**Immutable and Mutable**

**Python Data Types**

* In Python, there are two main data types: mutable and immutable.
* Mutable types include lists, sets, and dictionaries, while immutable types encompass integers, floats, Booleans, and strings.
* Understanding these types is crucial when defining variables and managing memory in programming.

A diagram of a data flow

AI-generated content may be incorrect.

🔁 **Mutable Data Types** (Can be changed after creation)

* These types **allow** you to modify, add, or remove elements without changing their identity (memory location).
* Data types whose reference value can be changed.

A screenshot of a computer

AI-generated content may be incorrect.

**🔒 Immutable Data Types (Cannot be changed after creation)**

* Once these types are created, their **contents cannot be altered**. Any modification creates a **new object**.
* Data types whose reference value cannot be changed.
* Changing a variable's value creates a new object and updates the reference.

A screenshot of a computer

AI-generated content may be incorrect.

Python in PowerShell :

A screen shot of a computer program

AI-generated content may be incorrect.

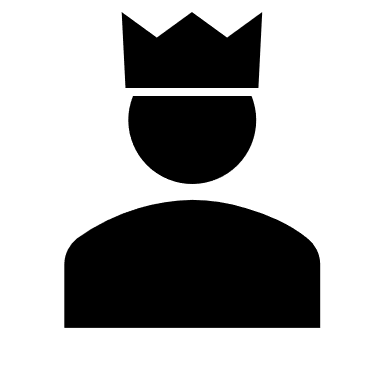
**Memory Management in Python: Key Concepts**

* **Everything is an Object:**  
  In Python, every value is an object stored in memory.
* **References, Not Copies:**  
  Assigning a value to a variable creates a **reference** to the object in memory, not a copy of the data.

A black screen with white text

AI-generated content may be incorrect.

1. **String Assignment and Reassignment**



“Raj ”

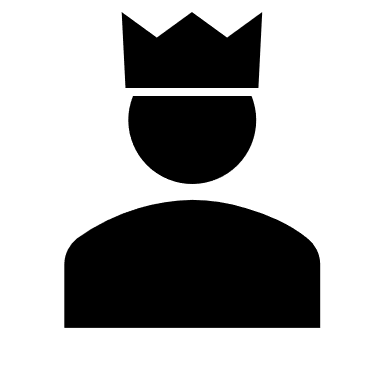
*username*

1. Username first points to raj. A string "Raj " is assigned to the variable username.
2. Printing Username shows “Raj”.

A black background with white text

AI-generated content may be incorrect.

1. Then, username = "Hello Earth" reassigns username to a new string "Hello Earth".

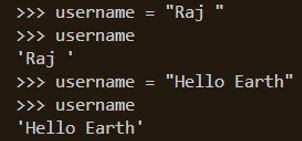


“Raj”

*username*

“Hello Earth”

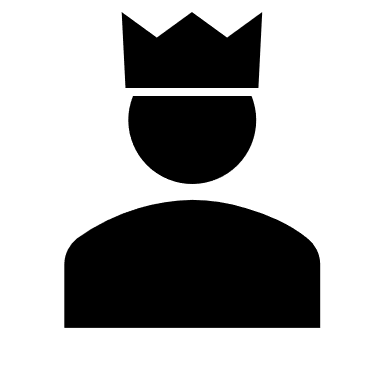
1. *Now, "Hello Earth" is assigned to the same variable username.* *Printing username now shows 'Hello Earth'.*

**

*This* ***replaces*** *the old value. “Raj” is automatically clean up by garbage collector. Since strings are* ***immutable****, a* ***new string object*** *is created and assigned.*

1. ***Integer Assignment and Copy***

“Hello Earth”



*username*

20

X

10

* *Variable x is assigned the value 10.* A black background with white text

  AI-generated content may be incorrect.

Y

* *Then y is set equal to x. y = x makes y reference the same integer object as x (both point to 10).*
* *Printing x and y both show 10.*
* *Both x and y point to the* ***same immutable integer value*** *in memory.*
* *Now x is changed to 20, which means Python creates a* ***new integer object*** *for x.* *x is reassigned to a new integer object 20.*
* *y still references the old integer 10.* *y is still 10\*\* because integers are* ***immutable****, and y was pointing to the old value 10.*
* *Printing y shows 10, confirming that changing x's reference does not affect y.*

***Conclusion***

* *Strings and integers are****immutable****in Python.*

*Once created, their values cannot be changed. Any modification results in a new object being created.*

* *Assignments create references to objects, not copies.*
* *Changing a variable's value reassigns its reference without affecting other variables pointing to the original object.*