

내장함수: 메소드와는 다르게 귀속이 없음, print()

리터럴(값): 100 90.8 "ab" True 날것

(객체)변수: 리터럴이 변수(키) 안에 들어가면 value(값)라고 부름(리터럴을 키에 담으면 value)

메소드: 클래스 안에 선언된 함수, Test.prn()

클래스: 사용자 자료형, class Test{ function prn(){} }

모듈: .py, .py가 파이썬 문법에 맞는지 안맞는지 검증해서 맞으면 .pyc를 생성, 검증된 .pyc를 가지고 있는 것이 모듈, 검증이 돼야 실행이 되니까

패키지: 꾸러미, 폴더 단위

식별자: (객체)변수 함수 메소드 클래스 모듈 등을 구분하기 위해서 붙이는 이름, unicode 형식, 변수 자료형 서브루틴 등을 가리키는 토큰, _로도 시작(private)가능, __로 끝나는 식별자는 특별한 이름을 의미, 대소문자 구분, 클래스명은 시작하는 대문자를 제외하고 모두 소문자, @\$% 등의 문장 부호는 사용하지 않음, 키워드(내장단어)를 제외하고 정함

tuple : 값으로 구분, 인덱스로 호출, a[0]

list : 인덱스로 구분, 인덱스로 호출, a[1]

dict : 키로 구분, 키로 호출, a['apple']

비트: 0과 1로 나타내는 단위

8bit = 1byte = 00000000 1byte당 주소가 생성 32bit일 때 4개의 주소 중 시작주소로만 탐색

	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	0
10	0	0	0	0	1	0	1	0

32bit(기본) = 4byte = 32int

00000000 00000000 00000000 00000000

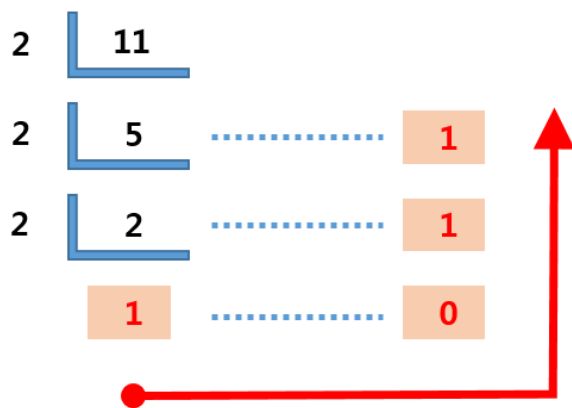
64bit = 8byte = 64int

0xFB + 0b10 # 16진수 + 2진수 / 251 + 2

8진수는 8^3 2진수 → 10진수

$$\begin{aligned} 1011 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\ &= 8 + 0 + 2 + 1 \\ &= 11 \end{aligned}$$

10진수 → 2진수 8진수는 8로 나누고 16진수는 16으로 나눔



문자: 2, 8, 16진수를 문자로 변환할 때 %나 %x를 사용 %d(8진) %x(16진)

* 아스키(ASCII)코드: 문자를 2진수로 표현하는 규칙

- "A"를 입력하면(10진수로 65) 컴퓨터는 2진수로 바뀌서 01 000 001로 기억
- 101(8진 3자리로꿍어) "%101"를 입력하면 "A" 반환
- 41(16진 4자리로꿍어) "%x41"를 입력하면 "A" 반환

문자열 앞에 b를 주면 byte형태로 변환시켜줌 a = b'cde1'

stream : byte단위로 데이터를 주고받는 통로

연산자: operator, 대상(피연산자 operand)을 두고 계산을 할 때 사용하는 문자, 연산자와 피연산자는 주어진 식을 계산하고 결과를 리턴한다

- 산술연산자: + * % ** //
- 비교(관계)연산자: >=
- 논리연산자: &(and) |(or) ~(not) ^(XOR 서로다를때만True)

두개의 연산이 있을 경우 and 연산은 첫연산의 결과가 False이면 연산수행하지 않고 False를 리턴한다
두개의 연산이 있을 경우 or 연산은 첫연산의 결과가 True이면 연산수행하지 않고 True를 리턴한다

- 비트연산자: False는 0으로 True는 1로, Bitwise and / Bitwise or / Bitwise not / Bitwise XOR / Bitwise right shift / Bitwise left shift를 이용해서 결과(bit)를 리턴하는 연산자
10|4 1010|0100 리턴14 ~10는 -10-1이므로 -11을 리턴
- 시프트연산자: >>(오른쪽부터 없애기) <<(왼쪽부터 없애 후 오른쪽에 0 채우기) 10>>2에서 10의 2진수인 1010에서 마지막10 없애면 10이 남고 이것의 10진수인 2를 리턴
- 대입연산자: +=
- 기타연산자: is not not in 1 not in [2, 7, 0]의 리턴은 True