**import** random

**print** random.randint(0, 5)

------------------------------------

**import** random

myList = [2, 109, **False**, 10, "Lorem", 482, "Ipsum"]

random.choice(mylist)

------------------------------------------

**from** random **import** shuffle

x = [[i] **for** i **in** range(10)]

shuffle(x)

--------------------------------------

**import** random

**for** i **in** range(3):

**print** random.randrange(0, 101, 5) – tot 100 per stappen van 5

**import** time

**import** datetime

**print** "Time in seconds since the epoch: %s" %time.time()

**print** "Current date and time: " , datetime.datetime.now()

**print** "Or like this: " ,datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

**print** "Current year: ", datetime.date.today().strftime("%Y")

**print** "Month of year: ", datetime.date.today().strftime("%B")

**print** "Week number of the year: ", datetime.date.today().strftime("%W")

**print** "Weekday of the week: ", datetime.date.today().strftime("%w")

**print** "Day of year: ", datetime.date.today().strftime("%j")

**print** "Day of the month : ", datetime.date.today().strftime("%d")

**print** "Day of week: ", datetime.date.today().strftime("%A")

**getting certain weekday**

**import** datetime

mydate = datetime.date(1943,3, 13) #year, month, day

**print**(mydate.strftime("%A"))

range([start], stop[, step])

* start: Starting number of the sequence.
* stop: Generate numbers up to, but not including this number.
* step: Difference between each number in the sequence

>>> for i in range(4, 10, 2):

...     print(i)

...

4

6

8

---------------------------------------------------------------------------------------------------------------------------------------

\n = linefeed (prints the stuff after this on the next line)  
\r = carriage return (basically also used for printing stuff on the next line)  
\’ = print a single quote ( ‘ ) in your text  
\” = print a double quote ( “ ) in your text  
\\ = print a backslash ( \ ) in your tekst

\t = print a TAB  
\s = blank space

The default value of end is \n meaning that after the print statement it will print a new line. So simply stated end is what you want to be printed after the print statement has been executed

Eg: - print ("hello",end=" +") will print hello +

--------------------------------------------------------------------------------------------------------------------------------------

# Function definition is here

def printme( str ):

"This prints a passed string into this function"

print str

return;

# Now you can call printme function

printme("I'm first call to user defined function!")

printme("Again second call to the same function")

append() Adds an element at the end of the list

clear() Removes all the elements from the list

copy() Returns a copy of the list

count(lijstnaam)Returns the number of elements with the specified value

extend( ) Add the elements of a list (or any iterable), to the end of the current list

index() Returns the index of the first element with the specified value

insert() Adds an element at the specified position

pop() Removes the last element in the list

lijst.remove(“dog”) Removes the item with the specified value

reverse() Reverses the order of the list

sort() Sorts the list

all() Return True if all elements of the list are true (or if the list is empty).

any() Return True if any element of the list is true. If the list is empty, return False.

enumerate() Return an enumerate object. It contains the index and value of all the items of list as a tuple.

len() Return the length (the number of items) in the list.

list() Convert an iterable (tuple, string, set, dictionary) to a list.

max() Return the largest item in the list.

min() Return the smallest item in the list

sorted() Return a new sorted list (does not sort the list itself).

sum() Return the sum of all elements in the list.

Ik ben {naam}.format(naam = vippy) extra kan {: en dan iets uit de lijst)

{:03} maakt van 5 – 005

Iet specifiek uit een lijst =

Lijst =[]  
print = nummer {0[plaats 8 fzo]}.format(lijst) neemt plaats 1 wa in haakjes staat en dan plaats 8 in die lijst

{:>5}\t{:>12}\t{:>8} voor ta

< Left aligned to the remaining space

^ Center aligned to the remaining space

> Right aligned to the remaining space

= Forces the signed (+) (-) to the leftmost position

Checken of een getal oneven is =   
if getal % 2 != 0:

Kut modulo

fun = "party"

print(fun[1:4]) art

print(fun[1:4:2]) at

print(fun[:2]) pa

print(fun[3:]) ty

print(fun[0::2]) pry

print(fun[-1::-1]) ytrap

strings zijn onveranderbaar dus fruit[2] = r zal niet werken

naamvar.strip(character dat wegmoet) = 5555hello555 wordt met strip(5) gwn hello

.upper  
.lower  
.find = index van specifiek deel vinden (pakt auto enkel eerste index, voor 2de zelfde woord  
first = naam.find(“bye”)   
naam.find(“bye”, first + 1))

str.replace(old, new[, max])

asccii

ord(teken of letter) = van a naar 97 int – ascii  
chr(“”)= van 97 naar a

 voor strings achterstevoren te printen





