Do's & Don'ts for Writing Variable Names

1) write meaningful names.

2) **camel case** multi-word names, capitalize first letter of each word except the 1st.

e.g ⇒ customerName, orderTotal.

3) **snake case** separate words with underscores.

customer_name, order_total.

Dont's

1) use of short names.

eg. a, b, c. etc

2) use of all caps / all lowercase

eg. CUSTOMER / customer. etc

3) start variable names with numbers / underscores.

eg. 90customer / _customers. etc

⇒ code easier to understand, debug & maintain.

⇒ by following these do's & dont's you can improve quality & readability of your code

code

```
# Add 5 to the value of age
age_with_5_years = age + 5

# check if age is greater than 20.
is_adult = age > 20

# Calculate the age in 10 years
future_age = age + 10

# Print results.
print("Age with 5 years:", age_with_5_years)
print("Is adult:", is_adult)
print("Future age:", future_age)
```

explanation :-

we used the '+' operator to add 5 years to age, the '>' operator to check if age is greater than 20, and the '+' operator again to calculate the age in 10 years.

In this ex, we defined 2 variables age & name. We assigned the value 25 to age & the string "codes" to name. Then, we used the print function to display the values of variables.

◎ Operators :-

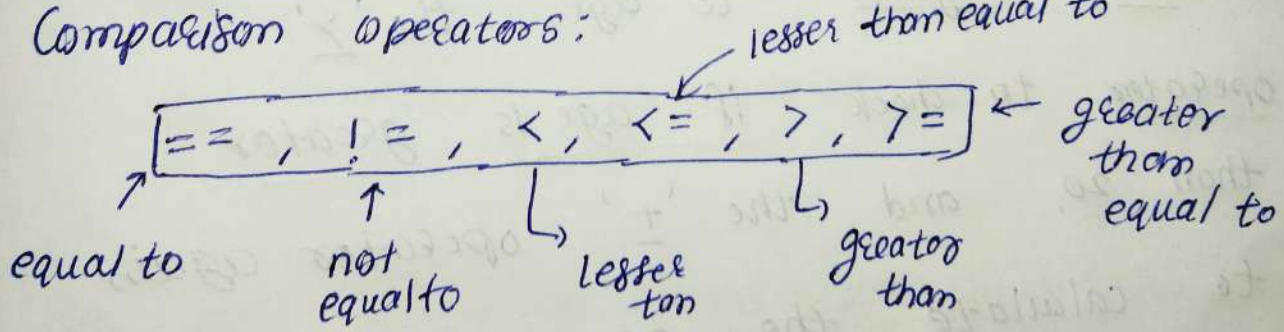
"Symbols used to perform operations on data stored in variables."

Common Python operators

★ Arithmetic operators:

+, -, *, /, % (modulus)

★ Comparison operators:



★ Logical operators:

and, or, not

#lec 4

Rising Coders Era

Variables & operators

⊙ Variables :-

" containers that hold information They allow us to store and access information throughout our program. "

We can think of them as boxes with labels that tells us what they contain.

#code

```
# Define a variable named age & store
# the value 25.
age = 25

# Define a variable named name &
  store the string "coders"
name = "coders"

# Print values of variables
print ("Age", age)
print ("Name:", name)
```