내장함수 이해

9주차_01_01

한 동 대 학 교
김경미 교수

학습목표

- 함수를 왜 사용하는지 이해하기
- 내장함수가 무엇인지 이해하기

함수사용 이유

- 코드가 길어질 때 모듈화하여 간결성 높인다
- 반복되는 구분을 모듈화 하여
  - 고치기 쉽고(easy to modify)
  - 운영과 관리를 용이하게 하며(flexible to maintain)
  - 프로그램 가독성을 높게 한다(readability)

함수

- 함수의 종류
  - 내장 함수(Built-in function)
  - 사용자 정의 함수(User defined function)
- 함수를 정의하려면
  - 함수 이름과 명령문들을 순서대로 쓴다
  - 함수 이름을 불러서 함수를 "호출" 한다 (function call)

사용해 본 내장함수

- print()
- input()
- int()
- float()
- range()
- len()
- deepcopy()

내장 함수(Built-in Function)

- 시스템에서 제공해 주는 함수들
  - 사용한 내장 함수

```
>>> int('55')
>>> print( 'kmkim' )
>>> range(10)
```

  - 어떤 함수들은 모듈을 import한 후 사용 가능

```
>>> import math
>>> math.sqrt(4) / 2.0
1.0
```

내장함수 종류

http://docs.python.org/3.3/library/functions.ht

abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

내장함수 활용 I


```
# abs() // 절대값 반환
>>> abs(-3)
3
>>> abs(-10)
10

# chr() // 유니코드 값을 입력받아 그에 해당하는 문자 반환
>>> chr(97)
a
>>> chr(80)
P

# reversed() // 요소의 순서를 뒤집어서 반환
>>> t = ('a', 'b', 'c')
>>> list(reversed(t))
['c', 'b', 'a']
```

내장함수 활용 2


```
# float() // 실수 형으로 변환
>>> float(3)
3.0
>>> float(10)
10.0


# format() // 주어진 서식에 맞춰 값을 문자열로 변환
>>> print("I want {0} apples".format(3))
I want 3 apples
```

내장함수 활용 3


```
# id() // 객체의 메모리 주소 값을 반환
>>> s='apple‘
>>> print(id(s))
```

```
140119636514480
```

```
# len() // 문자열의 길이 반환
>>> len(”Hello Python”)
12
>>> len(“piano”)
5
```

내장함수 활용 4


```
# list() // 리스트를 생성하여 반환
>>> s ='apple'
>>> list(s)
['a', 'p', 'p', 'l', 'e' ]

# max() // 가장 큰 요소 반환
>>> b = [10, 2, 6, 8]
>>> max(b)
10

# min() // 가장 작은 요소 반환
>>> a= [6,2,3,10]
>>> min(a)
2
```

내장함수 활용 5


```
# ord() // 유니코드 번호 반환
>>> ord('B')
66
>>>ord('b')
98

# pow() // 제곱 값을 반환
>>> import math
>>> pow(4,2)
16
>>> pow(2,3)
8
```

내장함수 활용 6


```
# print() // 해당 값을 출력
>>> print('handong')
handong
>>> a=4
>>> print(a)
4

# range() // 지정된 횟수만큼 숫자를 생성하는 객체 반환
>>> for i in range(5):
        print(i)
0
1
2
3
4
```

내장함수 활용 7


```
# round() // 반올림한 수를 반환
>>> import math
>>> round(5.4)
5
>>> round(5.5)
6

# sorted() // 정렬된 리스트를 반환
>>> a = [2, 4, 1, 9, 100]
>>> sorted(a)
[1, 2, 4, 9, 100]

# str() // 문자열 타입으로 반환
>>> str(3+9)
'12'
>>> str(5.0)
'5.0'
```

내장함수 활용 8


```
# sum() // 모든 요소의 합 반환
>>> a= [3,8,6]
>>> sum(a)
17

>>> b =[3,6,1,8,9]
>>> sum(b)
27

# tuple() // 튜플을 생성하여 반환
>>> alist=[1,2,'a','b']
>>> tuple(alist)
(1, 2, 'a', 'b')
```

연습문제 I

- 내장 함수를 활용하여 다음을 출력하시오
- 리스트 n=[I, 3, 5, 7, 99, 97, 95, 93, 9I]를 대상으로 아이템의 개수, 모든 아이템의 합계를 출력하고 리스트 n을 순서 거꾸로 출력하시오

연습문제 I, 코드


```
n=[ 1, 3, 5, 7, 99, 97, 95, 93, 91]

print('아이템 수: ', len(n))
print('총 합계 : ', sum(n))

print('원래 리스트 :', n)
print('순서 거꾸로 리스트 :',list(reversed(n)))
```

```
아이템 수:  9
총 합계 :  491
원래 리스트 : [1, 3, 5, 7, 99, 97, 95, 93, 91]
순서 거꾸로 리스트 : [91, 93, 95, 97, 99, 7, 5, 3, 1]
```

강의 요약

- 함수를 왜 사용하는지 이해하기
  - 반복되는 구문을 모듈화
  - 수정이 간편
  - 운영과 관리가 용이
  - 가독성이 높아짐
- 내장함수가 무엇인지 이해하기
  - 시스템에서 제공해주는 함수들

목표 달성 질문

- 함수를 사용하는 목적을 3가지 기술하시오
- 지금까지 사용해 본 내장함수 3개를 쓰시오

감사합니다