

Content Covered



- Assignment
- Comparison
- Logical
- Bitwise
- Membership
- Identity
- Conditional

Arithmetic

- Numeric Operators:
 - + => Addition
 - - => Subtraction
 - * => Multiplication
 - / => Division
 - // => Floor Division
 - % => Modulus
 - ** => Exponent (Power)

Assignment

- Used to assign value to variable
- Operators:
 - `=` , `+=` , `-=` , `*=` , `/=` , `%=` , `//=` , `**=`
 - `var_1 = var_2 = var_3 = 5`
 - `var_1, var_2 = 3, 4`
 - `var_1, var_2 = var_2, var_1`

Comparison

- Compare two variable or values
- Result of is bool & used in control flow
- Example:
 - `a = 5`
 - `b = 10`
 - `print(a < 5)`
 - `print(a > b)`
 - `print(a >= b)`
 - `print(a == b)`

Logical

- Used to combine multiple conditions
- Precedence not > and > or
- Use bracket to change precedence
- and, or, not
- Syntax: condition_1 and condition_2

Bitwise

- Perform logical operation but bitwise
- Operators are: `&`, `|`, `~`, `<<`, `>>`
- Example:
 - `a = 1`
 - `b = 2`
 - `print("Bit wise and {}, Bit wise or {}".format(a&b, a | b))`

Membership

- Used to check if character or string lies in a variable
- Result is bool
- in and not in are used to check membership
- Example:
 - `print('a' in 'apple')`
 - `print('apple' in ['apple', 'orange'])`
 - `print('b' not in 'apple')`

Identity

- is and is not is used to check if value is None or not
- Result is bool
- Example
 - `a = None`
 - `print(a is None)`

Conditional

- Different logic for different condition
- Condition is statement giving bool output
- Can have multiple block of if statements
- Indentation is required
- Syntax:
 - `if {condition}:`
 `--- statements ---`
 - `elif {condition}:`
 `--- statements ---`
 - `else:`
 `--- statements ---`



Exercise