

## 문자열 처리

### 유용한 문자열 함수

```
소문자 만들기: "A".lower()="a"
대문자 만들기: "a".upper()="A"
제목형식 만들기: "hi world".title()="Hi World"
문자열 치환: "123".replace('2','z')="1z3"
문자 갯수 세기: "1123".count("1")=2
특정 문자 오프셋 구하기: "123".index("2")=1
특정 문자 탐지: "is" in "fish" == True
"a string" 인코딩: "a string".encode(코덱 명)
"a string" 디코딩: "a string".decode(코덱 명)
ROT13 코덱으로 사례:
>>> print "RAPBQR-ZR".decode("rot13")
ENCODE-ME
```

### 일부 문자열 인코더/디코더 명:

```
Base64, bz2, hex, rot13, uu, zip,
string_escape
```

### 문자열을 리스트로 변환(기본 분리자는 공백):

```
newlist="astr".split(separator)
>>> print "A B C".split()
['A', 'B', 'C']
>>> print "A B C".split()
['A', 'B', 'C']
>>> print "A,B, ,C".split(",")
['A', 'B', ' ', 'C']
>>> print "WANNA BANANA?".split("AN")
['W', 'NA B', '', 'A?']
```

### 리스트(또는 다른 반복하는 객체)를 문자열로 변환:

```
"astring".join([list])
>>> print "".join(['A','B','C'])
ABC
>>> print ",".join(['A','B','C'])
A,B,C
```

## 데이터 유형 변환하기

### 어떤 것이든 문자열로 변환:

```
newstring = str(<any variable>)
newstring = str(100)
```

### 문자열을 정수형으로 변환:

```
newint = int(<string>[,base])
아래함수는 모두 변수 10을 정수 10으로 할당
>>> ten = int("1010",2)
>>> ten = int("0010")
>>> ten = int("000A",16)
```

### 실수형에서 소수점이하 버리고 정수형으로 변환:

```
newint = int(<float>)
>>> print int(3.1415)
3
>>> int(3.6)
3
```

### 문자열 또는 숫자를 실수형으로 변환:

```
afloat = float(<var>)
>>> print float("1.5")
1.5
>>> print float(1)
1.0
```

### 문자열을 ASCII 숫자로 변환:

```
newint = ord(<string length 1>)
>>> print ord("A")
65
```

### ASCII 숫자를 한 자리 문자로 변환:

```
newstr = chr(<integer 1 to 255>)
>>> print chr(65)
A
```



## 파이썬 2.7 핵심

번역: SANS 코리아

포켓 참고 가이드

<https://www.sans.org/sec573>  
<https://www.itlkorea.kr/sans/>

## 파이썬 실행 3가지 방법

### -c로 명령어 라인 실행:

```
$ python -c ["스크립트 문자"]
python -c "print('Hello World!')"
```

### 파이썬 인터프리터 스크립트 실행:

```
$ cat helloworld.py
print("Hello World")
$ python helloworld.py
Hello World
```

### 파이썬 상호작용 셸:

```
$ python
>>> print("Hello World")
Hello World
```

## 파이썬 명령어 라인 옵션

```
$ python -c "스크립트 문자열"
스크립트가 있는 문자열을 실행
$ python -m <module> [module args]
경로에서 모듈을 찾아 스크립트로 실행
예: python -m "SimpleHTTPServer"
$ python -i <python script>
스크립트 실행 후, 상호작용 셸로 전환
```

## 루프 목록 및 사전

### 리스트 핵심내용:

```

빈 리스트 생성: newList=[]
인덱스에 값 할당: alist[index]= value
인덱스의 값에 접근: alist[index]
리스트에 아이템 추가: alist.append(new item)
리스트로 삽입:
alist.insert(at position, new item)
리스트에 아이템의 갯수 세기: alist.count(item)
일치하는 한 아이템 삭제: alist.remove(del item)
인덱스에 아이템 제거 del alist[index]
    
```

### 사전 핵심내용:

```

빈 사전 생성: dic={}
비어있지 않은 사전 초기화
dic = { "key": "value", "key2": "value2"}
값 할당: dic["key"]="value"
키가 존재하는 지 결정: "key" in dic
키에 값 접근: dic["key"], dic.get("key")
모든 키를 반복적으로 보기: dic.keys()
모든 값을 반복적으로 보기: dic.values()
(key,value) 쌍으로 반복적으로 보기: dic.items()
    
```

### 루프 예:

```

0 ~ 9까지 For 루프: for x in range(10):
5 ~ 11까지 For 루프: for x in range(5,11):
문자열에서 문자 For 루프: for char in astring:
리스트에서 아이템 For 루프:
for x in alist:
리스트에서 인덱스 및 값을 뽑아내는 For 루프:
for index,value in enumerate(alist):
사전에서 키 For 루프: for x in adict.keys():
사전에서 모든 아이템 For 루프:
for key,value in adict.items():
while <logic test> do:
    
```

### 루프 제어문(for 및 while):

```

즉시 루프 빠져나오기 break
나머지 루프 건너뛰고, 다시 루핑 continue
    
```

## 기타

### 코드에 주석 추가:

```
# 문자로 라인을 시작하면 주석이 됩니다.
```

### 함수 정의:

다음은 "add"라는 함수입니다. 2개의 인수 num1과 num2를 받아들이고 그 합을 반환합니다. "print (add (5,5))"를 호출하면 "10"이 인쇄됩니다.

```

def add(num1, num2):
    #code blocks must be indented
    #each space has meaning in python
    myresult = num1 + num2
    return myresult
    
```

### if then else 구문:

```

if <logic test 1>:
    #when <logic test 1>이 True 인 경우
    #코드가 실행된다.
elif <logic test 2>:
    #<logic test 1>이 False 이고,
    #<logic test 2>가 True 이면, 코드가 실행된다.
else:
    #if 및 모든 elif 가 False이면 실행된다.
    
```

## 문자열 슬라이싱 및 인덱싱 등

x[시작:끝:증가]	x=[4,8,9,3,0]	x="48930"
x[0]	4	'4'
x[2]	9	'9'
x[:3]	[4,8,9]	'489'
x[3:]	[3,0]	'30'
x[:-2]	[4,8,9]	'489'
x[::2]	[4,9,0]	'490'
x[::-1]	[0,3,9,8,4]	'3984'
len(x)	5	'5'
sorted(x)	[0,3,4,8,9]	['0','3','4','8','9']

## SEC573 PyWars 핵심내용

### pyWars 객체 생성

```
>>> import pyWars
>>> game= pyWars.exercise()
```

### 계정 관리

```
>>> game.new_acct("username", "password")
>>> game.login("username", "password")
>>> game.logout()
```

### 질문 쿼리:

```
>>> game.question(<question #>)
```

### 데이터 쿼리:

```
>>> game.data(<question #>)
```

### 답 제출:

```
>>> game.answer(<question #>,
solverfunc(game.data(<question#>)))
```

## 수학연산 및 논리연산

수학 연산	예	X=7, Y=5
더하기	X + Y	12
빼기	X - Y	2
곱하기	X * Y	35
나누기	X / Y	1
지수	X ** Y	16807
모듈러	X % Y	2
논리 연산		
등가	X == Y	FALSE
보다 큰	X > Y	FALSE
보다 작은	X < Y	TRUE
적거나 같은	X <= Y	TRUE
같지 않다	X != Y 또는 X <> Y	TRUE
다른 논리연산: AND, OR 또는 NOT		