

CHAP.3

0과 1의 세계에 빠져보자

학습 내용

- 반복문
- 조건문
- 문자열
- 구조적프로그래밍
- 객체지향프로그래밍

1부터 10까지 더한 값은

```
a=1+2+3+4+5+6+7+8+9+10  
print(a)
```

55

```
a=1  
a=1+2  
a=3+3  
a=6+4  
a=10+5  
a=15+6  
a=21+7  
a=28+8  
a=36+9  
a=45+10  
print(a)
```

최종 코드

필요 없는 변수 제거

```
b=0
a=0
for i in range(10):
    b=b+1
    a=a+b
print(a)
```

```
a=0
for i in range(10):
    a=a+(i+1)
print(a)
```

참고 – 재귀 함수; recursion

함수에서 자기자신을 다시 호출하는 형태

```
def sum(n):  
    if n==1:  
        return 1  
    return n+sum(n-1)
```

```
a = sum(10)  
print(a)
```

덧셈으로 곱셈 계산하기

곱셈은 덧셈을 여러 번 하는 것

```
>>>a = 2
>>>b = a + a + a
>>>print(b)
6
>>>b = a * 3
>>>print(b)
6
```

```
a=2
b=3
s=0
for i in range(b):
    s = s + a
print(a, "X", b, "=", s)
```

a.py

```
C:\>python a.py
2 x 3 = 6
```

함수로 바꾸기

함수정의하고 indentation을 주어 함수의 코드 블록으로 바꾼다.

```
def multiply(a, b):
```

```
    s = 0
```

```
    for i in range(b):
```

```
        s = s + a
```

```
    return s
```

```
a=2
```

```
b=3
```

```
c = multiply(a, b)
```

```
print(a,"X",b,"=",c)
```

```
c = multiply(4, 5)
```

```
print(4,"X",5,"=",c)
```

```
2 x 3 = 6
```

```
4 x 5 = 20
```

테스트가 중요하다

다양한 상황의 입력 값 준비

0	0
2	0
0	2
2	-2
-2	2
-2	-2

} 테스트 값 준비하기


```

def multiply(a, b):
    s = 0
    a_abs = a
    b_abs = b

    negatives = 0;
    if a<0:
        negatives += 1
        a_abs = -a
    if b<0:
        negatives += 1
        b_abs = -b

    #positive: True, negative: False
    sign = negatives % 2 == 0

    for i in range(b_abs):
        s = s + a_abs

    if (sign==False):
        s = -s

    return s

```

이렇게 구현할 수도 있다.

```

a=-2
b=-3
c = multiply(a, b)
print(a,"X",b,"=",c)

```

이렇게 까지 하면 더 좋다

서울로 가는 길은 많다. 지금은 가기만 해도 좋다. 2학년 말이 되면 제일 좋은 길을 갈 수 있어야 한다.

```
def multiply(a, b):  
    s = 0  
    if (b<0):  
        for i in range(-b):  
            s = s - a  
    else:  
        for i in range(b):  
            s = s +a  
    return s
```

```
a=-2  
b=-3  
c = multiply(a, b)  
print(a,"X",b,"=",c)
```

$$-2 \times -3 = 6$$

() 용법

여러 개를 묶어 준다. 튜플(tuple)이라는 이름을 갖고 있다. 나중에 자세히 살펴본다.

```
a = 1
b = 2
print(a, b)
t = a
a = b
b = t
print(a, b)
```

일반적으로
두 값을 바꾸는 방법

```
1 2
2 1
```

```
a = 1
b = 2
print(a, b)
(a, b) = (b, a)
print(a, b)
```

파이썬에서
두 값을 바꾸는 방법

a, b = b, a
보통은 괄호는 생략

작은 수 큰 수 둘다 남기는 함수 만들기

이럴 때 튜플을 쓴다.

```
import random
def getMaxMin(a, b):
    if a < b :
        maxValue = b
        minValue = a
    else:
        maxValue = a
        minValue = b
    return (maxValue, minValue)
```

두 값을 하나로 묶어서 남긴다

```
a=random.randint(0, 100)
b=random.randint(0, 100)
max, min = getMaxMin(a, b)
print("max:", max, "min:", min)
```

벨셈으로 나눗셈 계산하려면...

```
def divide(a, b):
```

```
    pass
```

이 부분을 채우는 것

```
a=1
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=3
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=7
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
1 / 3 = 0 x 3 + 1
```

```
3 / 3 = 1 x 3 + 0
```

```
7 / 3 = 2 x 3 + 1
```

나눗셈은 뺄셈 반복하는 것

곱셈과는 달리 몇 번 반복할지 모른다.

```
def divide(a, b):
```

```
    pass
```

for로는 힘들겠다

```
a=1
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=3
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=7
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
1 / 3 = 0 x 3 + 1
```

```
3 / 3 = 1 x 3 + 0
```

```
7 / 3 = 2 x 3 + 1
```

while

몇 번 반복할지 모를 때 쓰는 반복문

```
for i in range(3):
```

```
    print(i)
```

```
print("end of 'for' ")
```

0

1

2

end of 'for'

```
i=0
```

```
while(i < 3):
```

```
    print(i)
```

```
    i = i + 1
```

```
print("end of 'while' ")
```

0

1

2

end of 'while'

나눗셈 함수 구현 – 빼기 반복 방식으로

/ 또는 // 연산자 사용 불가

```
def divide(a, b):
```

```
    pass
```

while 문 이용한다

```
a=1
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=3
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
a=7
```

```
b=3
```

```
q, r = divide(a, b)
```

```
print(a,"/",b,"=",q,"x",b,"+",r)
```

```
1 / 3 = 0 x 3 + 1
```

```
3 / 3 = 1 x 3 + 0
```

```
7 / 3 = 2 x 3 + 1
```


별표 출력

반복문 연습의 고전

*

**

```
>>>print("*")
```

```
>>>print("**")
```

```
>>>print("***")
```

```
>>>print()
```

```
print("*", end="")
```

```
print()
```

```
print("*", end="")
```

```
print("*", end="")
```

```
print()
```

```
print("*", end="")
```

```
print("*", end="")
```

```
print("*", end="")
```

```
print()
```

1 단계

반복문 바꾼다. 변하는 부분은 변수로. 우리가 애용하는 그 수식 적용

```
for i in range(1):  
    print("*", end="")  
print()
```

```
for i in range(2):  
    print("*", end="")  
print()
```

```
for i in range(3):  
    print("*", end="")  
print()
```

```
j=0  
j=j+1  
for i in range(j):  
    print("*", end="")  
print()  
j=j+1  
for i in range(j):  
    print("*", end="")  
print()  
j=j+1  
for i in range(j):  
    print("*", end="")  
print()
```

완성 코드

반복되는 부분을 찾아서 반복문 적용

```
j=0
j=j+1
for i in range(j):
    print("*", end="")
print()
j=j+1
for i in range(j):
    print("*", end="")
print()
j=j+1
for i in range(j):
    print("*", end="")
print()
```

```
j=0
for k in range(3):
    j=j+1
    for i in range(j):
        print("*", end="")
    print()
```

range

더 세밀한 사용법을 알아보자.

```
for i in range(3):  
    print(i)
```

0
1
2

```
for i in range(1, 3):  
    print(i)
```

1
2

1부터 10까지 숫자 중 짝수만 출력하기

range 세 번째 값의 역할. 2씩 증가하게 한다.

```
for i in range(1, 11):  
    if (i % 2 == 0):  
        print(i, end=" ")
```

2 4 6 8 10

```
for i in range(2, 11, 2):  
    print(i, end=" ")
```

2 4 6 8 10

1부터 10까지 숫자 중 홀수만 출력하기

range 세 번째 값의 역할. 2씩 증가하게 한다.

```
for i in range(1, 11):  
    if (i % 2 == 1):  
        print(i, end=" ")
```

1 3 5 7 9

```
for i in range(1, 11, 2):  
    print(i, end=" ")
```

1 3 5 7 9

카운트 다운

range 세 번째 값에 음수도 지정할 수 있다.

```
for i in range(5, 0, -1):  
    print(i, end=' ')
```

5 4 3 2 1

```
for i in range(3, 0, -1):  
    print(i)  
print("평")
```

3
2
1
평

카운트 다운 완성

time 모듈은 sleep()이라는 함수를 제공한다. 지정한 초 동안 아무것도 안한다.

```
for i in range(3, 0, -1):
    print(i)
    for j in range(10000000):
        pass
print("평")
```

```
import time
for i in range(10, 0, -1):
    print(i)
    time.sleep(1)  모듈이름 + 점 + 함수이름
                   1초 동안 멈춘다.
```

```
print("평")
print("%c"%(7))
#print("{0:c}".format(7))
```

소리 나게 하는 방법 정도로 알고 있자.
7, %c, {0:c} 은 나중에 나온다.

문자열 다루기

- 컴퓨터는 숫자만 저장할 수 있다.
- 문자를 대신하는 숫자를 배정해 놔다.
- 예들 들면, 'A'는 65, 'B'는 66이런 식이다.
- 이렇게 문자를 숫자로 배정해 놓은 표가 있다.
- 이 표는 세계공통이다.
- 우리가 사용하는 것은 미국표준협회에서 정한 표다.
- ASCII code table이라 부른다. American Standard Code for Information Interchange
- 알아두면 유익한 코드; 7, 8, 13, 27, 32, 65

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	,
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	\$	68	D	100	d	132	Ñ	164	§	196	f	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41)	73	I	105	i	137	â	169	©	201	...	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	'	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	“	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Ö	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	-	240	Ⓜ
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	`	244	Ü
21	<NAK>	53	5	85	U	117	u	149	ï	181	μ	213	'	245	ı
22	<SYN>	54	6	86	V	118	v	150	ñ	182	ð	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	Σ	215	◇	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	Π	216	ÿ	248	—
25		57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	/	250	˙
27	<ESC>	59	;	91	[123	{	155	õ	187	ª	219	€	251	˚
28	<FS>	60	<	92	\	124		156	ú	188	º	220	<	252	¸
29	<GS>	61	=	93]	125	}	157	ù	189	Ω	221	>	253	”
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˚
31	<US>	63	?	95	_	127		159	ü	191	ø	223	fl	255	˘

문자 코드 관련 함수

chr

```
>>>65
65
>>>chr(65)
'A'
>>>chr(66)
'B'
>>>chr(67)
'C'
```

```
>>>for i in range(65, 65+26):
...     c = chr(i)
...     print(c, end='')
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

문자를 코드로

ord

```
>>>ord('A')
```

```
65
```

```
>>>ord('B')
```

```
66
```

```
>>>for i in range(ord('A'), ord('Z')+1):
```

```
...     c = chr(i)
```

```
...     print(c, end='')
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

반복문 연습



Photo by Josh Hild on Unsplash

별 문자 찍기

```
*  
**  
***
```

```
***  
**  
*
```

```
  *  
  **  
 ***
```

```
  *  
 ***  
*****
```

별 문자 출력 힌트

```
*
**
***
```

```
***
**
*
```

```
    *
  **
 ***
```

```
    *
  ***
 *****
```

```
      *
    ***
  *****
 *****
```

range(3,0,-1)

공백 갯수가 2, 1, 0개
별표 갯수가 1, 2, 3개

공백 갯수가 2, 1, 0개
별표 갯수가 1, 3, 5개

```
      *
    ***
  *****
 *****
      *
      *
```

4
2

다양한 문자 출력 방법

`%c, %d, %f`

```
>>>a=65
>>>b=66
>>>print(a, b)
65 66
>>>print(chr(a), chr(b))
A B
>>>print('%c'%a)
65
>>>print('%c %c'%(a, b))
A B
```

```
>>>print('%d %d'%(a, b))
65 66
>>>print('%f %f'%(a, b))
65.000000 66.000000
>>>print('%.2f %.2f'%(a, b))
65.00 66.00
>>>print('%3d %3d'%(a, b))
 65  66
>>>print('%03d %03d'%(a, b))
065 066
```


문자열 출력 영역, 좌우 정렬 지정

%s

```
>>>a = 'hello'
>>>b = 'world'
>>>print('%s%s'%(a, b))
helloworld
>>>print('%6s%6s'%(a, b))
_hello_world
>>>print('%-6s%6s'%(a, b))
hello_ _world
```

```
>>>a=1
>>>b=1
>>>print('%3d%3d'%(a, b))
_ _1_ _2
>>>print('%-3d%-3d'%(a, b))
1_ _2_ _
```

가장 진보된 출력 형식 지정 방식

`{}`

```
>>>a=65
>>>b=66
>>>print('{ } {}'.format(a, b))
65 66
>>>print('{0} {1}'.format(a, b))
65 66
>>>print('{1} {0}'.format(a, b))
66 65
>>>print('{1} {1}'.format(a, b))
66 66
```

```
>>>a=65; b=66
>>>print('{0:c} {1:c}'.format(a, b))
A B
>>>print('{0:d} {1:d}'.format(a, b))
65 66
>>>print('{0:d} {1:f}'.format(a, b))
65 66.000000
>>>print('{0:d} {1:.2f}'.format(a, b))
65 66.00
>>>print('{0:d} {1:10.2f}'.format(a, b))
65 _ _ _ _ _ 66.00 10칸
```