코드에서 3번째 url 즉 www.twitter.com 의 위치를 찾아서 출력해보라.

In [23]:

```
page = "<body>
<div id='topContainer'>
<div id='topLeft'>
<ul>
<li><a href='http://www.facebook.com'>What If?</a></li>
<li><a href='http://www.google.com'>Blag</a></li>
<li><a href='http://www.twitter.com/'>Store</a></li>
<li><a rel='author' href='/about'>About</a></li>
</ul>"
```

숙제 3 : 조건 구문

1.

어떤 가게에서 할인 행사를 진행 중입니다. 가격이 10만원 이하이면 10% 할인을 하고, 10만원 이상이면 20% 할인을 합니다. 사용자에게 구매가격을 물고 할인율 (10% 나 20%) 과 최종 가격을 보여주는 구문을 작성하시오

힌트:

사용자에게 무엇을 물어보고 답을 얻고 싶다면?

input 함수를 이용하면 사용자에게 입력을 요구 할 수 있다.

ex) `x = input('구매가격이 무엇인가요?')`

2.

어떤 축구팀에서 팀에 참가할 10살에서 12살에 해당하는 소녀를 모집하는 중입니다. 사용자의 나이와 사용자가 남자인지 여자인지 ('m'이나 'f'를 사용) 묻는 프로그램을 작성하시오. 사용자가 축구팀에 참가할 수 있는지를 나타내는 메시지를 출력합니다. (보너스: 사용자가 여자가 아닐 경우 나이를 묻지 않도록 만들어 보시오.)