```
root@Nestor:~# cd ~/iot/lesson10
root@Nestor:~/iot/lesson10# cat hash_value.py
"""
https://docs.python.org/3/using/cmdline.html#envvar-PYTHONHASHSEED
If PYTHONHASHSEED is not set or set to random, a random value is used to to seed the hashes of str and bytes o
bjects.
If PYTHONHASHSEED is set to an integer value, it is used as a fixed seed for generating the hash() of the type
s covered by the hash randomization.
Its purpose is to allow repeatable hashing, such as for selftests for the interpreter itself, or to allow a cl
uster of python processes to share hash values.
The integer must be a decimal number in the range [0,4294967295]. Specifying the value 0 will disable hash ran
domization.

https://www.programiz.com/python-programming/methods/built-in/hash
hash(object) returns the hash value of the object (if it has one). Hash values are integers.
They are used to quickly compare dictionary keys during a dictionary lookup.
Numeric values that compare equal have the same hash value even if they are of different types, as is the case
 for 1 and 1.0.
For objects with custom __hash__() methods, note that hash() truncates the return value based on the bit width
 of the host machine.
"""

# hash for integer unchanged
print('The hash for 1 is:', hash(1))

# hash for decimal
print('The hash for 1.0 is:',hash(1.0))
print('The hash for 3.14 is:',hash(3.14))

# hash for string
print('The hash for Python is:', hash('Python'))

# hash for a tuple of vowels
vowels = ('a', 'e', 'i', 'o', 'u')
print('The hash for a tuple of vowels is:', hash(vowels))

# hash for a custom object
class Person:
    def __init__(self, age, name):
        self.age = age
        self.name = name
    def __eq__(self, other):
        return self.age == other.age and self.name == other.name
    def __hash__(self):
        return hash((self.age, self.name))
person = Person(23, 'Adam')
print('The hash for an object of person is:', hash(person))
root@Nestor:~/iot/lesson10# python3 hash_value.py
The hash for 1 is: 1
The hash for 1.0 is: 1
The hash for 3.14 is: 3228180212899174 43
The hash for Python is: 2514187322219292545
```

```
The hash for a tuple of vowels is: 6668126310924024688
The hash for an object of person is: 7738301349601281989
root@Nestor:~/iot/lesson10# python3 hash_value.py
The hash for 1 is: 1
The hash for 1.0 is: 1
The hash for 3.14 is: 3228180212899174 43
The hash for Python is: 59639737759159441 60
The hash for a tuple of vowels is: -2861034602432199367
The hash for an object of person is: -38003946151123767 52
root@Nestor:~/iot/lesson10#
```