

# PYTHON BASIC

연산자

```
mirror_mod.use_x = False
mirror_mod.use_y = False
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
```

# 연산자(Operator)

- 특정 **계산 규칙**을 나타내는 기호
- 변수와 값에 대해 연산을 수행한 뒤 **결과를 반환**
- **우선순위**가 존재

# 대입 연산자

연산자	의미
=	연산자의 오른쪽에 있는 값을 왼쪽에 있는 변수에 대입

# 산술 연산자

연산자	의미
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
**	거듭제곱
//	몫 구하기
%	나머지 구하기

# 복합 대입 연산자

연산자	의미
<code>+=</code>	덧셈한 <b>결과</b> 를 변수에 <b>대입</b>
<code>-=</code>	뺀셈한 <b>결과</b> 를 변수에 <b>대입</b>
<code>*=</code>	곱셈한 <b>결과</b> 를 변수에 <b>대입</b>
<code>/=</code>	나눗셈한 <b>결과</b> 를 변수에 <b>대입</b>
<code>**=</code>	거듭제곱한 <b>결과</b> 를 변수에 <b>대입</b>
<code>//=</code>	몫 구하기한 <b>결과</b> 를 변수에 <b>대입</b>
<code>%=</code>	나머지 구하기한 <b>결과</b> 를 변수에 <b>대입</b>

# 비교 연산자

연산자	의미
>	왼쪽의 값이 오른쪽의 값보다 <b>큰지</b> 검사
>=	왼쪽의 값이 오른쪽의 값보다 <b>크거나 같은지</b> 검사
<	왼쪽의 값이 오른쪽의 값보다 <b>작은지</b> 검사
<=	왼쪽의 값이 오른쪽의 값보다 <b>작거나 같은지</b> 검사
==	왼쪽의 값이 오른쪽의 값과 <b>같은지</b> 검사
!=	왼쪽의 값이 오른쪽의 값과 <b>다른지</b> 검사

# 논리 연산자

연산자	의미
and	왼쪽의 값과 오른쪽의 값이 모두 True면 True 반환(논리곱)
or	왼쪽의 값과 오른쪽의 값이 모두 False면 False 반환(논리합)
not	True면 False, False면 True 반환(부정)

# 객체 비교(Identity) 연산자

연산자	의미
is	왼쪽의 객체와 오른쪽의 객체가 <b>같은 메모리를 참조하는지</b> 검사
is not	왼쪽의 객체와 오른쪽의 객체가 <b>다른 메모리를 참조하는지</b> 검사



# Equality vs Identity

## ▪ Equality

- 동등성
- 같은 값을 갖는지에 대한 여부
- `==`, `!=` 연산자로 비교

## ▪ Identity

- 동일성
- 같은 객체를 참조하는지에 대한 여부
- `is`, `is not` 연산자로 비교

# 멤버십 연산자

연산자	의미
in	왼쪽의 요소가 오른쪽의 객체에 <b>포함되는지</b> 검사
not in	왼쪽의 요소가 오른쪽의 객체에 <b>포함되지 않는지</b> 검사