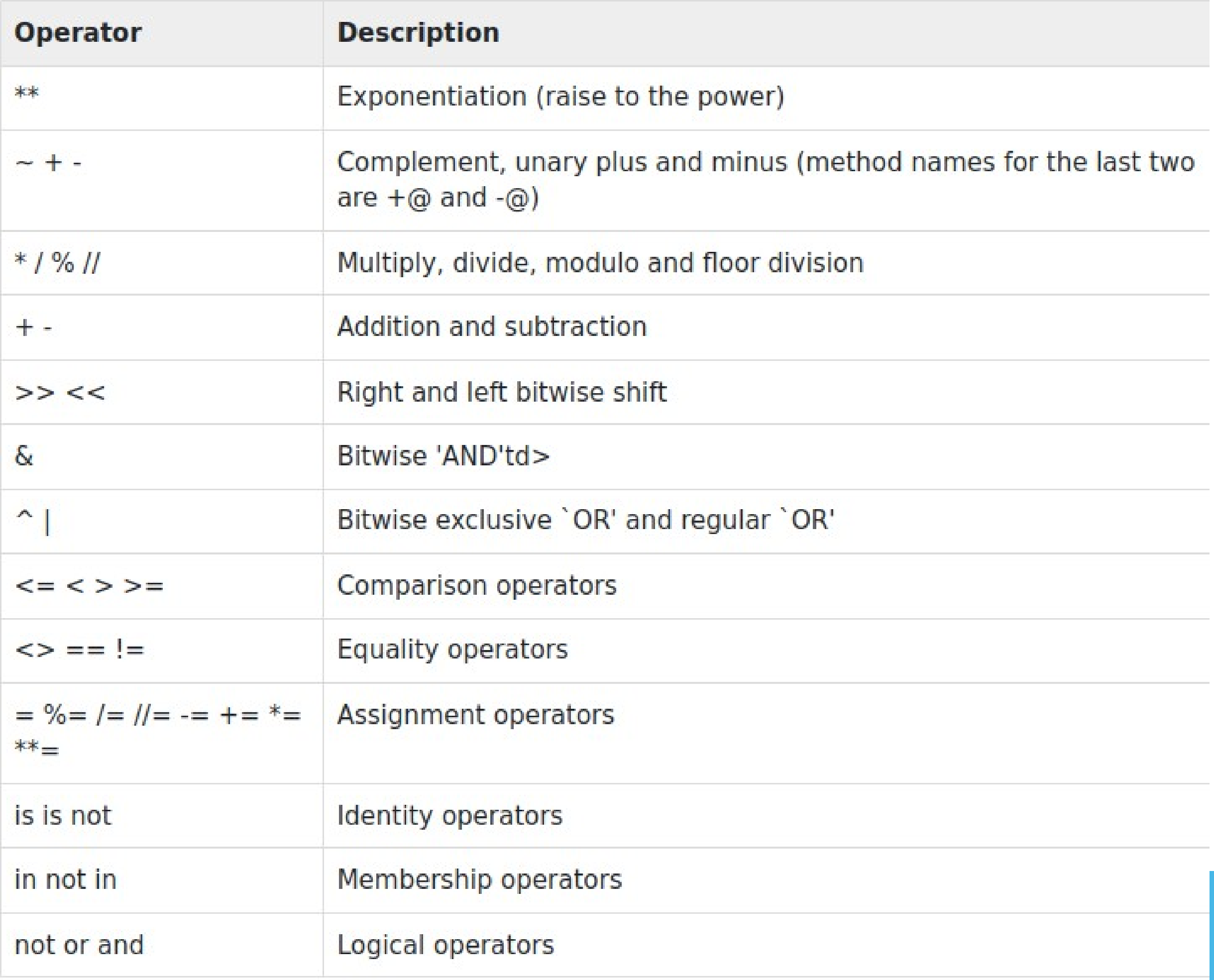
# Problems on Operator Precedence

# Operator Precedence



| Operator | Description |
|---|---|
| ** | Exponentiation (raise to the power) |
| ~ + - | Complement, unary plus and minus (method names for the last two are +@ and -@) |
| * / % // | Multiply, divide, modulo and floor division |
| + - | Addition and subtraction |
| >> << | Right and left bitwise shift |
| & | Bitwise 'AND'td> |
| ^ \| | Bitwise exclusive `OR' and regular `OR' |
| <= < > >= | Comparison operators |
| <> == != | Equality operators |
| = %= /= //= -= += *= **= | Assignment operators |
| is is not | Identity operators |
| in not in | Membership operators |
| not or and | Logical operators |

Note: ‘and’ has got higher precedence than ‘or’

# Problem 1

Solve 25 + 10 * 20

# Solution 1

25 + 10 * 20 is calculated as 25 + (10 * 20)

and not as (25 + 10) * 20

- >>> 25 + 10 * 20
- 225

# Problem 2

What will be output of:

```
print(3*1**3)
```

# Solution 2

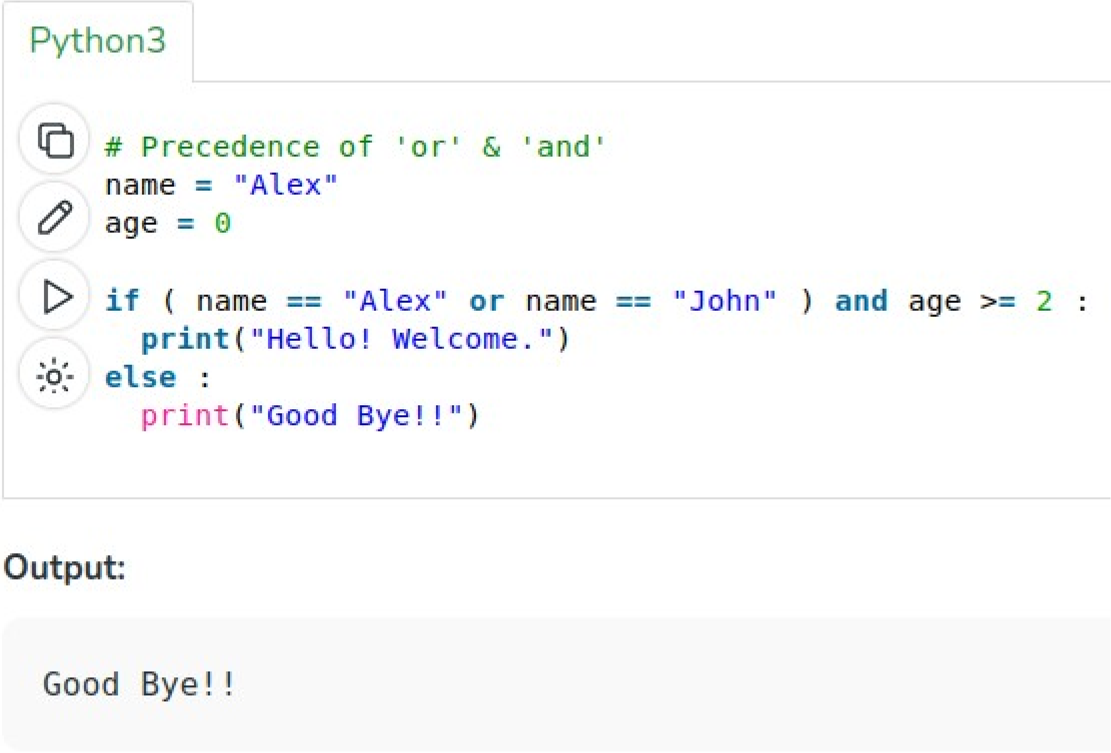
3*1**3 is equivalent to: 3 * (1**3)

```
(base) ashish@ashish:~$ python
Python 3.9.13 (main, Aug 25 2022, 23:26:10)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 3*1**3
3
```

# Problem 3

```
# Precedence of 'or' & 'and'
name = "Alex"
age = 0
if ( name == "Alex" or name == "John" ) and age >= 2 :
  print("Hello! Welcome.")
else :
  print("Good Bye!!")
```

# Solution 3



```python
# Precedence of 'or' & 'and'
name = "Alex"
age = 0

if ( name == "Alex" or name == "John" ) and age >= 2 :
  print("Hello! Welcome.")
else :
  print("Good Bye!!")
```

**Output:**

```
Good Bye!!
```

# Problem 4

```
# Precedence of 'or' & 'and'
name = "Alex"
age = 0
if name == "Alex" or name == "John" and age >= 2 :
  print("Hello! Welcome.")
else :
  print("Good Bye!!")
```

# Solution 4 (Part 1)

Condition: *name == "Alex" or name == "John" and age >= 2*

It is equivalent to:

*name == "Alex" or (name == "John" and age >= 2)*

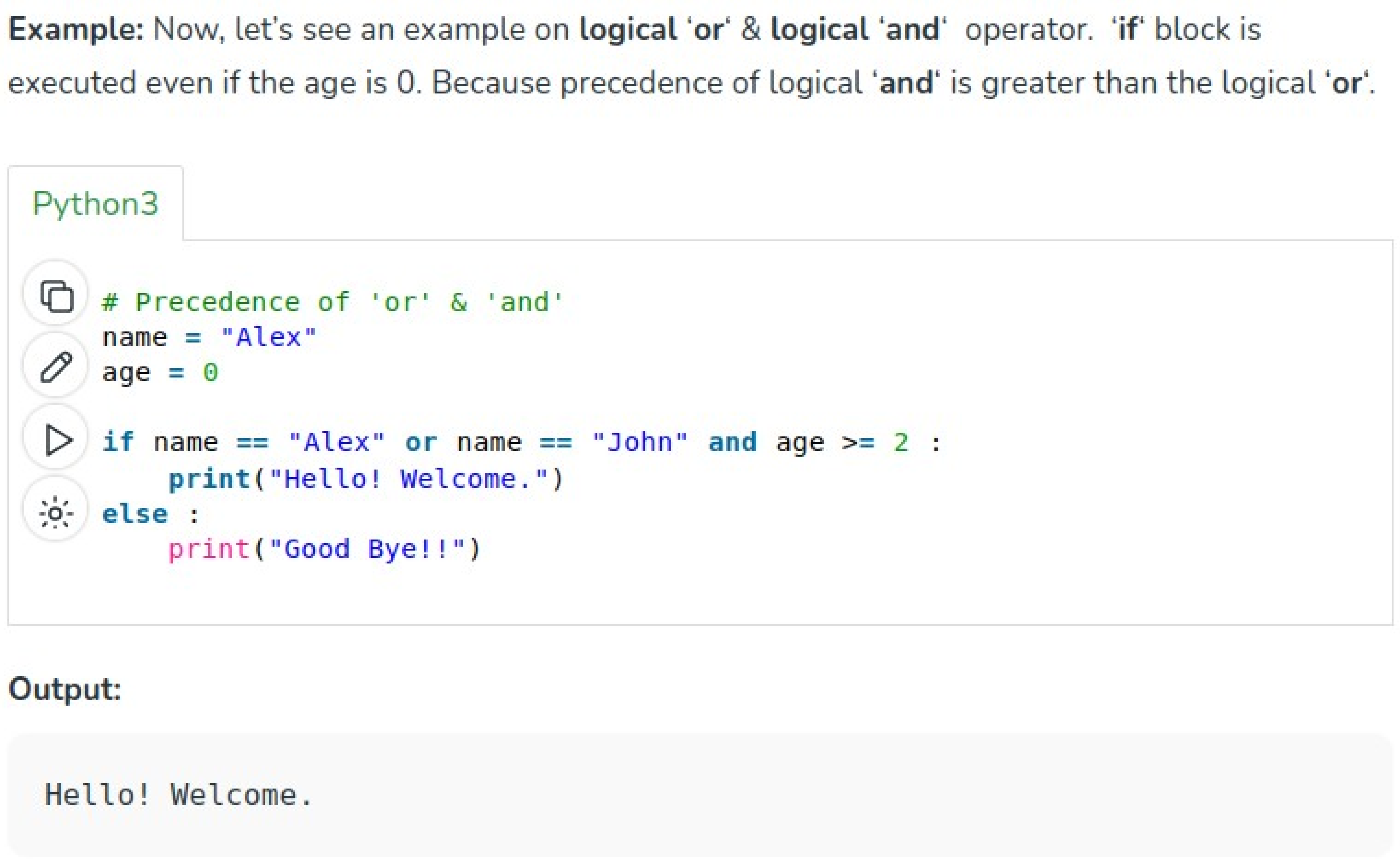
For:

name = "Alex"

age = 0

What will it print(): True or False?

# Solution 4 (Part 2)



**Example:** Now, let’s see an example on **logical** ‘**or**‘ & **logical** ‘**and**‘ operator. ‘**if**‘ block is executed even if the age is 0. Because precedence of logical ‘**and**‘ is greater than the logical ‘**or**‘.

```
# Precedence of 'or' & 'and'
name = "Alex"
age = 0

if name == "Alex" or name == "John" and age >= 2 :
    print("Hello! Welcome.")
else :
    print("Good Bye!!")
```

**Output:**

```
Hello! Welcome.
```

# Operator Associativity

Operator Associativity: If an expression contains two or more operators with the same precedence then Operator Associativity is used to determine. It can either be Left to Right or from Right to Left.

Example: ‘*’ and ‘/’ have the same precedence and their associativity is Left to Right, so the expression “100 / 10 * 10” is treated as “(100 / 10) * 10”.

# Problem 5

```
print(100 / 10 * 10)
print(5 - 2 + 3)
print(2 ** 3 ** 2)
```

# Solution 5

```
# Left-to-right associativity
# 100 / 10 * 10 is calculated as
# (100 / 10) * 10 and not as 100 / (10 * 10)
print(100 / 10 * 10)

# Left-to-right associativity
# 5 - 2 + 3 is calculated as
# (5 - 2) + 3 and not as 5 - (2 + 3)
print(5 - 2 + 3)
```

```
# right-to-left associativity
# 2 ** 3 ** 2 is calculated as
# 2 ** (3 ** 2) and not as (2 ** 3) ** 2
print(2 ** 3 ** 2)
```

```
>>> 2 ** 3 ** 2
512
```

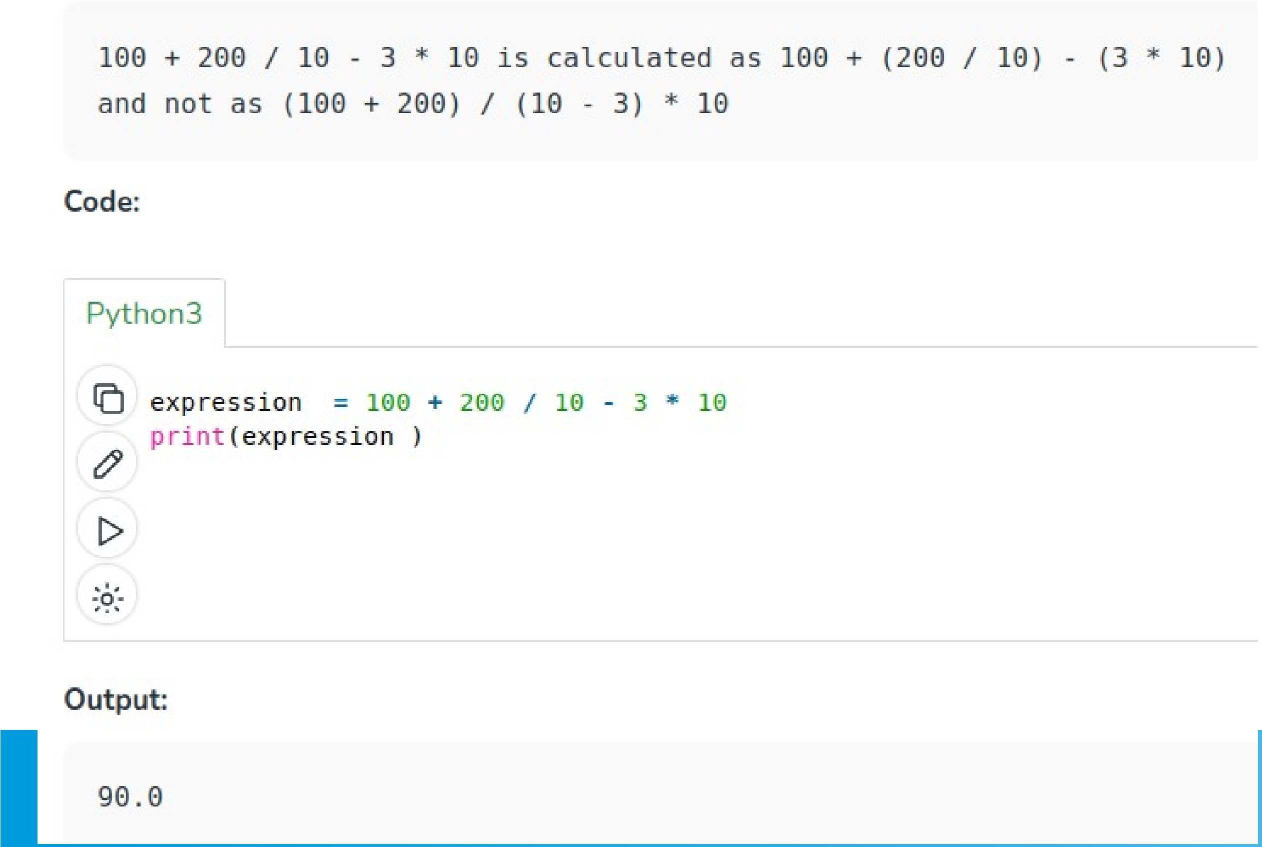
Two to the power (three to the power 2).

# Problem 6

```
print(10 + 10 / 10 - 10 * 10)
print(100 + 200 / 10 - 3 * 10)
```

# Solution 6

```
>>> print(10 + 10 / 10 - 10 * 10)
-89.0
```



```
100 + 200 / 10 - 3 * 10 is calculated as 100 + (200 / 10) - (3 * 10)
and not as (100 + 200) / (10 - 3) * 10
```

**Code:**

```python
expression  = 100 + 200 / 10 - 3 * 10
print(expression )
```

**Output:**

```
90.0
```