Python Basic

표준 모듈(Standard Modules)

모듈(Module)

- 모듈
  - 주요 기능들을 묶어서 파이썬 코드로 작성한 스크립트 파일.
- 표준 모듈
  - 자주 사용하는 기능을 미리 작성한 스크립트 파일.
  - 파이썬 설치 시 같이 설치됨.
- 표준 라이브러리
  - 표준 모듈의 집합.

모듈 사용법

- import 모듈명
  - '모듈명.변수명(또는 함수명)'의 형태로 사용.

    ex) import math; print(math.pi)

- from 모듈명 import 변수명(또는 함수명)
  - 해당 모듈에서 특정 변수(또는 함수)만 가져와서 사용하는 경우.

    ex) from math import pi; print(pi)

- as 별칭
  - 모듈명, 변수명, 함수명에 별칭을 지정하여 사용하는 경우.

    ex) import math as m; print(m.pi)

    ex2) from math import pi as p; print(p)

math 모듈

► math 모듈

상수	설명	상수	설명
pi	원주율 상수	e	자연 대수 상수
inf	무한대 값	nan	숫자가 아닌 값

함수	설명	함수	설명
sqrt(x)	x의 제곱근	pow(x, y)	x의 y승
ceil(x)	x의 소수점 올림	floor(x)	x의 소수점 내림
fabs(x)	x의 실수형 절대값	trunc(x)	x의 소수점 이하 버림
log(x)	x의 자연 로그	log(x, base)	base에 대한 x의 로그
log10(x)	10에 대한 x의 로그	gcd(a, b)	a, b의 최대공약수

time 모듈

► time 모듈

함수	설명	함수	설명
time()	에폭(Epoch) 시간	ctime()	현재 시간
sleep(숫자)	프로그램을 숫자 초만큼 멈춤		

► datetime 모듈

클래스	함수	설명
datetime	now(), today()	현재 시간

time 모듈

► time 모듈

```
import time

print(time.time())
print(time.ctime())
```

```
import datetime

nw = datetime.datetime.now()
print("%d년 %d월 %d일 %d시 %d분 %d초" \
% (nw.year, nw.month, nw.day, nw.hour, nw.minute, nw.second))
```

time 모듈

► time 모듈

```
import time

def func1():
    list = []
    for i in range(100000):
        list.append(i)
    return list

def func2():
    list = [i for i in range(100000)]
    return list

start = time.time(); func1(); end = time.time()
print("func1 performance time:", end - start, "s")

start = time.time(); func2(); end = time.time()
print("func2 performance time:", end - start, "s")
```

time 모듈

- time 모듈

```
import time

print("Countdown!!!")
time.sleep(1)
for i in range(1, 4):
    print(4 - i)
    time.sleep(1)
print("The End!")
```

```
import time

for c in "Python is fun . . .":
    print(c, end="")
    time.sleep(0.4)
```

calendar 모듈

► calendar 모듈

함수	설명
calendar(년도)	전달받은 년도의 달력을 반환
prcal(년도)	전달받은 년도의 달력을 출력
month(년도, 월)	전달받은 년도의 해당 월의 달력을 반환
prmonth(년도, 월)	전달받은 년도의 해당 월의 달력을 출력
isleap(년도)	전달받은 년도가 윤년인지 여부를 반환
weekday(년, 월, 일)	전달받은 날의 요일을 출력(0-6 ~ 월-일)

calendar 모듈

- calendar 모듈

```
import calendar

print(calendar.calendar(2020))
print(calendar.month(2020, 12))
calendar.prcal(2020)
calendar.prmonth(2020, 12)
```

```
import calendar

year = int(input("년도를 입력하세요:"))
print("윤년입니다." if calendar.isleap(year) else "평년입니다.")
```

```
import calendar

weekdays = ["월", "화", "수", "목", "금", "토", "일"]
day = calendar.weekday(2020, 12, 31)
print(weekdays[day])
```

random 모듈

► random 모듈

함수	설명
random()	0.0 이상 1.0 미만의 실수 생성
uniform(a, b)	a 이상 b 미만(혹은 이하)의 임의의 실수 생성
randrange(stop)	stop 미만의 임의의 정수 생성
randint(a, b)	a 이상 b 이하의 임의의 정수 생성
choice(seq)	시퀀스의 임의의 요소를 리턴
sample(seq, n)	시퀀스의 임의의 요소 n개를 리스트로 묶어서 리턴
shuffle(seq)	시퀀스의 요소의 순서를 무작위로 섞음

► random 모듈

```
import random

menu = ["김밥", "라면", "떡볶이", "순대", "튀김", "만두"]
print("오늘 뭐 먹지?", random.choice(menu))
```

```
import random

lotto = random.sample(range(1, 46), 6)
lotto.sort()
print("당첨 번호:", lotto)
```

decimal 모듈

- Decimal 클래스
  - 실수의 오차를 없애는 클래스.
  - 문자열 또는 튜플을 전달받아 Decimal 객체로 리턴.
  - Decimal 객체는 실수형과 연산할 수 없음.

```
import decimal

print(0.1 + 0.2)
print(decimal.Decimal("0.1") + decimal.Decimal("0.2"))

print(decimal.Decimal(123))
print(decimal.Decimal("3.14"))
print(decimal.Decimal("3.14e3"))
print(decimal.Decimal((0, (3, 1, 4), -2))) # (부호, (표현할 숫자), 자리 수)
print(decimal.Decimal("-infinity"))
print(decimal.Decimal("-0"))
print(decimal.Decimal("nan"))
```

decimal 모듈

- 표기 방식 변경
  - setcontext() 함수를 이용해 표기 방법을 바꿀 수 있음.

컨텍스트	설명
BasicContext	유효 자리수 9, ROUND_HALF_UP 처리
ExtendedContext	유효 자리수 9, ROUND_HALF_EVEN 처리
DefaultContext	유효 자리수 28, ROUND_HALF_EVEN 처리

```
import decimal

a = decimal.Decimal("1111111111")
b = decimal.Decimal("1111111111")
decimal.setcontext(decimal.BasicContext)
print(a * b)

decimal.setcontext(decimal.DefaultContext)
print(a * b)
```

fractions 모듈

- Fraction 클래스
  - 분수를 표현하는 클래스.
  - 분자, 분모를 전달받아 Fraction 객체로 리턴.
  - 실수와 연산 시 실수형으로 변환됨.

```
import fractions

a = fractions.Fraction(3, 10)
b = fractions.Fraction(-2, 20)
print(a)
print(b)
print(a + b)
print(a + 1)
print(b - .2)
```

array 모듈

- array 클래스
  - 다른 언어의 전통적인 배열을 구현하는 클래스.
  - 자료형과 초기값을 전달받아 array 객체로 리턴.
  - 대량의 자료를 메모리 낭비 없이 저장하고 고속으로 액세스 가능.

```
arr = array.array("i", [i for i in range(90) if i % 11 == 0])

print(arr[7])
print(arr[3:6])

arr.append(99)
print(arr)
arr.insert(3, 123)
print(arr)

arr.remove(123)
print(arr)
del arr[0]
print(arr)
```

array 모듈

- 자료형 코드
  - array 객체를 생성할 때 자료형을 지정하기 위해 전달하는 코드.
  - 소문자는 부호 있는 타입, 대문자는 부호 없는 타입.

코드	자료형	설명
b, B	char	1 byte 정수
h, H	short	2 byte 정수
i, I	int	2 byte 정수
l, L	long	4 byte 정수
q, Q	long long, __int64	8 byte 정수
f	float	4 byte 실수
d	double	8 byte 실수