20160914

CSC148

REFERENCES, MUTABILITY, ALIASES

" DATA MODEL OF PYTHON "

PYTHON DATA HAS...:

- ID :: IDENTIFIES WHERE THE DATA IS
- TYPE :: DETETIRMINES HOW THE DATA CAN BE USED
- VALUE :: CONTAINS WHAT THE DATA IS

PYTHON INTERPRETER REFERS TO A VALUE.

EXAMPLE

x = 1000
y = 1000

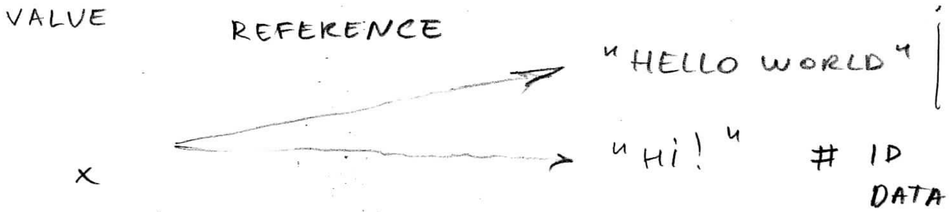
	=	≠
ID		✓*
TYPE	✓	
VALUE	✓	

* SOME OPTIMIZATION TAKES PLACE FOR SOME DATA

MUTABILITY :: THE ABILITY OF THE DATA VALUE TO CHANGE;

EXAMPLE x = " HELLO WORLD " # THE STRING IS IMMUTABLE.

x = " Hi! "



REASSIGNMENT

ID OF THE PREVIOUS DATA ITEM IS STORED AS PART OF THE INTERPRETER IMPLEMENTATION

VARIABLE REASSIGNMENT CHANGES THE REFERENCE NOT THE VALUE

ALIASES NAMES REFERRING TO THE SAME MEMORY LOCATION

EXAMPLE

```
x = "DAVID IS AWESOME"
y = x
```

=> id(x) = id(y)

! REFERENCES ARE INDEPENDENT OF ALIASES

OPERATION == ∵ COMPARISON OF VALUE

OPERATION IS ∴ COMPARISON OF IDS'

EXAMPLE

```
x = "HELLO"
y = "HELLO"
```

=> id(x) ≠ id(y)

IS IS EQUIVALENT TO ID (x) == id(y)

MUTABLE DATA STRUCTURES

LISTS
DICTIONARIES
USER-DEFINED CLASSES

EXAMPLE

```
x=[1, 2,3]
y=[1,2,3]
```

=> id(x)=id(y)

DEFINITION OF MUTABLE DATA CREATES AN ID TO WHICH ALL THE REFERENCES POINT TO.

HENCE, ALL THE REFERENCES ARE UPDATED WHEN THE DATA CHANGES.

EXAMPLE:
COPYING LISTS

```
x= [1,2,3]
y= x [:]
```

FUNCTION PARAMETERS ARE ALIASES FOR ARGUMENT DATA.

FUNCTIONS CAN MUTATE INPUTS

EXAMPLE

```
def mutator (x):
    x[0]=100
lst = [1,2,3]
mutator (lst)
```

=> the data of lst has changed.