

# PYTHON BASIC

## 변수와 자료형

```
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
elif_operation == "MIRROR Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end --add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob

#mirror_ob.select = 0
```

# 변수(Variable)

- 동적 타이핑(Dynamic typing) 적용
  - 변수 선언 시 자료형을 명시하지 않음
  - 변수에 데이터 할당 시 자료형이 **자동으로 결정**
  - **type()** 함수로 변수의 자료형 확인 가능
- 모든 변수가 **객체**(참조 자료형)
- 식별자(Identifier) 규칙
  - 알파벳, 숫자, 언더스코어(\_)
  - 숫자로 시작할 수 없음
  - 예약어(Keyword) 사용 불가

# 예약어 (Keyword)

if	else	elif	is
for	while	break	continue
True	False	None	del
and	or	not	in
def	return	nonlocal	global
try	except	finally	raise
import	from	class	pass
yield	assert	with	as

# 정수형

- 소수점 없이 표현되는 수
- 2진수, 8진수, 16진수 표기 가능
  - 2진수: 숫자 앞에 '0b' 표기, `bin()` 함수로 2진수 문자열로 변환
  - 8진수: 숫자 앞에 '0o' 표기, `oct()` 함수로 8진수 문자열로 변환
  - 16진수: 숫자 앞에 '0x' 표기, `hex()` 함수로 16진수 문자열로 변환

# 실수형

- 소수점과 함께 표현되는 수
- 고정소수점, 부동소수점 표기 가능
  - 고정소수점: ex) 3.14159265
  - 부동소수점: ex) 2.99792458e8

# 복소수형

- 실수와 허수로 표현되는 수
- [실수부의 숫자] + [허수부의 숫자]j
- .real: 실수부의 숫자 반환
- .imag: 허수부의 숫자 반환
- conjugate(): 켤레 복소수 반환
  - 켤레 복소수: 서로 곱해서 실수가 되는 수

# 문자열

- 문자들의 집합
- 유니코드 지원
- 값 변경 불가(Immutable)
- 표현 방식
  - ① "문자열"
  - ② '문자열'
  - ③ ""문자열""
  - ④ '''문자열'''
- 따옴표 앞에 r을 표기하여 raw 문자열 작성 가능
- + 연산자로 문자열 연결, \* 연산자로 문자열 반복
  - 문자열과 숫자 간의 연결 불가

# Escape Sequence

문자열	의미
\\	백슬래시(\)
\'	작은 따옴표(')
\"	큰 따옴표(")
\a	ASCII 벨(BEL)
\b	ASCII 백스페이스(BS)
\f	ASCII 폼 피드(FF)
\n	ASCII 라인 피드(LF)
\r	ASCII 캐리지 리턴(CR)
\t	ASCII 탭(TAB)



# ord(), chr() 함수

- ord(): 전달받은 문자에 대한 **아스키 코드 숫자 반환**
- chr(): 전달받은 숫자에 대한 **아스키 코드 문자 반환**

# 논리형

- 참과 거짓을 나타내는 데이터
- 논리 연산자와 함께 조건문에서 사용
- 다른 자료형에서 값에 따라 참 또는 거짓을 나타낼 수 있음

# 값에 따른 참/거짓 판별

자료형	값	참/거짓
정수형	1	참
정수형	0	거짓
실수형	1.0	참
실수형	0.0	거짓
문자열	"python"	참
문자열	""	거짓
리스트	[1, 2, 3]	참
리스트	[]	거짓
None	None	거짓

# None

- 아무 것도 아니다, 하나도 없다는 뜻의 객체
- Java 등 다른 언어의 **null**과 유사
- `==`, `!=` 연산자보다 **is**, **is not** 연산자로 비교

# 형 변환

- `int()`: 정수형으로 형 변환하는 함수
- `float()`: 실수형으로 형 변환하는 함수
- `complex()`: 복소수형으로 형 변환하는 함수
- `str()`: 문자열로 형 변환하는 함수
- `bool()`: 논리형으로 형 변환하는 함수