

	0	1	2	3	4	5	#??#	
0	0							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

ThesisBot Example

AAAAAAA Python
AAAAAAA

Python for Design- ers

AAAAAAA

50 ideas to start. Using BageBot.

AAAAAAA

Introduction

AAAAAAA 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 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**

AAAAAAA 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

I 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.

AAAAAAA [myAu-
thor2016]AAAAAAA

AAAAAAA

AAAAAAA

H3 header here

AAAAAAA There will be a lot of coding in this course. But I'll try my ultimate best to clarify

	0	1	2	3	4	5	#??#	
0	0		course. But I'll try my ultimate best to clarify					0
1			as much as I can and to relate everything to					1
2			practical problems that you can recognize and					2
3			visualize. I am pretty sure that you will see					3
4			that programming is not as magic as some					4
5			programmers want you to believe. And what is					5
6	1		more important, knowing about how pro-					6
7			gramming works yourself, can actually save					7
8			you a lot of time. Even if you don't want to be					8
9			a pro-programmer. The course is set up as a					9
10			growing environment. Because the					10
11			development					11
12	2		2 of a course like this is a design process in itself - increased knowledge					12
13			and understand-ing about					13
14			how it should be done -					14
15			there will be contin-uous					15
16			improvement on the code					16
17			and the examples. Feedback					17
18	3		from subscribers and the					18
19			regular updates of Python					19
20			make that the course will					20
21			adapt and grow over time.					21
22			So the subscrip-tion fee of					22
23			the course will grow too.					23
24	4							24
25			AAAAAAA					25
26			AAAAAAAThis makes the plan for course into					26
27			an alternative construction of a kickstart					27
28			project. If you are an early adapter, trusting					28
29			that the course will grow and develop in a di-					29
30	5		rection that you need, then you just pay the					30
31			current amount. After that every addition is					31
32			available free of charge. The Udemy courses					32
33			always have a lifetime subscription for the fee					33
34			that you initially paid for it. If you wait for a					34
35			few months, more content will be added and					35
36	6		the price will be subsequently higher, adding					36
37			approximately \$16 per hour video.					37
38			AAAAAAA					38
39			Another H3 header here					39
40			AAAAAAA Any time you jump on the bandwag-					40
41			on, you will pay the price as it is at that mo-					41
42	7		ment, based on the volume of the content at					42
43			that moment. Relatively low in the beginning,					43
44			putting your trust in the expectation					44
45			we'll develop the course					45
46			further. We start with 2					46
47								47
48	8							48
49								49
50								50

2 of a course like this is a design process in itself - increased knowledge and understand-ing about how it should be done - there will be contin-uous improvement on the code and the examples. Feedback from subscribers and the regular updates of Python make that the course will adapt and grow over time. So the subscrip-tion fee of the course will grow too.

AAAAAAA
AAAAAAAThis makes the plan for course into an alternative construction of a kickstart project. If you are an early adapter, trusting that the course will grow and develop in a di-
rection that you need, then you just pay the current amount. After that every addition is available free of charge. The Udemy courses always have a lifetime subscription for the fee that you initially paid for it. If you wait for a few months, more content will be added and the price will be subsequently higher, adding approximately \$16 per hour video.
AAAAAAA
Another H3 header here
AAAAAAA Any time you jump on the bandwag-on, you will pay the price as it is at that mo-
ment, based on the volume of the content at that moment. Relatively low in the beginning,
putting your trust in the expectation
we'll develop the course further. We start with 2

	0	1		5	#??#	
0	0		with 2 hours of instructions			0
1			and examples. If you wait			1
2			for a while, you will pay			2
3			more for the same content.			3
4			So, if you are a designer, or			4
5			you have other reasons to			5
6	1		use Python in your profes-			6
7			sional life or your personal			7
8			life, you are already using			8
9			Python or you expect to do			9
10			that in the future, then join-			10
11			ing this growing environ-			11
12	2		ment is likely to be prof-			12
13			itable for you. There are			13
14			many good examples			14
15			around showing the great			15
16			potential of programming in			16
17			Python, but most are solu-			17
18	3		tions in search for a prob-			18
19			lem to be solved. Using pro-			19
20			gramming in your daily			20
21			practice requires a reversed			21
22			approach. You want to			22
23			achieve something and			23
24	4		what is the best pattern this			24
25			can be done. Instead of			25
26			reading the translation of			26
27			“Do you know where the			27
28			station is?” in a tourist			28
29			guide, you are interested in			29
30	5		conversations in this for-			30
31			oreign language where you			31
32			can decide on the topic.			32
33			This course is trying to do			33
34			that. And since these pat-			34
35			terns are so divers and			35
36	6		changing overtime, you			36
37			need an environment that			37
38			will adapt and grow, instead			38
39			of presenting a fixed “how			39
40			to” course. At the end of the			40
41			course an overview of pos-			41
42	7		sible future topics is given.			42
43			AAAAAAA			43
44			AAAAAAAThis list will be maintained over			44
45			time, adding wishes and needs expressed by			45
46			you, the user of the course. The development			46
47						47
48	8					48
49						49
50						50

with 2 hours of instructions and examples. If you wait for a while, you will pay more for the same content. So, if you are a designer, or you have other reasons to use Python in your professional life or your personal life, you are already using Python or you expect to do that in the future, then joining this growing environment is likely to be profitable for you. There are many good examples around showing the great potential of programming in Python, but most are solutions in search for a problem to be solved. Using programming in your daily practice requires a reversed approach. You want to achieve something and what is the best pattern this can be done. Instead of reading the translation of “Do you know where the station is?” in a tourist guide, you are interested in conversations in this foreign language where you can decide on the topic. This course is trying to do that. And since these patterns are so divers and changing overtime, you need an environment that will adapt and grow, instead of presenting a fixed “how to” course. At the end of the course an overview of possible future topics is given.

AAAAAAA
AAAAAAAThis list will be maintained over time, adding wishes and needs expressed by you, the user of the course. The development

	0	1	2	3	4	5	#??#	
0	0		of the examples will try to stay in sync with					0
1			changes in the outside world. To what extent					1
2			this will succeed is a future promise, but by					2
3			joining in at early stage, you express the trust					3
4			that this will happen. As a reward for this trust					4
5			you get all future content for the current price.					5
6	1		AAAAAAA					6
7			Subhead here					7
8			AAAAAAA This course is the twin of Processing					8
9			for Designers course. Much of text is the same,					9
10			as the structure of the Processing and Python					10
11			is very similar. Also the code examples are					11
12	2		very much alike, except that they are adapted					12
13			to the syntax of each language. And in the ad-					13
14			vanced part of the courses the examples start					14
15			to drift apart, because the available functions					15
16			and libraries is different. You can decide to go					16
17			through both courses if you want to learn the					17
18	3		differences. But if you already have a prefer-					18
19			ence or you made a choice, then following					19
20			only one of the two courses may be sufficient					20
21			as a start. If you are starting fresh on program-					21
22			ming, the choice can be based on the expertise					22
23			that is available in your environment, that is a					23
24	4		very practical reason. You choice also be					24
25			based on the difference in flavor between the					25
26			languages. In preparation of deepening in each					26
27			of there languages here is a brief summary					27
28			about their characteristics. Processing is based					28
29			on Java, an industrial strength programming					29
30	5		language, where the type of objects needs to					30
31			be specified at the start of a program. Python					31
32			has a much more free usage of types, which					32
33			makes it good for “sketchy” programming, but					33
34			it is less reliable in circumstances where the					34
35			prediction of flawless execution is important.					35
36	6		But in reverse, this makes Python much more					36
37			flexible in the storage of information. Espe-					37
38			cially the mixing of data type and the storage					38
39			in the standard dictionary type, allow Python					39
40			to build data structures that are very hard to					40
41			achieve in Processing.					41
42	7		AAAAAAA					42
43			Subhead here					43
44			AAAAAAA The origin of Processing is more in					44
45			the processing of images, - focussed on pixels					45
46								46
47								47
48	8							48
49								49
50								50

	0	1	2	3	4	5	#??#	
0	0		the processing of images, - focussed on pixels and interaction - than Python. Python can for instance be found inside web servers and as scripting language in desktop applications such as FontLab and RoboFont. In general Processing programs are more linear, smaller and dedicated to a specific task, where Python programs tend to be part of larger systems. In that respect Python should be more compared on the level of Java, the language that Processing is built on top of. Another difference is the amount and type of available libraries of code					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		is another important factor. There a 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.					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		AAAAA AAAAAA					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

