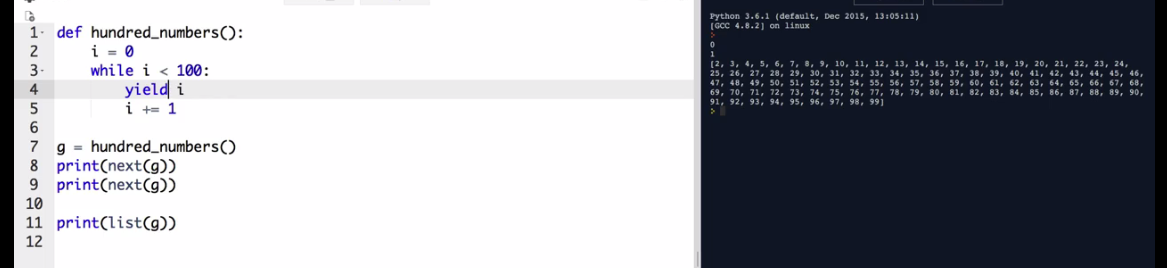
Generators: Remembers that state of function where we left



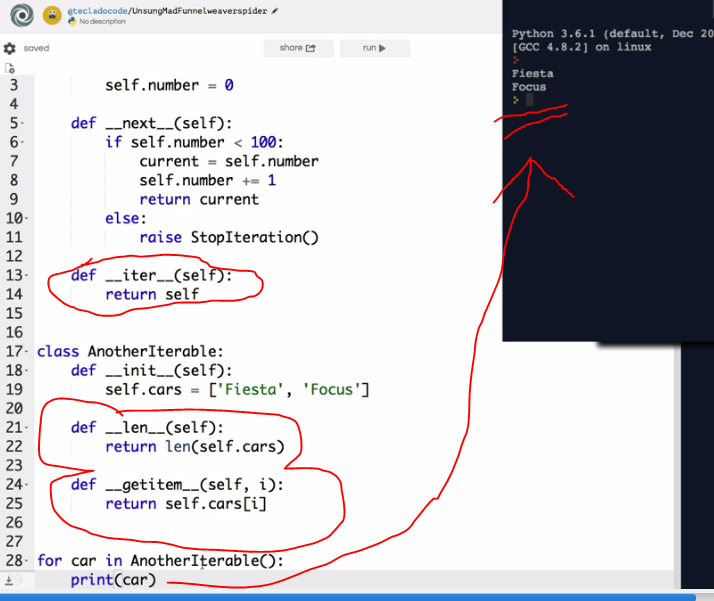
Here when hundred function called it generates the value from 0 to 100

When it called as “next(g)” – it only printed 0 and remembered where it stopped

Again, when it has called as “next(g)” - it only printed 1 and remembered where it stopped

Then when “list(g)” called – it printed right after where it left (from 2)

To make a class iterable it has to have these dunder function (red circled)



Iterator: Used to go to the next value

Iterable: Used to go over every value

Generator comprehension:

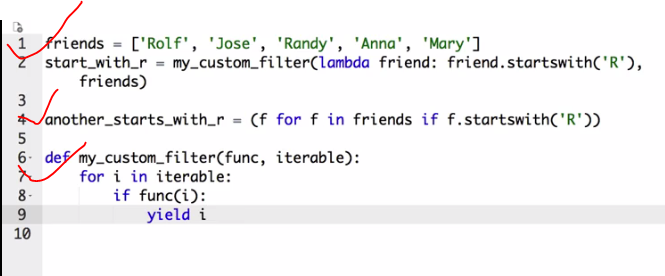
My\_numbers\_gen = (x for x in [1, 2, 3, 4, 5])

Print (next (My\_numbers\_gen)) – 1

Use of startwith function, basically used to find value in a list or dictionary



These 3 are identical method, you can use any one of them but list comprehension is good and more pythonic.



Use of map function, this function is also use for finding value in list



Any (list or dict) – returns true if there is at least one value or false if none

All(list or dict or tuple) – returns true if all the value is true or false if one of them is false

