











Python 기초

멋쟁이사자처럼 9기 양소열

Python 이란

전세계에서 가장 인기 많은 언어이고,
문법이 정말 간결하고, 배우기 쉽습니다.

May 2022	May 2021	Change	Programming Language		Ratings	Change
1	2	▲		Python	12.74%	+0.86%
2	1	▼		C	11.59%	-1.80%
3	3			Java	10.99%	-0.74%
4	4			C++	8.83%	+1.01%
5	5			C#	6.39%	+1.98%
6	6			Visual Basic	5.86%	+1.85%
7	7			JavaScript	2.12%	-0.33%
8	8			Assembly language	1.92%	-0.51%
9	10	▲		SQL	1.87%	+0.16%
10	9	▼		PHP	1.52%	-0.34%

Python 이란

이번 세션에는 간단한 게임을 만들어보면서
파이썬 짝먹을 해 볼 예정입니다

부담 없이 편하게 들어 주시면 좋겠습니다 😊

(백엔드 트랙분들은 파이썬 공부 열심히 해야 합니다,,!)

이번 세션에서 만들어 볼 프로그램

Up & Down 게임

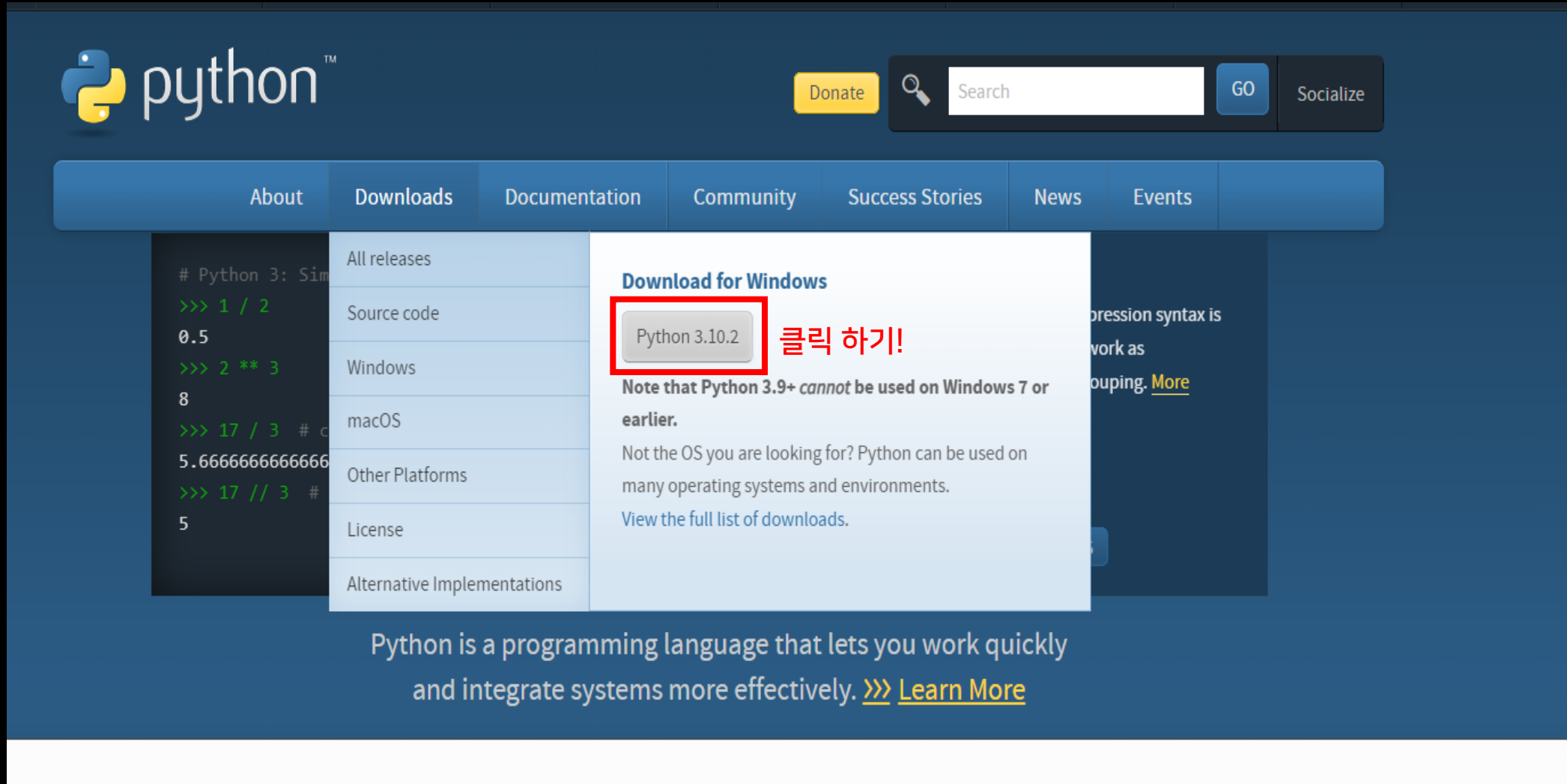
```
1 import random
2
3 randNum = random.randrange(1, 101) # 랜덤한 숫자를 컴퓨터가 하나 뽑는다.
4
5 # 맞출 기회를 10번으로 조정한다.
6 for i in range(10):
7     # 사용자로부터 숫자를 입력 받는다.
8     userNum = int(input('1부터 100사이의 난수를 맞추세요. '))
9
10    # 만약 랜덤한 숫자와 사용자가 입력한 숫자와 같으면 정답을 출력하고 종료
11    if randNum == userNum:
12        print('정답입니다.')
13        break
14
15    # 만약 횟수가 10번이 아니고 9번까지는 힌트를 주면서
16    # 기회를 다시 부여한다.
17    elif i < 9:
18        print('틀렸습니다. 다시 입력하세요.')
19        if randNum > userNum:
20            print('난수는 입력한 숫자보다 큼니다.')
21        else:
22            print('난수는 입력한 숫자보다 작습니다.')
23
24    # 10번 기회를 다 쓰면 게임에서 패배
25    else:
26        print('게임에서 졌습니다.')
27        break
28
29 print('난수 : ', randNum)
```

1부터 100사이의 난수를 맞추세요 . 50
틀렸습니다 . 다시 입력하세요 .
난수는 입력한 숫자보다 작습니다 .
1부터 100사이의 난수를 맞추세요 . 35
틀렸습니다 . 다시 입력하세요 .
난수는 입력한 숫자보다 큼니다 .
1부터 100사이의 난수를 맞추세요 . 40
틀렸습니다 . 다시 입력하세요 .
난수는 입력한 숫자보다 큼니다 .
1부터 100사이의 난수를 맞추세요 . 45
정답입니다 .
난수 : 45

Python 환경설정

<https://www.python.org/>

Python 환경설정



The screenshot shows the Python.org homepage. The 'Downloads' menu is open, listing various options. The 'Windows' option is highlighted, and a sub-menu is displayed showing the download link for 'Python 3.10.2'. A red box highlights this link, and a red text label '클릭 하기!' (Click here!) points to it. The page also features a search bar, a 'Donate' button, and social media links.

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

All releases
Source code
Windows
macOS
Other Platforms
License
Alternative Implementations

Download for Windows

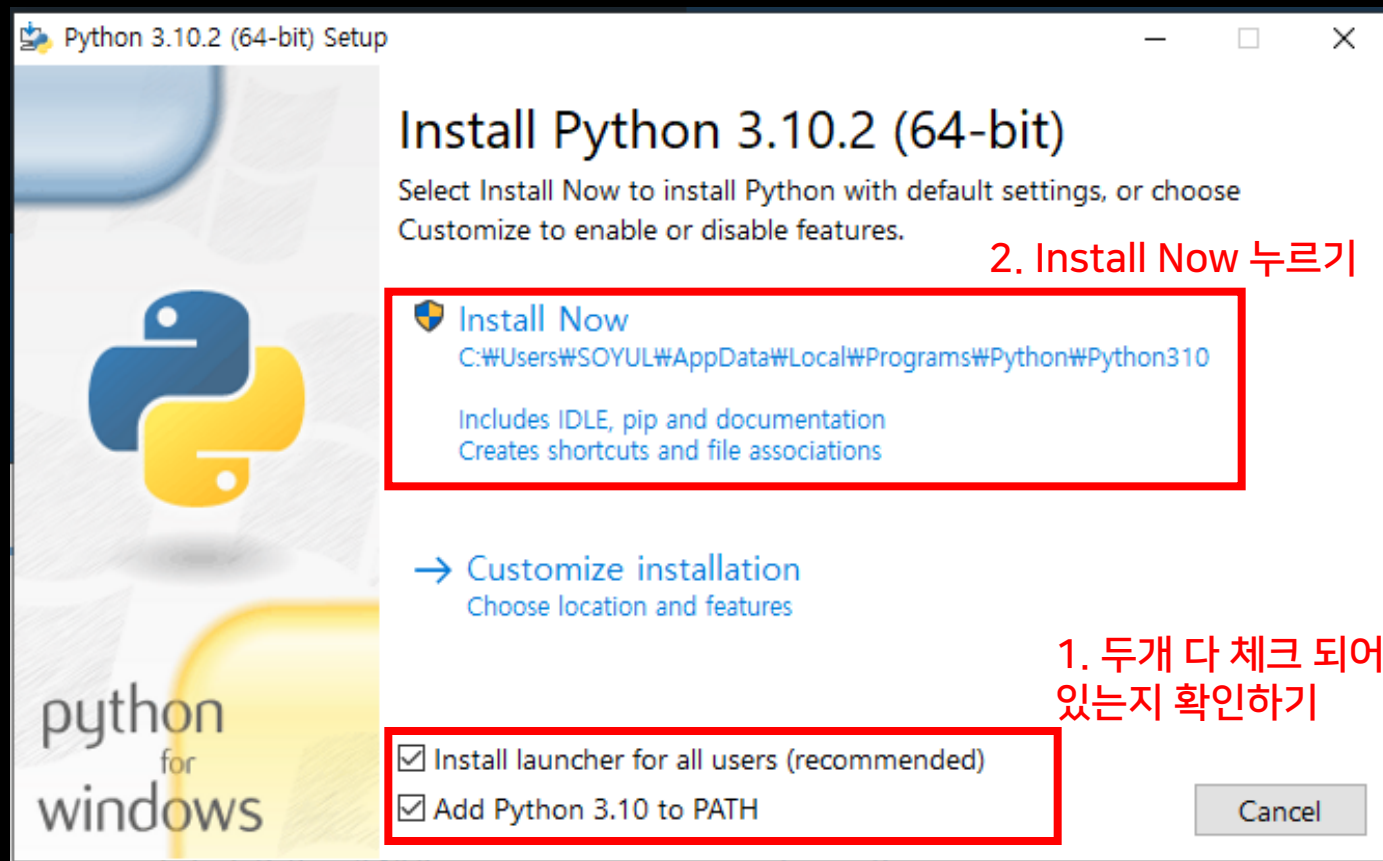
Python 3.10.2 **클릭 하기!**

Note that Python 3.9+ *cannot* be used on Windows 7 or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.
[View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Python 환경설정

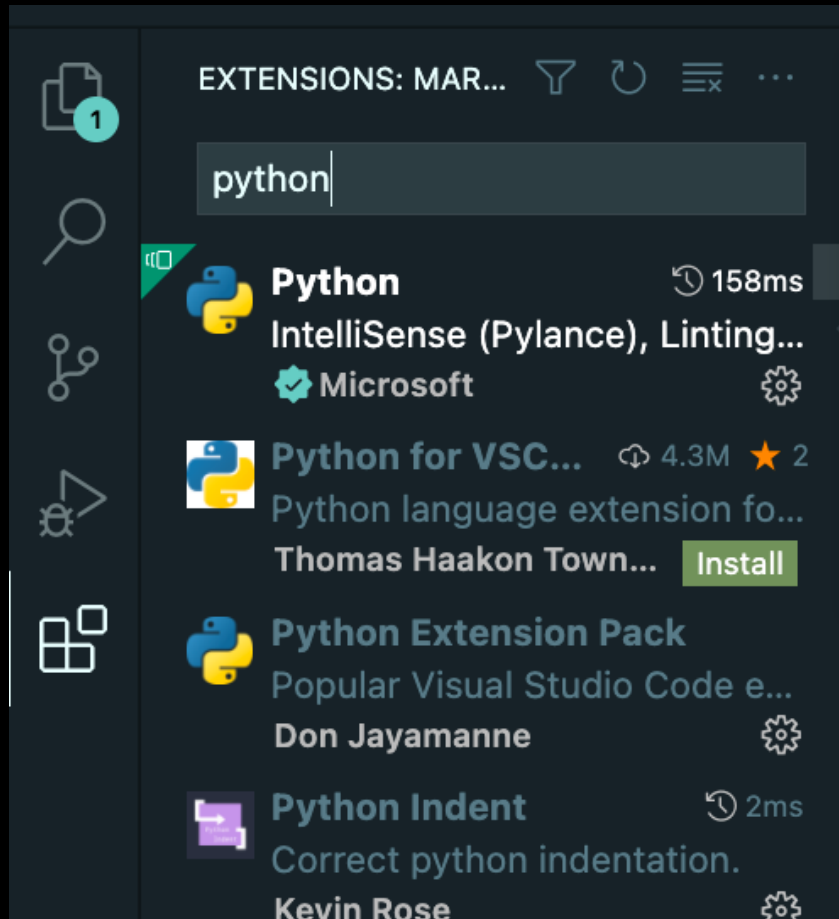


Python 환경설정 - mac

<https://hbase.tistory.com/106>

파이참 설치 전까지 따라하기!!

Python Extension 설치



1. 블록 같은 아이콘을 **클릭**합니다.
2. Python을 검색하고 확장 프로그램을 설치하여 줍니다.

print()

()안에 있는 값을 **출력** 해줍니다

입력

```
print("✅ UP & DOWN 게임 시작")
```

출력

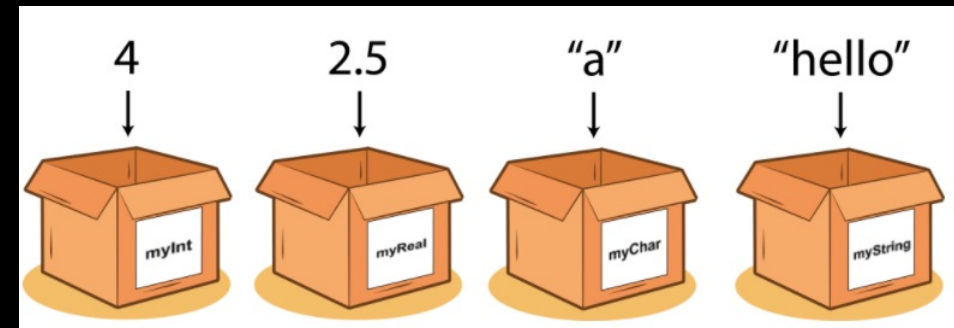
```
✅ UP & DOWN 게임 시작
```

변수

값을 저장할 수 있는 공간

용기에 데이터를 저장하는 개념

변수이름 = "무엇을 저장할 것인지"



입력

```
a = 20  
b = "멋사"  
c = True
```

```
print(a)  
print(b)  
print(c)
```

출력

```
20  
멋사  
True
```

리스트

변수에 하나의 값이 아니라,
여러 개의 값을 저장할 때 사용합니다

다른 프로그래밍 언어에서는 배열로 자주 부릅니다

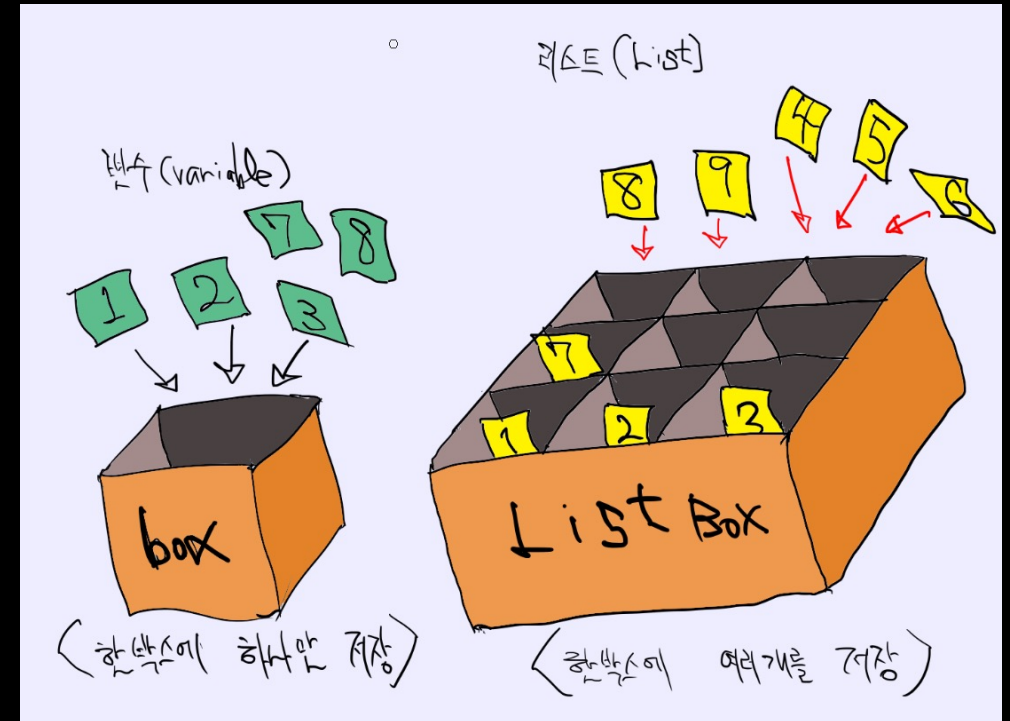
리스트 이름 = []

입력

```
a = [1, '멋사', False, 2, 3, 4]
print(a)
```

출력

```
[1, '멋사', False, 2, 3, 4]
```



input('요청 하고 싶은 내용')

사용자로부터 무언가 **입력**을 받습니다.

```
userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))
```

1. input 함수로부터 사용자에게 **입력**을 받습니다.
2. int()를 통해 **입력 받은 것을 숫자형**으로 바꿔줍니다.
3. 숫자형으로 바꾼 것을 **userNum이라는 변수에 저장**합니다.

Python 모듈

모듈이란? 함수, 변수 또는 클래스를 모아 놓은 파일

한마디로 남이 만들어 놓은 각종 기능이나, 함수를 불러오는 것입니다.

import 파일이름

-> 해당 파일에서 함수와 클래스를 모두 불러온다.

from 파일이름 **import** divide

-> 해당 파일에서 divide함수(클래스)만 불러온다.

<예시>

import random

-> python 개발자가 만든 random 함수를 사용하겠다.

random

파이썬에서 기본으로 제공하는 함수입니다.

랜덤으로 임의의 수를 하나 뽑을 때 사용합니다.

```
import random  
randNum = random.randrange(1,101)
```

1. 누군가 만든 random 모듈을 불러옵니다.(import random)
2. 1부터 100 사이의 랜덤한 수를 뽑습니다.(random.randrange)
3. 뽑은 수를 randNum 변수에 저장합니다.(randNum = random ~~)

Python 조건문 - if

주어진 **조건을 판별**하고, **참/거짓**일때 각각 다른 동작을 수행합니다.

Python은 들여쓰기가 **매우매우매우** 중요합니다

If 조건:

참 일 때 수행 할 동작

else:

거짓 일 때 수행 할 동작

입력

```
score = 90
If score > 50:
    print("합격")
else:
    print("불합격")
```

출력

합격

Python 조건문 - elif

앞선 조건이 거짓일때 또다른 조건을 더 줄 수 있습니다.

elif조건은 원하는 만큼 만들 수 있습니다.

If 조건1:

참 일때 수행 할 동작

elif 조건2:

조건1이 거짓이고, 조건2가 참일때 수행 할 동작

elif 조건3:

앞선 조건이 다 거짓이고, 조건3이 참일때 수행 할 동작

else:

앞선 조건이 모두 다 거짓일때(이도저도 아닐 때)

입력

```
score = 60
If score > 70:
    print("A+")
elif score > 50:
    print("B+")
else:
    print("과락")
```

출력

B+

Python 조건문 - if

```
if randNum == userNum:  
    print('정답입니다')  
else:  
    print('틀렸습니다. 다시 입력해주세요')
```

만약 사용자가 입력한 값 이랑 랜덤으로 뽑은 값이랑 같으면 '정답입니다' 출력
만약 사용자가 입력한 값 이랑 랜덤으로 뽑은 값이랑 다르면 '틀렸습니다' 출력

Python 조건문 - for

어떤 동작을 계속해서 **반복할 때** 사용한다.

for i in range(시작, 끝)
반복할 동작

=> 시작 ~ 끝 - 1 범위만큼 동작을 반복한다.

for i in range(n)
반복할 동작

=> n번 만큼 해당 동작을 반복한다.

입력

```
for i in range(5):  
    print(i, ", 안녕하세요")
```

출력

```
0, 안녕하세요  
1, 안녕하세요  
2, 안녕하세요  
3, 안녕하세요  
4, 안녕하세요
```

Python 조건문 - while

for문 처럼 어떤 동작을 반복해서 수행한다.

for문이란 약간 다른점은, for문은 반복횟수가 정해져 있음.

while문은 작성한 조건이 참인 동안에는 계속 반복문을 수행

while 조건문:

반복할 동작

=> 해당 조건이 참일때까지 동작을 반복수행.

while True:

반복할 동작

=> 무한반복, 빠져나오고 싶을때는 동작에 break를 추가함.

ex) While True:

```
print("이 문장은 1번만 실행!")  
break
```

입력

```
count = 0  
while count < 5:  
    print(count, "안녕하세요")  
    count = count + 1
```

출력

```
0, 안녕하세요  
1, 안녕하세요  
2, 안녕하세요  
3, 안녕하세요  
4, 안녕하세요
```

프로그램 소스코드 추가

```
for i in range(10):  
    userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))  
    if randNum == userNum:  
        print('정답입니다.')  
    else:  
        print('난수는 입력한 숫자보다 작습니다.')
```

1. 아까 만든 프로그램에 **for문을 넣어서 횟수를 10번으로** 고정시켜줍니다.
2. 이때 i는 한번 실행할 때마다 0 -> 1-> 2 -> 3 ~~~ -> 9 까지 변합니다.

프로그램 소스코드 추가

```
for i in range(10):
    userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))
    if randNum == userNum:
        print('정답입니다.')
    elif i < 9:
        print('틀렸습니다. 다시 입력해주세요.')
    else:
        print('게임에서 졌습니다.')
```

1. 2가지 케이스를 3가지 경우의 수로 나누어 줍니다.
2. 1) 사용자가 정답을 맞췄을 때 (randNum == userNum)
2) 9번의 기회에서 정답을 틀릴 때 (i < 9)
3) 10번 시도했지만 정답을 틀릴 경우(패배)

프로그램 소스코드 추가

```
for i in range(10):
    userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))
    if randNum == userNum:
        print('정답입니다.')
    elif i < 9:
        print('틀렸습니다. 다시 입력해주세요.')
        if randNum > userNum:
            print('난수는 입력한 숫자보다 큽니다.')
        else:
            print('난수는 입력한 숫자보다 작습니다.')
    else:
        print('게임에서 졌습니다.')
```

1. 9번의 기회에서 틀렸을 때, **사용자에게 힌트를 제공**하기 위해서 2가지의 케이스로 쪼갭니다.
 - 1) 사용자가 **입력한 값이 랜덤 한 값보다 클 때** '더 크다'라고 출력
 - 2) 사용자가 **입력한 값이 랜덤 한 값보다 작을 때** '더 작다'라고 출력

break

while이나 for문같은 반복문을 **바로!! 탈출하게** 해줍니다.

예를 들어 **for i in range(100)**같은 코드가 있으면, 원래는 코드 100번을 실행 시키지만, 조건문을 주고 해당 조건에 맞을 때 **break**를 걸어준다면 100번을 실행하지 않고 **반복문을 즉시 종료**합니다.

입력

```
count = 0
while count < 5:
    if count == 2:
        break
    print(count, "안녕하세요")
    count = count + 1
```

출력

```
0, 안녕하세요
1, 안녕하세요
```

프로그램 소스코드 추가

```
for i in range(10):
    userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))
    if randNum == userNum:
        print('정답입니다.')
        break
    elif i < 9:
        print('틀렸습니다. 다시 입력해주세요.')
        if randNum > userNum:
            print('난수는 입력한 숫자보다 큽니다.')
        else:
            print('난수는 입력한 숫자보다 작습니다.')
    else:
        print('게임에서 졌습니다.')
        break
```

1. `break` 문장을 추가하여, 정답일 때와 기회를 다 썼을 때 강제종료 시켜줍니다

프로그램 완성

```
import random

print('✅ UP & DOWN 게임 시작')
randNum = random.randrange(1, 101)

for i in range(10):
    userNum = int(input('1부터 100사이의 숫자를 맞추세요. '))
    if randNum == userNum:
        print('정답입니다.')
        break
    elif i < 9:
        print('틀렸습니다. 다시 입력하세요. ')
        if randNum > userNum:
            print('난수는 입력한 숫자보다 큽니다.')
        else:
            print('난수는 입력한 숫자보다 작습니다.')
    else:
        print('게임에서 졌습니다.')
        break
print('난수 :', randNum)
```

다음 세션

다음 세션이 트랙분기 전 **마지막 공통 세션**이라서, 여러 **개발 꿀팁**과
유용한 **확장 프로그램**을 소개하려고 합니다.

다음 세션에서 **추가적으로 설명해줬으면 하는 부분**이 있으면,
운영진한테 카톡 보내주세요!!
다음 세션 강의할 때 **적극 반영** 하겠습니다~

감사합니다 😊