ThesisBot Example

AAAAAAA Petr van Blokland
AAAAAAA

Python for Designers

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

1 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[myAuthor2016]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 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. And what is more important, knowing about how programming works yourself, can actually save you a lot of time. Even if you don’t want to be a pro-grammer. The course is set up as a growing environment. Because the development

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 direction 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 bandwagon, you will pay the price as it is at that moment, based on the volume of the content at that moment. Relatively low in the beginning, putting your trust in the expectation

3 we’ll develop the course further. We start with 2

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 of the examples will try to stay in sync with changes in the outside world. To what extent this will succeed is a future promise, but by joining in at early stage, you express the trust that this will happen. As a reward for this trust you get all future content for the current price.

AAAAAAA

Subhead here

AAAAAAA This course is the twin of Processing for Designers course. Much of text is the same, as the structure of the Processing and Python is very similar. Also the code examples are very much alike, except that they are adapted to the syntax of each language. And in the advanced part of the courses the examples start to drift apart, because the available functions and libraries is different. You can decide to go through both courses if you want to learn the differences. But if you already have a preference or you made a choice, then following only one of the two courses may be sufficient as a start. If you are starting fresh on programming, the choice can be based on the expertise that is available in your environment, that is a very practical reason. You choice also be based on the difference in flavor between the languages. In preparation of deepening in each of there languages here is a brief summary about their characteristics. Processing is based on Java, an industrial strength programming language, where the type of objects needs to be specified at the start of a program. Python has a much more free usage of types, which makes it good for “sketchy” programming, but it is less reliable in circumstances where the prediction of flawless execution is important. But in reverse, this makes Python much more flexible in the storage of information. Especially the mixing of data type and the storage in the standard dictionary type, allow Python to build data structures that are very hard to achieve in Processing.

AAAAAAA

Subhead here

AAAAAAA The origin of Processing is more in the processing of images, – focussed on pixels 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 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.

AAAAAAA AAAAAAA