

	0	1	2	3	4	5	6	
0	0		ThesisBot Example					0
1								1
2			Petr van Blokland					2
3								3
4			Python for					4
5			Designers					5
6	1							6
7								7
8								8
9			50 ideas to start. Using BageBot.	Introduction				9
10			This is course is about Python. If you now					10
11			think that it's about snakes and not about pro-					11
12	2		gramming, you don't want to continue. But if					12
13			you are here with the expectation that you					13
14			will learn about programming techniques and					14
15			objects and classes dedicated for the design					15
16			practice, then you are on the right track. By the					16
17			way, you don't have to be a designer by pro-					17
18	3		fession, in order to follow this course. It's	1				18
19			characteristic is that we really start from					19
20			scratch, using daily life examples to visualize					20
21			the programs. Their structure, their behavior					21
22			and their usage. That is a different approach					22
23			from many other programming courses, which					23
24	4		often start with a technical solution in search					24
25			for a problem.					25
26			is that we really start from scratch, using daily					26
27			life examples to visualize the programs. Their					27
28			structure, their behavior and their usage. That					28
29			is a different approach from many other pro-					29
30	5		gramming courses, which often start with a					30
31			technical solution in search for a problem.					31
32								32
33			H3 header here					33
34			There will be a lot of coding					34
35			in this course. But I'll try my ultimate best to					35
36	6	1	This is the text of the footnote.					36
37			Which can be a very long foot-					37
38			note to run over multiple lines					38
39			with indented left margin.					39
40		2	This is another footnote on the					40
41			same page.					41
42	7	3	This is the text of the footnote.					42
43			Which can be a very long foot-					43
44			note to run over multiple lines	2				44
45			with indented left margin.					45
46								46
47								47
48	8							48
49							#??#	49
50								50

	0	1	2	3	4	5	6	
0	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			make that the course will adapt and grow over					4
5			time. So the subscription fee of the course will					5
6	1		grow too.					6
7			of a course like this is a design process in itself					7
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		regular updates of Python make 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 construction of a kickstart 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 Udemmy 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	4		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	5		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	6	3 This is the text of the footnote.	or you have other reasons to use Python in					36
37		Which can be a very long foot-	your professional life or your personal life, you					37
38		note to run over multiple lines	are already using Python or you expect to do					38
39		with indented left margin.	that in the future, then joining this growing					39
40			environment is likely to be profitable for you.					40
41			There are many good examples around show-					41
42	7		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
48	8							48
49							#??#	49
50								50

	0	1	2	3	4	5	6	
0	0		achieve something and what is the best pat-					0
1			tern this can be done. Instead of reading the					1
2			translation of "Do you know where the station					2
3			is?" in a tourist guide, you are interested in					3
4			conversations in this foreign language where					4
5			you can decide on the topic. This course is					5
6	1		trying to do that. And since these patterns are					6
7			so divers and changing overtime, you need an					7
8			environment that will adapt and grow, instead					8
9			of presenting a fixed "how to" course. At the					9
10			end of the course an overview of possible fu-					10
11			ture topics is given.					11
12	2		we'll develop the course further. We start with					12
13			2 hours of instructions and examples. If you					13
14			wait for a while, you will pay more for the					14
15			same content. So, if you are a designer, or you					15
16			have other reasons to use Python in your pro-					16
17			fessional life or your personal life, you are al-					17
18	3		ready using Python or you expect to do that in					18
19			the future, then joining this growing environ-					19
20			ment is likely to be profitable for you. There					20
21			are many good examples around showing the					21
22			great potential of programming in Python, but					22
23			most are solutions in search for a problem to					23
24	4		be solved. Using programming in your daily					24
25			practice requires a reversed approach. You					25
26			want to achieve something and what is the					26
27			best pattern this can be done. Instead of read-					27
28			ing the translation of "Do you know where the					28
29			station is?" in a tourist guide, you are interest-					29
30	5		ed in conversations in this foreign language					30
31			where you can decide on the topic. This					31
32			course is trying to do that. And since these					32
33			patterns are so divers and changing overtime,					33
34			you need an environment that will adapt and					34
35			grow, instead of presenting a fixed "how to"					35
36	6		course. At the end of the course an overview					36
37			of possible future topics is given.					37
38			This list will be maintained over time, adding					38
39			wishes and needs expressed by you, the user					39
40			of the course. The development of the exam-					40
41			ples will try to stay in sync with changes in					41
42	7		the outside world. To what extent this will					42
43			succeed is a future promise, but by joining in					43
44			at early stage, you express the trust that this					44
45			will happen. As a reward for this trust you get					45
46			all future content for the current price.					46
47			Sub-					47
48	8							48
49							#??#	49
50								50

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

	0	1	2	3	4	5	6	
0	0		<p>Processing is built on top of. Another difference is the amount and type of available libraries of code is another important factor. There are some minor differences in the syntax of the two languages - minor, but for some people they are really annoying, being accustomed to one kind of notation, such as the use of curly brackets to indicate the start and end of blocks of code in Processing (and Java) and the way Python detects the start and end of a block: entirely by the amount indent of a set of code line. In this course the differences between Processing and Python will be mentioned if that is really important, but this course will mainly focus on the use of Processing in the design practice.</p>					0
1								1
2								2
3								3
4								4
5								5
6	1							6
7								7
8								8
9								9
10								10
11								11
12	2							12
13								13
14								14
15								15
16								16
17								17
18	3							18
19								19
20								20
21								21
22								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							36
37								37
38								38
39								39
40								40
41								41
42	7							42
43								43
44								44
45								45
46								46
47								47
48	8							48
49							###	49
50								50