Dictionnane Declaration? Suser = { "mame": "salabi",

"age": 19 halus, "sene": M
"shalls": ["Html", "Css"] > user = dict () >user = { } -s Lie Key need to be immutable (list not allowed)
-s Lis Value es all types
-s key must be unique. Suser ["mame"]
Salibio Suser ["mame"][0] Html. Fonctionss > len (d) # who elevents # plus petit clé Smin (d) # plus grand clé

# supprimer elevent essocié ei clé I > man (1) > del 150] # retourer list des clés Slist (d) dict

Dictionnaire Methodes get(): > Loget (Ney, Value)

A Doption, Valen retourner si lock of

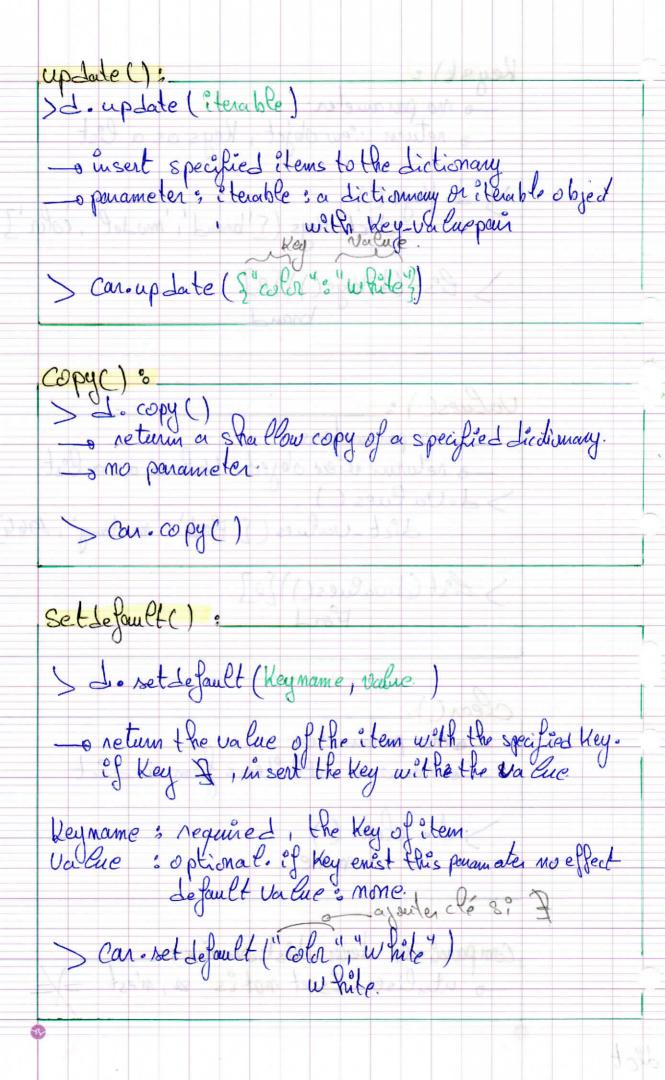
Lofault Value's Mone

Logaines, the key of item you want to return > Can = { "Brand" : "Find", "model" : "Mustang" "gear " : 1964 > n = car. get ("model") > print (x) Mustang. items(): - No parameters > do items () > car = 3" brand": "ford", "model": "Mustang", "year": 1964} > n = car. (tems() doct\_items ([(brand', Fird'), (model ! Hustang'), (
> list (x)[0] # neturn Tuple.
('brand', Fird') dict

> List\_Keys (['brand', 'model', color']) > Est (d. Keys())[0] Values(): \_ b mo parameters -s return view object. Values as a l'est > d. Values () Lict-Values (['Ford', mustang', 1964]) > Eist (dovalues())[0] clear (). - mo parameter. o remove all êtems form à dict > 1. clear () Comparaison deun distinuaires

- utuliser l's et not is &, m'est = /=

dict



from deys(): > I from keys ( Heys, value) o return a Lise timony with the specified keys and Specified Value value & optional, the value for all keys default value : mone. >n = ("Keys", Key2", Key3") 5 de from Keys (n) 5'keys'; none, 'Key2'; none } > d. pop (key name, Lefault Value) - remove specified item from dict. returne de Value to remove - o key mame & required, the key to value removed \_o defaultvalue; optimal, Albhue to return if key }
if key \$\frac{1}{2} et this parameter no specified retur enos Popitem ()3 > 2. popitem () - remove the Past inserted Elem return key-value as Tuple. - o mo parameters

Lict