[Python 기본 코딩]

```
#리스트2.1 : 여러 가지 변수
a = 123  # 변수a에 정수123을 대입
b_123 = 123.456  # 변수b_123에 소수123.456을 대입
hello_world = "Hello World!"  # 변수hello_world에 문자열"Hello World!"를 대입
#리스트2.2 : 값 표시의 예
a = 123
print(a)
#리스트2.3 : 값을 합해서 표시하는 예
print(123, 123.456, "Hello World!")
#리스트2.4 : 변수의 대입
b = 456
#리스트2.5 : 변수에 저장된 값을 표시
print(b)
#리스트2.6 : 여러 가지 Python의 연산자
a = 3
b = 4

c = a + b  # 덧셈
print("덧셈:", c)

d = a - b  # 뺄셈
print("뺄셈:", d)

e = a * b  # 곱셈
print("곱셈:", e)

f = a ** b  # 거듭제곱 ( a의b승 )
print("거듭제곱:", f)
#리스트2.7 : 나눗셈 연산자
g = a / b  # 결과는 소수
print("나눗셈(소수): ", g)

h = a // b  # 결과는 정수
print("나눗셈(정수): ", h)

i = a % b  # 나머지
print("나머지: ", i)
#리스트2.8 : 변수 자신에게 연산을 실시한다
j = 5
j += 3  # 3을 더한다. j = j + 3과 같다
print("3을 더한다: ", j)

k = 5
k -= 3  # 3을 뺀다. k = k - 3과 같다
print("3을 뺀다: ", k)
#리스트2.9 : 큰 수, 작은 수의 표기
a = 1.2e5  # 120000
print(a)

b = 1.2e-4  # 0.00012
print(b)
#리스트2.10 : 리스트로 여러 개의 값을 합한다
a = [1, 2, 3, 4]
print(a)
#리스트2.11 : 리스트의 요소를 인덱스로 꺼낸다
b = [4, 5, 6, 7]
print(b[2])  # 맨 앞부터 0, 1, 2, 3,...라고 인덱스를 붙인 경우, 인덱스가 2인 요소
#리스트2.12 : 리스트에 요소를 추가한다
c = [1, 2, 3, 4, 5]
c.append(6)  # 리스트에 6을 추가
print(c)
#리스트2.13 : 리스트 안에 리스트를 넣는다
d = [[1, 2, 3], [4, 5, 6]]
print(d)
#리스트2.14 : 리스트의 전체 요소를 여러 번 나열한 새로운 리스트를 만든다
e = [1, 2]
print(e * 3)  # 리스트 e의 전체 요소를 세 번 나열한 새로운 리스트
#리스트2.15 : 튜플 요소에 접근한다
a = (1, 2, 3, 4, 5)  # 튜플의 작성
```

```
b = a[2]  # 인덱스가 2인 요소를 취득
print(b)
#리스트2.16 : 튜플의 요소에 접근한다
c = (3,)
print(c)
#리스트2.17 : 리스트나 튜플의 요소를 한데 모아서 변수에 대입한다
d = [1, 2, 3]
d_1, d_2, d_3 = d
print(d_1, d_2, d_3)

e = (4, 5, 6)
e_1, e_2, e_3 = e
print(e_1, e_2, e_3)
#리스트2.18 : if문에 의한 조건분기
a = 5

if a > 3:  # a가 3보다 크면
    print(a + 2)  # 인덴트를 맨 앞에 삽입한다
else:  # a > 3를 만족하지 않으면
    print(a - 2)
#리스트2.19 : '=='의 연산자에 의한 비교
b = 7
if b == 7:  # b가 7과 같으면
    print(b + 2)
else:  # b == 7를 만족하지 않으면
    print(b - 2)
#리스트2.20 : for문과 리스트를 사용한 루프
for a in [4, 7, 10]:  # 리스트의 각 요소가 변수a에 들어간다
    print(a + 1)  # 루프에서 실행하는 처리에는 인덴트를 넣는다
#리스트2.21 : range를 사용한 루프
for a in range(5):  # a에는 0부터 4가 들어간다
    print(a)
#리스트2.22 : 함수를 정의하고, 호출한다
def my_func_1():  # my_func_1이 함수명
    a = 2
    b = 3
    print(a + b)

my_func_1()  # 함수 호출
#리스트2.23 : 인수를 동반하는 함수
def my_func_2(p, q):  # p, q가 인수
    print(p + q)

my_func_2(3, 4)  # 함수를 호출할 때에 값을 전달한다
#리스트2.24 : 인수와 반환값을 동반하는 함수
def my_func_3(p, q):  # p, q가 인수
    r = p + q
    return r  # r이 반환값

k = my_func_3(3, 4)  # 반환값으로서 받은 값을 k에 넣는다
print(k)
#리스트2.25 : 반환값을 튜플로 한다
def my_func_3(p, q):  # p, q가 인수
    r = p + q
    s = p - q
    return (r, s)  # 반환값을 튜플로 한다

k, l = my_func_3(5, 2)  # 튜플의 각 요소를 k,l에 대입한다
print(k, l)
#리스트2.26 : 로컬 변수와 글로벌 변수
a = 123  # 글로벌 변수

def show_number():
    b = 456  # 로컬 변수
    print(a, b)  # 양쪽에 접근할 수 있다

show_number()
#리스트2.27 : 글로벌 변수와 같은 이름의 로컬 변수
a = 123  # 글로벌 변수

def set_local():
```

```
    a = 456  # a는 위에 적은 것과 다른 로컬 변수
    print("Local:", a)

set_local()
print("Global:", a)  # 글로벌 변수의 값은 변하지 않는다
#리스트2.28 : 인수로서 사용되는 변수의 스코프
a = 123  # 글로벌 변수

def show_arg(a):  # a는 위에 적은 것과 다른 로컬 변수
    print("Local:", a)

show_arg(456)
print("Global:", a)  # 글로벌 변수의 값은 변하지 않는다
#리스트2.29 : 함수 내에서 글로벌 변수의 값을 변경한다
a = 123  # 글로벌 변수

def set_global():
    global a  # nonlocal로도 가능
    a = 456  # 글로벌 변수의 값을 변경
    print("Global:", a)

set_global()
print("Global:", a)
#### □문제
# Jupyter Notebook의 셀에 리스트, 튜플, if문, for문, 함수 예문을 최소 하나씩 써봅시다.
#### □정답 예
#리스트2.30 : 정답 예
# ---리스트의 예 ---
print("--- 결과: 리스트 ---")
my_list = [1, 2, 3, 4, 5]
print(my_list[2])

print()  # 빈 행

# ---튜플의 예 ---
print("--- 결과: 튜플 ---")
my_tuple = (1, 2, 3, 4, 5)
print(my_tuple[3])

print()  # 빈 행

# --- if문의 예---
print("--- 결과: if문 ---")
a = 5
b = 2
if a == 5:
    print(a + b)

print()

# --- for문의 예 ---
print("--- 결과: for문 ---")
for m in my_list:
    print(m + 1)

print()

# --- 함수의 예 ---
print("--- 결과: 함수 ---")
def add(p, q):
      return p + q
print(add(a, b))
```