



```
38         self.file.seek(0)
39         self.fingerprints.update(job_dir)
40
41     @classmethod
42     def from_settings(cls, settings):
43         debug = settings.getbool("DEBUG", False)
44         return cls(job_dir(settings), debug)
45
46     def request_seen(self, request):
47         fp = self.request_fingerprint(request)
48         if fp in self.fingerprints:
49             self.request_fingerprint.add(fp)
```

Python

▼ print string

```
Print ("hello,world ")
Print ('\n') #new line
```

▼ math

```
print (50 + 50)
print (50 - 50)
print (50 * 50)
print (50 / 50 )
print (50 * 50 / 50 + 50 - 50) #pemdas
print (50 ** 50) #power
print (50 % 6) #module
print (50 / 6 ) #left overs
print ( 50 // 6 ) #no leftovers
```

▼ variables and methods

```
x = " all is fair in love and war"
name ="M8SJT8" ***#string***

age =19 ***#int int(30)**

gpa = 3.1 ***#float float(3.1)**

print ( x.upper() ) ***#upper case all strings***

print ( x.lower() ) ***#lower case***

print ( x.title() ) ***#title case***

print (len(x) ) ***#number of strings***

print ( "My name is " + **name** + "and i am " + str(**age**) + "years old ")

age +=1 ***#add to variable**
```

```
print (age)
```

▼ condition statement

```
def drink (money ) :  
if money >=5 :  
    return "ok"  
else :  
    return "no"  
print (drink (5))  
print (drink (4))
```

▼ Boolean expressions

```
print ( "boolean expresions" )  
bool1 = True  
  
bool2 = 3*3 == 9  
  
bool3 = 3*3 != 9  
  
bool4 = False  
  
print (bool1,bool2, bool3,bool4 )  
  
**#relational and boolean opertors**  
  
test1 = (7>5) and (7<5)  
  
test2 = (7>5) or (7<5)  
  
test3 = not True  
  
print (test1 + test2 + test3 )
```

▼ functions

```
name ="M8SJT8" #string  
age = 19 **#int int(19)**  
  
print ("here is an example function : " )  
  
**def who ():**  
  
name ="M8SJT8"  
  
age = 19  
  
print ( name + '\n' + str(age) )  
  
**who()**  
  
#add function  
  
**def add_number (num):**  
  
num+=100  
  
print (num)  
  
**add_number (100)**
```

```

**def add (x,y):**
print (x,y)
add(7,7)

**def multiply  (x,y):**
return x*y

print (multiply(7,7))

#route

**def sqr_route (x) :**
print  (x** .5)

sqr_route(64)

```

▼ LOOPING

```

**#for loop - start to finish of an iterate**

name = [ "mosaad" , "sallam" , "zoom" , "Touch"]

for x in name :

print (x)

**#while loops - execute as long as true**

i = 1

while i < 10 :

print (i)

i+=1

```

▼ LISTS

```

name = ["mosaad", "sallam", "zoom", "touch"]

print (name[1]) #print the second item
print (name[0]) #print the first item
print (name[1 :3])
print (name [:2])
print (len (name ))
name.append("M8S2T8") #add element
print (name )
name.pop() #remove the last element
print (name)

```

▼ Dictionaries -key/value pairs {}

```

drinks = {"white russuan" :7,"old fashion":10,"lemon drop" :8 } #drinks is key , price is value
print (drinks)
employees = {"finance" : ["bob" , "linda", "tina" ], "IT" : [ "gene","loise", "teddy"], "HR": [ "jemmy","JR","mort"] }
print (employees)
employees ['legal']=["mr.frond"] #add new key :value
print (employees)

```

```
employees.update({"sales":["andie",10]} #add key:value
print (employees)
```

▼ Delete Variable

You can use the `del` statement to remove a variable.

▼ Sockets Script

```
import socket

HOST = '127.0.0.1'
PORT = 7777

S=socket.socket(socket.AF_INET,socket.SOCK_STREAM)

s.connect ((HOST,PORT))
```

▼ Port Scanner Script

```
#!/bin/python
import sys
import socket
from datetime import datetime
#Define our target
if len(sys.argv) == 2:
    target = socket.gethostbyname(sys.argv[1]) #translate hostname to IPv4
else :
    print ("Invalid amount of arguments .")
    print("Syntax: python3 scanner.py <ip>")
#add a pretty banner
print ("_" *50)
print ("M8SZT8 Script")
print ("scanning target " + target )
print ("time started : " +str(datetime.now()))
print ("_"*50)

try:
    print ("Enter the port scanning range :")
    print ("the small port first please :)")
    x = int(input())
    y = int(input())
    for port in range(x , y):
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        socket.setdefaulttimeout(1)
        result = s.connect_ex((target,port))
        print ("Chicking port{}".format(port))
        if result == 0:
            print ("port {} is open".format(port))
            s.close()
except KeyboardInterrupt:
    print ("\nExiting program.")
    sys.exit()

except socket.gaierror:
    print ("Hostname Could not be resolved .")
    sys.exit()
except socket.error:
    print ("Couldn't connect to server.")
    sys.exit()
```

