파이썬 프로그래밍

7차시

내장 함수와 깃허브

⚠ 학습개요

… 내장 함수 개요

… 깃허브 소개와 교재의 깃허브 소개

… 복습 : bin(), oct(), hex(), int()

… 프로젝트 lab1, lab2

⚠ 학습목표

… 함수와 내장 함수를 설명할 수 있다.

… 깃허브를 이해하고 활용할 수 있다.

… print() int() 등 내장 함수를 잘 활용할 수 있다.

… 정수나 실수를 입력 받아 다양한 연산을 수행하는 프로그래밍을 할 수 있다.

… 진수를 정한 후 해당 진수를 입력 받아 다양한 진수 형태로 출력하는 프로그래밍을 할 수 있다.

Chapter 1.

내장 함수 개요

PYTHON PROGRAMMING

⚠ 함수 개요

+ 함수 정의와 활용

- 7장에서 자세히 학습

+ 함수 정의

- 특정한 기능을 수행하는 프로그램 단위인 함수
  - 여러 입력을 받아 특정한 기능을 수행하고 결괏값을 반환하는 코드
  - 사용자가 직접 정의해 사용
  - 파이썬에 설치된 다양한 함수를 활용

⚠ 함수 개요

[그림7-1] 함수 개념 커피머신과 믹서기

⚠ 내장 함수

+ 사용자 정의 함수의 함수 정의와 함수 호출

- **사용자 정의 함수**(user-defined functions)와 **내장 함수**(built-in functions)로 구분
- **print( )**와 **input( ), str( ), int( ), float( ), len( )** 등과 같은 내장 함수 사용

⚠ 내장 함수

+ 사용자 정의 함수(user-defined functions)

우리가 직접 만드는 함수

함수 이름 인자

```
def fahr_to_celsius(temp):
    return ((temp - 32) * (5/9))
```

반환 값

def keyword name parameter

```
def fahr_to_kelvin(temp):
    return ((temp - 32) * (5/9)) + 273.15
```

return statement

return value

[그림7-2] 사용자 정의 함수

⚠ 내장 함수

✚ 내장 함수(built-in functions)

✚ [표] 파이썬 설치와 함께 이미 만들어 놓은 함수(1/2)

내장함수				
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	rod()	sum()

⚠ 내장 함수

+ 내장 함수(built-in functions)

+ [표] 파이썬 설치와 함께 이미 만들어 놓은 함수(2/2)

내장함수				
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()	

Chapter 2.

깃허브 소개와 교재의 깃허브 소개

PYTHON PROGRAMMING

⚠ 깃허브

+ 프로젝트 관리(버전 관리) 시스템

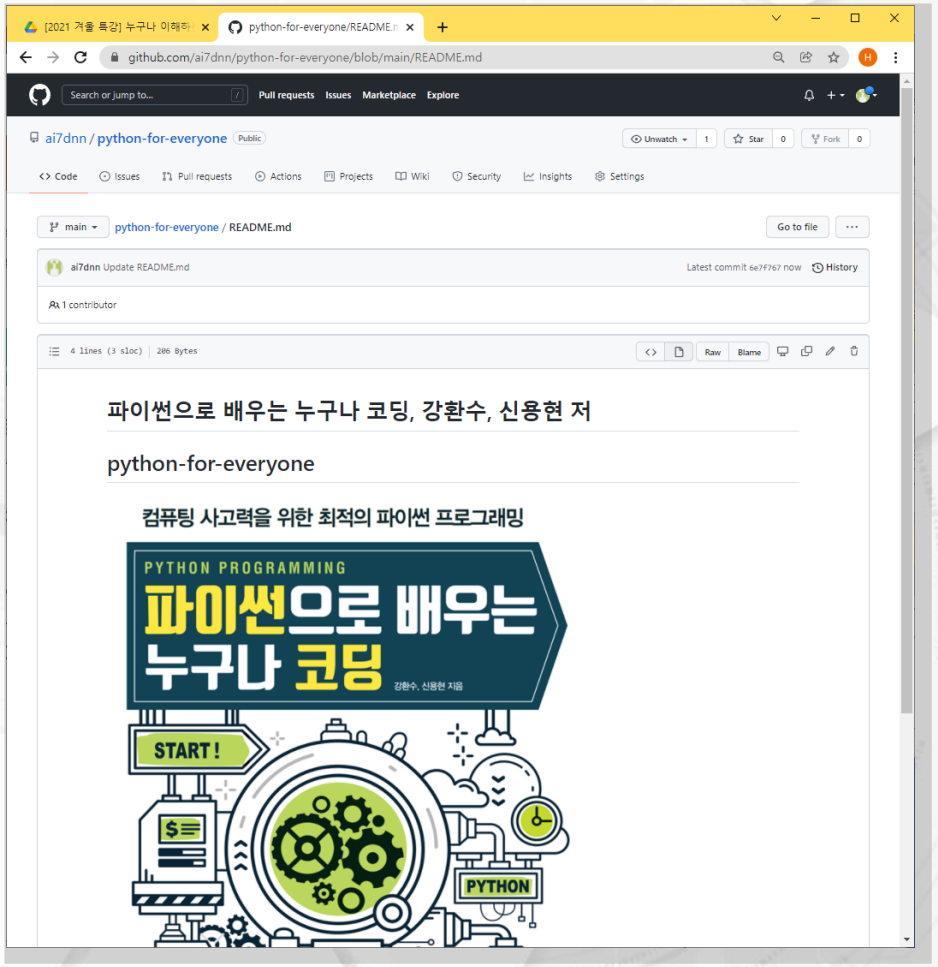
- 오픈 소스 프로젝트
  - 조직의 소프트웨어 개발 프로젝트 지원 시스템
- 버전 관리 시스템인 git의 서버 지원 시스템

+ 계정 생성 후 사용

- 구글 계정으로 생성
- 공개된 다른 계정을 사용하는 것은 계정이 필요 없음

+ 교재의 깃허브 사이트

- **https://github.com/ai7dnn/python-for-everyone**



Chapter 3.

복습

bin(), oct(), hex(), int()

PYTHON PROGRAMMING

⚠ IDLE 실습

+ 진수 변환 함수

- bin(), oct(), hex()
  - Return the binary(octal, hexadecimal) representation of an integer
  - 정수 값을 바로 2진수, 8진수, 16진수로 표현되는 "문자열"로 변환
    - 10진수 뿐 아니라 다른 진수의 표현도 가능
    - 반환 값은 해당 진수의 문자열 표현
- int(십진수), int('십진수의 문자열 표현')
  - 모두 10진수 반환

⚠ IDLE 실습

+ 진수 변환 함수

- int('2, 8, 16 진수의 문자열 표현')
  - 오류
- int('해당 진수의 문자열 표현', n)
  - 문자열 표현의 수를 진수 n으로 해석해 10진수로 출력

+ 문제

- 10과 20을 각각 2진수로 표현

⚠ IDLE 실행


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> dir(bin)
['__call__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__name__', '__ne__', '__new__', '__qualname__', '__reduce__', '__reduce_ex__', '__repr__', '__self__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__text_signature__']
>>> bin.__doc__
"Return the binary representation of an integer.₩n₩n >>> bin(2796202)₩n '0b1010101010101010101010'"
>>> bin(10)
'0b1010'
>>> bin(0b1010)
'0b1010'
>>> bin(0o55)
'0b101101'
>>> bin(10) + bin(20)
'0b10100b10100'
>>> int(bin(10), 2)
10
>>> int(bin(20), 2)
20
>>> int(bin(10), 2) + int(bin(20), 2)
30
>>> bin(int(bin(10), 2) + int(bin(20), 2))
'0b11110'
>>> bin(30)
'0b11110'
>>>
Ln: 25 Col: 0

끝

Chapter 4.

프로젝트
lab1, lab2

PYTHON PROGRAMMING

⚠ 프로젝트 Lab 1

표준 입력한 실수와 연산식의 산술 연산 및 결과 출력 난이도 응용

표준 입력으로 2개의 실수를 입력 받아 더하기, 빼기, 곱하기, 나누기의 연산을 출력한다.
이후 다시 표준 입력으로 하나의 연산식을 한 줄에 입력 받아
그 결과를 출력하는 프로그램을 작성해 보자.

문제 이해 (Understanding)

이 문제는 서로 다른 2개의 부분을 해결하는 과정이다. 먼저 2개의 실수를 두 번의 표준 입력으로 입력 받아 입력된 두 실수의 사칙연산 결과를 출력한다.
다음으로 한 줄에 입력되는 산술 연산식을 문자열로 입력 받은 후 함수 eval( )을 사용해 그 연산식의 결과를 바로 출력한다.

⚠ 프로젝트 Lab 1

표준 입력한 실수와 연산식의 산술 연산 및 결과 출력 난이도 응용

표준 입력으로 2개의 실수를 입력 받아 더하기, 빼기, 곱하기, 나누기의 연산을 출력한다.
이후 다시 표준 입력으로 하나의 연산식을 한 줄에 입력 받아
그 결과를 출력하는 프로그램을 작성해 보자.

설계 (Design)

알고리즘(Algorithm)

① 두 실수를 입력받는다.
- 변수 num1, num2에 입력, 함수 input()과 float() 사용

② 두 실수의 사칙연산 결과를 출력한다.
- 연산자 +, -, *, /를 사용해 출력

③ 하나의 연산식을 입력받아 그 결과를 출력한다
- 변수 expression에 연산식 저장, 함수 eval() 사용해 출력

표준 입출력 샘플

```
첫 번째 수 입력 >> 3.4
두 번째 수 입력 >> 1.5
합: 4.9
차: 1.9
곱하기: 5.1
나누기: 2.2666666666666666
연산식 입력 >> 3 * 4 / 2
연산식: 3 * 4 / 2
결과: 6.0
```

⚠ 프로젝트 Lab 1

+ 구현(Implementation)

```
1. # 두 실수의 사칙연산과 표준 입력 연산식의 계산
2. num1 = float(input('첫 번째 수 입력 >> '))
3. num2 = float(input('두 번째 수 입력 >> '))
4. print('합:' num1 + num2)
5. print('차:' num1 - num2)
6. print('곱하기:' num1 * num2)
7. print('나누기:' num1 / num2)
8. expression = input('연산식 입력(예 3.2 + 4 * 1.5) >> ')
9. print('연산식:' expression, '결과:', eval(expression))
```

⚠ 프로젝트 Lab 1

+ 구현(Implementation)

테스트와 실행 결과 (Testing & Simulation)

```
첫 번째 수 입력 >> 10.0
두 번째 수 입력 >> 2.5
합: 12.5
차: 7.5
곱하기: 25.0
나누기: 4.0

연산식 입력 >> 30 // 4
연산식: 30 // 4 결과: 7
```

```
첫 번째 수 입력 >> 3.9
두 번째 수 입력 >> 1.4
합: 5.3
차: 2.5
곱하기: 5.46
나누기: 2.785714285714286
```

실행오류

```
연산식 입력 >> 3 // * 3
Traceback (most recent call last):
  File "2 - pll - arithmetic.py", line 9 , in< module >
    print('연산식: ', expression, '결과:', eval(expression))
  File "<string>" ', line 1
    3 // * 3
         ^
SyntaxError: invalid syntax
```

⚠ 프로젝트 Lab 2

여러 진수를 표준 입력한 수를 여러 진수로 출력

난이도 실전

가장 먼저 변환할 수가 2진수, 8진수, 10진수, 16진수 중에 무엇인지 입력 받는다.

그런 다음, 표준 입력한 수의 2진수, 10진수, 16진수를 출력하는 프로그램을 작성하자.

문제 이해 (Understanding)

처음에 입력 받는 것은 2, 8, 10, 16 중 하나이며 진수를 결정할 기수다.
다음은 변환할 정수를 정수를 입력 받은 후 이미 입력 받은 기수를 사용해
10진수로 변환한다.
이후 적정한 함수를 사용해 각각 2진수, 8진수, 10진수, 16진수로 출력한다.

⚠ 프로젝트 Lab 2

여러 진수를 표준 입력한 수를 여러 진수로 출력

난이도 실전

가장 먼저 변환할 수가 2진수, 8진수, 10진수, 16진수 중에 무엇인지 입력 받는다.

그런 다음, 표준 입력한 수의 2진수, 10진수, 16진수를 출력하는 프로그램을 작성하자.

설계 (Design)

알고리즘(Algorithm)

① 무슨 진수인지 입력 받는다.
- 변수 base에 입력,
  함수 input()과 int() 사용

② 위의 진수로 적정한 정수 하나를 입력 받는다.
- 기수 base를 사용한 적정한 메시지 출력으로 정수 입력

③ 입력된 정수의 2진수, 8진수, 16진수를 출력한다.
- 함수 bin(), oct(), hex()

표준 입출력 샘플

```
입력할 정수의 진수(base)는? 2
2진수 정수 입력 >> 1011
2진수: 0b1011
8진수: 0o13
10진수: 11
16진수: 0xb
```

⚠ 프로젝트 Lab 2

+ 구현(Implementation)

```
1. # 진수와 그에 맞는 정수를 입력 받아 2진수, 8진수, 10진수, 16진수 출력
2. base = int(input('입력할 정수의 진수(base)는? '))
3. invar = input(str(base) + '진수 정수 입력 >> ')
4. data = int(invar, base) # 입력 문자열을 base 진수로 변환
5. # 여러 진수로 출력
6. print('2진수:', bin(data))
7. print('8진수:', oct(data))
8. print('10진수:', data)
9. print('16진수:', hex(data))
```

⚠ 프로젝트 Lab 2

+ 구현(Implementation)

테스트와 실행 결과 (Testing & Simulation)

```
입력할 정수의 진수(base)는? 16
16진수 정수 입력 >> 1f
2진수: 0b11111
8진수: 0o13
10진수: 31
16진수: 0x1f
```

```
입력할 정수의 진수(base)는? 8
8진수 정수 입력 >> 9
Traceback (most recent call last):
  File "2-pl02-numconvert.py", line 6 , in <module>
    data = int(invar, base) # 입력 문자열을 base 진수로 변환

ValueError: invalid literal for int() with base 8: '9'
```

실행오류

⚠ 함수와 내장 함수

… 자료 변환 함수 str( ), int( ), float( ):
정수에서 각각의 진수로 변환 함수 bin( ), oct( ), hex( )

… 함수 int()의 활용:
여러 진수 상수 형태 문자열을 10진수 변환

… 함수 int('1010', 2)의 활용:
문자열 '1010'을 2진수로 이해해 10진수로 반환

… 함수 int('strnum') 또는 int('strnum', 10):
10진수 형태의 문자열을 10진수 정수로 변환

⚠ 깃허브

⚠ 프로그래밍 실습

… 정수나 실수를 입력 받아 다양한 연산을 수행하는 프로그래밍
… 진수를 정한 후 해당 진수를 입력받아 다양한 진수 형태로 출력하는 프로그래밍
… 표준입력의 이해와 함수 input() 활용