```python
# Variable Naming Rules:
# 1. Variable names can only contain letters, numbers, and underscores (_).
# 2. Variable names cannot start with a number.
# 3. Variable names are case-sensitive (age, Age, AGE are different variables).
# 4. Variable names cannot be a reserved keyword in Python (e.g., if, else,
while, etc.).
# 5. Variable names should be descriptive and meaningful to improve code
readability.
# 6. Variable names should not contain spaces or special characters (e.g., @, $,
%, etc.).
# 7. Variable names should be kept short but meaningful.

# string (str) → text type
txt = "string value"
txt = 'also string value'

# integer, float, complex (int, float, complex) → number, numeric type
a = 5 # integer
b = 5.5 # float
c = 5 + 3j # complex

# lists, tuples, sets, dictionaries (list, tuple, set, dict) → collection types
my_list = [1, 2, 3] # list
my_tuple = (1, 2, 3) # tuple
my_set = {1, 2, 3} # set
my_dict = {"key": "value"} # dictionary

# boolean (bool) → true/false type
is_true = True # boolean
is_false = False # boolean

# none (NoneType) → null type
none_value = None # none type

# print the type of each variable
x = 5
print(type(x))

# python is case sensitive
a = 5
A = 10
print(a)
print(A)

txt = "batoul"
txt = "Batoul"

# Assigning multiple values to multiple variables in one line
x, y, z = 5, 10, 15
print(x, y, z)
```

```python
x = y = z = 5 # Assigning the same value to multiple variables
v =10
x = y = z = v # Assigning the same value to multiple variables
print(x, y, z)

v =20
x = y = z = v # Assigning the same value to multiple variables
print(x, y, z)

# Assigning from list
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits # Unpacking the list into variables
print(fruits)
print(x, y, z)

# Casting

#casting string to integer
folderName = "10" # 10 is added as type string. this is a text not a number type
folderName = int(folderName)  # we can cast it to int type or any numeric type
print(type(folderName)) # prints: <class 'int'>

# casting integer to float
x= 10
y = float(x)
print(y) # prints: 10.0
print(type(y)) # prints: <class 'float'>
```