	A probable of the
650	Map functions
	The Alliana and the second of
	It allows you to "map" a function to an
	It allows you to "map" a function to an Iterable object. i.e. you can call same function
	to every Hem on an Heralele.
	Some of the same o
gn;	ded square (num):
	def square (num): seturn num 4+2
q_{n}	mynems = [9,2,3,4]
an	map Csquare, mynumi
No	map at 0x205
77	- danbeda karones den
go	1
	To get nexulte . Sterate through
11.19	To get results, Sterate throughon man or call to a lest
	The house describer as the shall
qn:	lest (map (square, nynums))
	halo college braining
0/6:	[1,4,9,16]
-/	
-	The off the British of the bold of the state
	1.11
3.	Filter function
1,	
of the	9+ returns on sterator relding those stems
	9t returns an sterator yselding those stems of sterable for which function (stem) is true. s.e. you need to filter ley a function that yeturns either true or false.
	s.e. you need to filter ley a feine" that
	neturns ether tour or false.
	III. The state of

Then hasseng that ento felter calong with

your sterable) & you'll get back only the success

that would setturn true when passed to the

function

9n: def check even (new):

seturn num'-7 ==0

2n: num = [0,1,2,3,4,5,6] 9n: 18st (filter (check even, nums) 0/p: [0,2,4,6] (many) Lambda Expression They allow up to create anonymous funcon. They work same as those created & cassigned using def Note Lambda's body is sentar to what we put doitant total Note dambda es a sengle expression not a block of statements

gn:	def square (num): outurn num ##2
gn;	square (2)
gp:	4 stamataile total
gn:	Jambdaen: n++2
0/p:	(function_main < lamas)
	# you wouldn't assegn usually assegn a name to a lamba expression, this is for demonstration
	The Mark have access with the second
gn:	square (2)
o/p:	
	#Passing to MAP & FILTER
gn:	nynums = [1,2,3,4,5]
	(O2, ≈ 50) (t)
	Ust (map (lambda n:n 4 + 2, mynums)
0/p;	[1,4,9,16,25]
gn:	lest (filter (lambda n: n/2==0; onynums)
0/9	[2,4]
	You can also pass multiple arg.

for eg?

dambda n,y: n ty