Variable Naming Rules

- Vaviable should have meaningful name.
- It must start with a letter (or) an underscore.

    name, age , _age , id ..

- It contain only alpha- numeric characters (A-Z , a-z , 0-9) and underscore , no other special symboles is allowed. space also not allowed.

ex :- roll_no , stud-id , S1_name , S2_name, length 1, length 2

Multi-word variable names

→ Camel case :-

each word , except the first starts with a capital letter

ex :- my Variable Name

→ pascal case :-

each word start with capital letter

My variable Name

→ Snake case

each word is separated by an underscore

my_variable_name.

exercise

var a = 1 → Invalid

1 = a → Invalid

var_a = 1 → valid.

Primitive Data types in python

→ It tells type of data/value.

→ Data types are actually classes in python & variable are objects of the classes

eg : int is a class

float is a class

var = 4 [Here var is an object of the int class type]

→ python has multiple different data types.

Data types

it , float, complex ⇒ numeric type.
string
list , triple, range ⇒ seqence type
dictionary
Boolean
set

int :-

Contains whole number (+ve (or -ve)

-There is no limit to how long an integer value Can be. constrained by the memory of your system.

eg : a=123
Print (a) => then it will print 123

*-> With any prefix the given number would be Consider as decimal number

but with prefix

Ob (or OB [zero + b / zero + B] => Binary

Oo (or OO [zero + lower letter o / zero + upper letter O] => octal

Ox (or OX [zero + x / zero + X] => hexadecimal.

example:-

Print (Ob11) => will print 3

print (0o11) => will print 9

Print (0x11) => will print 17

-> check the type then use type () function
ex:- var = 4

```
print(type(var))
```
-> It will print <class, 'int'>

<u>float</u>:-

decimal numbers

eg: 4.2, 4.0, 2

<u>Strings</u>:-

Sequence of character

[illegible] "Sir" (or) 'sir'

print("Sir"[0]) => will print S

Print ("Sir"[3]) => will print r

<u>exercise</u>

name = ["Jenny's lectures "CS/IT" "
print(name)]

<u>Sol</u>:

```
name = "Jenny's lectures \"CS/IT\" "
print(name)
```

<u>Boolean</u> :- 2 possible values

- True
- False

[should make T & F ~~st~~ are in captial] eg.1 : var = True

Print (var) → will print True

Print (type (var)) → will print <class, 'bool'>

eg. 2

a = 1

b = 2

var = a < b

Print (var) → will print True

Print (type (var)) → will print <Class, 'bool'>

Type checking & type Conversion

eg

Print (len ("Mahi")).

it will print length of the String

i.e it will give "4".

→ so len is a function which accepts Strings

Print(len (123))

→ It will give type error

if we print (len("123")).

it will print 3

example

length = len (input ("What is your name ?"))

Print ("your name has" + length + "Characters")

→ It will give the type error as length is of integer type.

→ print(" your name has" + str(length) + " characters")

output:

Yourname has 5 characters

```
int()   → will convert into int
float() → will convert into float
str()   → will convert into string
```

coding exercise

· Take 2 numbers as input from the user & find sum (use input function)

ex.1 -

```
print (10 + 10) → 20
print ("10" + "10") → 1010
print (int ("10") + int ("10")) → 20
print (10 + float ("10.10")) → 20.10
```

ex.2

```
a = 100
b = 12                    ⇒ print 10012
print ( str (a) + str(b))
```

ex: 3

```
name = "Jenny"
print ( 10 + int (name) ) } ⇒ value error.
```

ex4.

```
name = "123"
new_name = int (name)        ] => Print 133
Print (10 + new_name)
```

Operators in python

Operators are used to perform operation on values/variables

eg:- 2+3 [here '+' is operator & 2,3 are operands]

Operators are special symbols eg: +, -, /, * etc.

Types of operators

- Arithmetic Operator
- (Relational) operator
- logical operator
- Assignment Operator
- Bitwise Operator
- Special → identity
  → membership

Arithmetic operations

This are used to perform mathematical operations like add, subtract, multiply, divide.

+ , - , * , / , //, % , **

Precedence & Associativity

paranthesis ( )
Exponent **
Multiplication & division * /
Addition & subtraction + -

} precedence.

5 + 2 * 3 - 1 + 10/5

5+2*3-1+10/5 => 12.0

5+2*(3-1) + 10/5 => 11.0

Assignment Operators

Used to assign values to variables

a = 1

ex: a = a+2   a+ = 2   + =

a = a-1   a- = 1   - =

a /= 2   * =

/ =

% =

// =

** =

a, b, c = 5, 8, 9
Print (a, b, c)
Then it will print
5 8 9

Comparison Operator

compares the value either returns true or false according to condition

== < > <= >= !=

eg a = 5

print (a == 5) => true
print (a < 5) => false
Print (a != 5) => false
Print (a > 5) => false
Print (a <= 5) => true
Print (a >= 5) => true

logical Operators

basically used to combine conditional statements

AND OR NOT

ex: a = 5
b = 4

Print (a and b) → returns true (4)

Print (a<5 & b==4) - false

Print (a<=5 and b==4) - true

print (a<5 or b==4) - True

Print (5 and 4) → 4

Print (0 and 4) → 0

Print (0 or 4) → 4

Print (not(a)) → false

a = false

~~true < print~~

and → return true if both statements are true

OR → return True if one of the statement is true

not reverse the result (or negate the result