Petr van Blokland Python for Designers So ideas to stat. Using BageBot Introduction his is course is about Python. If you now think that it's about snakes and not about programming, you don't want to continue, but if you are here with the expectation that you will learn about programming techniques and objects and classes didicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by profession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. In that we really start from scratch, using daily life examples to visualize the programs and their usage. That is a different approach for a problem. In the two really start from scratch, using daily life examples to visualize the programs and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. In this course, But I'll try, by ultimate best to claim in this course, and their usage. That is a different approach from many other programs courses, which often start with a technical solution in search for a problem. Has header here? There will be a lot of coding in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in this course, But I'll try, by ultimate best to claim in the course with a toth one be a very long footone to run over multiple lines one programmers want you to be believe.		
Petr an Blokland Python for Designers 50 ideas to stat. Using Bageot Introduction is secured is about Python. If you now think that it's about snakes and not about prosramming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes a dicated for the design practice, then you are not the right track. By the way, you don't have the bea designed by profession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which offen start with a technical solution in search for a problem. Is a different approach in order to feel with the scanding daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is a different approach in order to feel with the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach in orm many other programming courses, which often start with a technical solution in search for a problem. Is header here. There will be a lot of coding in this course. But I'll try, by ultimate best to clarify as much as I can and to relate every hing to proclical problems, that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.	0 1	
Petr an Blokland Python for Designers So ideas to stalt. Using Bage Bot. Introduction his county is about Python. If you now think that it's abolt snakes and not about programming you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by profession, in order to follow this course. It's characteristic is that we really start from scrutch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is, a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scrutch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scrutch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scrutch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is header hereThere will be a lot of coding in this course. But I'll try by ultimate best to clarify as much as I can and to relate every—thing to practical problems that you can recognize and visualize I am pretty sure that you will see that programming is not as mugic as some programmers want you to believe.	0	0
Python for Designers So ideas to stat. Using BageBot.IntroductionThis is course is about Python. If you now hink that it's about shout sprace is about Python. If you now hink that it's about snakes and not about programming, you don't want to continue. But if you are here with the expectation that you will learn about profesamming techniques and objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have the a designer by profession, in order to foll by this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every thing to practical problems that you can recognize and visualize. I am prettly sure that you will see that programming is not as magic as one to run over multiple lines as some programmers want you to believe.		1
Designers So ideas to start. Using BageBot Introduction This is course is about Python. If you now think that it's about snakes and not about programming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes obdicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by profession, in order to follow this course it's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from the programs of the start with a technical solution in search for a problem. Is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Ha header hereThere will be a lot of coding in this course. But I'll try by ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. Lampretty sure that you will see that programmin; is not as magic note to run over multiple lines as some programmers want you to believe.		2
Designers So ideas to start. Using BageBot Introduction This is course is about Python. If you now think that it's about snakes and not about programming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes obdicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by profession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try by ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. Lampretty sure that you will see that programming is not as magic as some programmers want you to believe.		3
Designers So ideas to start. Using BageBot Introduction This is course is about Python. If you now think that it's about snakes and not about programming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes objected for the design practice, then you are on the right track. By the way, you don't have to be a designer by procession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs of the start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Ha header here There will be a lot of coding in this course. But I'll try by ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. Lampretty sure that you will see that programming is not as magic so some programmers want you to believe.		4
So ideas to start. Using BageBot Introduction this course is about Python. If you now think that it's about snakes and not about proparamming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes didicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by procession, in order to follow this course. It's characteristic is that we really start from scratch, using doily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the really start from start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a cechnical solution in search for a problem. If a header here There will be a lot of coding in this course. But I'll try try ultimate best to clarify as much as I can and to relate every hing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		5
So ideas to start. Using BageBot Introduction this course is about Python. If you now think that it's about snakes and not about proparamming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes didicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by procession, in order to follow this course. It's characteristic is that we really start from scratch, using doily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the really start from start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a cechnical solution in search for a problem. If a header here There will be a lot of coding in this course. But I'll try try ultimate best to clarify as much as I can and to relate every hing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.	1	6
So ideas to start. Using BageBot Introduction This is course is about Python. If you now think that it's about snakes and not about proparamining, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes didicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by profession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structures, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try by ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		7
ion that it's about stakes and not about programming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes of dicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by programming techniques and objects and classes of dicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by programming to a session in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programs. Their structure, their behavior are problem. In this is the text of the footnote. It is a different approach from many other programming courses, which often start with a technical solution in search for a problem. It is a different approach from many other programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a cechnical solution in search for a problem. It is a different approach from many other programming courses. But I'll try by ultimate best to clarify as much as I can and to relate every hing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programmins is not as magic note to run over multiple lines as some programmers want you to believe.		8
think that it's about snakes and not about programming, you don't want to continue. But if you are here with tile expectation that you will learn about programming techniques and objects and classes dedicated for the design practice, then you are hor the right track. By the way, you don't have to be a designer by pro- fession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. 15		9
gramming, you don't want to continue. But if you are here with the expectation that you will learn about programming techniques and objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by pro- fession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try fly ultimate best to clarify as much as I can and to relate every- thing to practical problems hat you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		10
you are here with tile expectation that you will learn about programming techniques and objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by pro- fession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach for a problem. ### Structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. ### Haeder here* There will be a lot of coding in this course. But I'll try thy ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		11
will learn about programming techniques and objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by program, or order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems shat you can recognize and visualize. I ampretty sure that you will see that programmins is not as magic you will see that programmins is not as magic as some programmers want you to believe.	2	12
objects and classes dedicated for the design practice, then you are on the right track. By the way, you don't have to be a designer by pro- fession, in order to follow this course. It's characteristic—is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programmins is not as magic as some programmers want you to believe.		13
practice, then you are on the right track. By the way, you don't have to be a designer by pro- fession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		14
way, you don't have to be a designer by profession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach irom many other programs. Their structure, their behavior and their usage. That is a different approach irom many other programming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		15
fession, in order to follow this course. It's characteristic is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. Is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach rom many other programming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I ampretty sure that you will see that programmens is not as magic as some programmers want you to believe.		16
characteristic 1s that We really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		17
scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try rpy ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programmink is not as magic as some programmers want you to believe.	3	18
the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach is the program of their usage. The structure is the programs of the programs of the programs of the programs of the program of their usage. That is a different approach is a different approach in the scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach in the scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach in the scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach in the scarch of the programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		19
and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		20
often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header hereThere will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		21
often start with a technical solution in search for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		22
for a problem. is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.	4	23
is that we really start from scratch, using daily life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other pro- gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try ray ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.	4	24
life examples to visualize the programs. Their structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		25
structure, their behavior and their usage. That is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		26
is a different approach from many other programming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		28
gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that Which can be a very long foot- note to run over multiple lines gramming courses, which often start with a technical solution in search for a problem. H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.		29
## technical solution in search for a problem. ### H3 header here ### There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate every—thing to practical problems that you can recognize and visualize. I ampretty sure that you will see that programming is not as magic as some programmers want you to believe.	5	30
H3 header here There will be a lot of coding in this course. But I'll try my ultimate best to clarify as much as I can and to relate everything to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.	3	31
in this course. But I'll try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I ampretty sure that Which can be a very long foot- note to run over multiple lines as some programmers want you to believe.		32
clarify as much as I can and to relate every- thing to practical problems that you can This is the text of the footnote. Which can be a very long foot- note to run over multiple lines In this course. But I if try my ultimate best to clarify as much as I can and to relate every- thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		33
thing to practical problems that you can thing to practical problems that you can thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		34
1 This is the text of the footnote. Which can be a very long footnote to run over multiple lines Thing to practical problems that you can recognize and visualize. I am pretty sure that you will see that programming is not as magic as some programmers want you to believe.		35
Which can be a very long foot- you will see that programming is not as magic as some programmers want you to believe.	6.1 This is the tart of the fact	36
note to run over multiple lines as some programmers want you to believe.	1 THIS IS THE TEXT OF THE TOOL	37
as some programmers want you to believe.		38
with indented left margin	with indented left margin	39
And what is more important, knowing about		40
now programming works yourself can actual-		41
iy save you a lot of time. Even if you don't		42
want to be a programmer. The coursets set up		43
as a Stownis off which book as the provide		44
		45
process in itself - increased knowledge and	with machica for margin	46
		47
48 8	8	48
49 #??#		49
50		50

	0 1	2 3 4 5	6
0 (understanding about how it should be done -	0
1		there will be continuous improvement on the	1
2		code and the examples. Feedback from sub-	2
3		scribers and the regular updates of Python	3
4			4
5		make that the course will adapt and grow over	5
	1	time. So the subscription fee of the course will	6
7		grow too.	0
/		of a course like this is a design process in itself	/
8		- increased knowledge and understanding	8
9		about how it should be done - there will be	9
10		continuous improvement on the code and the	10
11		examples. Feedback from subscribers and the	11
12 2	2	regular update <mark>s</mark> of P ython mak <mark>e</mark> that the	12
13		course will adapt and grow over time. So the	13
14		subscription fee of the course will grow too.	14
15		This makes the plan for course into an alter-	15
16		native constru <mark>ct</mark> ion of a kickst <mark>ar</mark> t project. If	16
17		you are an early adapter, trusting that the	17
18	3	course will grow and develop in a direction	18
19		that you need, then you just pay the current	19
20		amount. After that every addition is available	20
21		free of charge. The Udemy courses always	21
22		have a lifetime subscription for the fee that	22
23		you initially paid for it. If you wait for a few	23
24	1	months, more content will be added and the	24
25		price will be subsequently higher, adding ap-	25
26		proximately \$16 per hour video.	26
27		Another H3 header here Any time you jump	27
28		on the bandwagon, you will pay the price as it	28
29		is at that moment, based on the volume of the	29
30		content at that moment Relatively low in the	30
31		beginning, putting your trust in the expecta-	31
32		tion we'll develop the course further. We	32
33		start with 2 hours of instructions and exam-	33
34		ples. If you wait for a while, you will pay more	34
35		for the same content. So, if you are a designer,	35
36	O mbi-i-th-t	or you have other reasons to use Python in	36
37	3 This is the text of the footnote.		37
38	Which can be a very long foot-	your professional life or your personal life, you	38
39	note to run over multiple lines	are already using Python or you expect to do	39
40	with indented left margin.	that in the future, then joining this growing	40
		environment is likely to be profitable for you.	
41	7	There are many good examples around show-	41
42		ing the great potential of programming in	42
43		Python, but most are solutions in search for a	43
44		problem to be solved. Using programming in	44
45		your daily practice requires a reversed ap-	45
46		proach. You want to achieve something and	46
47			47
	3		48
49			#??# 49
50			50

0 0	0 1	1	2	3	4	5	6	
0 0				ning and what is t	_			0
<u></u>				done. Instead of				<u></u>
2			translation of "Do you know where the station					2
3			is?" in a tourist guide, you are interested in					3
4				n this foreign lan				4
5				on the topic. This				5
6 1				t. And since these	-			6
7				hanging overtime				7
8				at will adapt and				8
9				fixed "how to" co				9
10				se an overview o	f possible fu-			10
11			ture topics is giv					11
12 2				ne course further.				12
13				uctions and exam				13
14				e, you will pay mo				14
15				So, if you are a de				15
16				ons to use Pythor			_	16
17			fessional life or	your personal lif	e, you are al-			17
18 3				hon or you expec				18
19				joining this grow				19
20				be profitable for				20
21				examples around				21
22				of programming i				22
23				ons in search for				23
24 4				g programming ir				24
25				es a reversed app				25
26				e something and				26
27				s can be done. In				27
28				ion of "Lo you kn				28
29				tourist guide, you				29
30 5				ions in this foreig				30
31				decide on the top				31
32				to do that. And si				32
33				divers and chang				33
34				vironment that w				34
35				f presenting a fixe				35
36 6				nd of the course				36
37				re topics is given				37
38				maintained over				38
39				ds expressed by				39
40				he development				40
41				stay in sync with				41
42 7				rld. To what exten				42
43				rid. To what extend ture promise, but				43
44				ou express the tr				44
45								45
46				a reward for this nt for the current				46
47					t biroginan.			46
48 8								48
48 8				+		+	#??#	48
49 50						+		

	0 1	2 3 4	5 6	
0 0		all future content for the current price. Sub-		0
1		head here This course is the twin of Process-		1
2				1
2		ing for Designers course. Much of text is the		
5		same, as the structure of the Processing and		3
4		Python is very similar. Also the code examples		4
5		are very much alike, except that they are		5
6 1		adapted to the syntax of each language. And in		6
7		the advanced part of the courses the examples		7
8		start to drift apart, because the available func-		8
9		tions and libraries is different. You can decide		9
10		to go through both courses if you want to		10
11		learn the differences. But if you already have a		11
12 2		preference or you made a choice, then follow-		12
13		ing only one of the two courses may be suffi-		13
14		cient as a start. If you are starting fresh on pro-		14
15		gramming, the choice can be based on the ex-		15
16		pertise that is available in your environment,		16
17		that is a very practical reason. You choice also		17
18 3		be based on the difference in flavor between		18
19				19
20		the languages. In preparation of deepening in		20
21		each of there languages here is a brief sum-		21
		mary about their characteristics. Processing is		
22		based on Java, <mark>an indus</mark> trial str <mark>e</mark> ngth program-		22
23		ming language, where the type of objects		23
24 4		needs to be sp <mark>ec</mark> ified at the start of a program.		24
25		Python has a much more free usage of types,		25
26		which makes it good for "sketchy" program-		26
27		ming, but it is l <mark>es</mark> s relia <mark>b</mark> le in ci <mark>rc</mark> umstances		27
28		where the prediction of flawless execution is		28
29		important. But in reverse, this makes Python		29
30 5		much more flexible in the storage of informa-		
31		tion. Especially the mixing of data type and		31
32		the storage in the standard dictionary type, al-		32
33		low Python to build data structures that are		33
34		very hard to achieve in Processing. Subhead		34
35		here The origin of Processing is more in the		35
36 6		processing of images, - focussed on pixels and		36
37		interaction – than Python. Python can for in-		37
38		stance be found inside web servers and as		38
39				39
40		scripting language in desktop applications		40
41		such as FontLab and RoboFont. In general Pro-		41
42 7		cessing programs are more linear, smaller and		
		dedicated to a specific task, where Python		42
43		programs tend to be part of larger systems. In		43
44		that respect Python should be more compared		44
45		on the level of Java, the language that Process-		45
46		ing is built on top of. Another difference is the		46
47				47
48 8				48
49			#??#	49
50				50

	0 1	2	2	4	5	6	
0 0		re seesing is by	ilt on ton of And		3	0	0
1		Processing is but ence is the amou					1
2		oraries of code i					2
3		There a some m					3
4		of the two langu					4
5		people they are					5
6 1		tomed to one kir					6
7		of curly brackets					7
8		of blocks of code					8
9		the way Python					9
10		olock: entirely b					10
11		code line. In this	course the diffe	erences be-			11
12 2		tween Processin					12
13		tioned if that is r	1				13
14		course will mair		use of Pro-			14
15		cessing in the de	esign practice.				15
16							16
17							17
18 3 19							18
20							19
21							20
22							
23							23
24 4							24
25							25
26							26
27							27
28							28
29							29
30 5							30
31							31
32							32
33							33
34							34
35							35
36 6			<u> </u>				36
37			<u> </u>				37
38			$\overline{}$				38 39
40							<u>40</u>
41							41
42 7							42
43							43
44							44
45							45
46							46
47				4			47
48 8							48
49						#??#	49
50							50