



based on the GitHub pushes over

ages at some point in 2018. In
a programming language using
the programming language

fault language of choice

Paul Jansen [148], 2025

tion, chances are that Python
survey [356], Python was the
L/CSS. In GitHub's Octoverse
programming language, ranking

elligence (AI) [261], Machine
imization, which are among the
Octoverse report [109] states
ity.

th research and application de-
[183, 224], Scikit-learn [228,
plotlib [139, 141, 149, 221],
ny Python packages supporting
ses (DBs) [330], or support for
sks, you can find suitable and

a simple and clean syntax and
to declare datatypes explicitly³.
Thus, Python was also named
e in the aforementioned Stack
age makes it somewhat slower
a much easier workflow when
possible to interactively write
ommands in a terminal instead
ake Python a good choice for

because I am its developer.



[course website]

o/programmingWithPython

(TeX Live 2023/Debian)

CONTENTS

v

Programming with Python

Thomas Weise (汤卫思)

May 8, 2025



Abstract

The goal of this book is to teach practical programming with the Python language to high school, undergraduate, and graduate students alike. Hopefully, readers without prior knowledge can follow the text. Therefore, all concepts are introduced using examples and discussed comprehensively. All examples are available online in the GitHub repository associated with this book, so that readers can play with them easily. Actually, the goal of the book is not just to teach programming, but to teach programming as a part of the software development process. This means that from the very beginning, we will attempt to push the reader towards writing clean code with comments and documentation as well as to use various tools for finding potential issues. While this book is work in progress, we hope that it will eventually teach all the elements of Python software creation. We hope that it can enable readers without prior programming experience to develop beautiful and maintainable software.

60

b c d e f
+ , - . /
; < = > ?
X L M N O
[\] ^ _
k l m n o
() ~

> r j f d
j ~ s p

格 佬 佬 佬 佬
姚 佼 佼 佼 佼

故 阮 妍 旭 姚
妹 妍 妍 妍 妍

Basic Lating characters as well as

on mark variant (`"""..."""`) is
`...'''`) [111, 328].

string. We first create the string by
enter to begin a new line.), and
begins with `"""` and the last one ends
ents exactly this three-line string.
th `"""`. The example in Figure 3.28
interpolation which spans over three

rs who want to learn how text
st-time readers can safely skip

certain fixed sizes, say, bytes that
er numbers. While Python supports
of 8 bytes, i.e., 64 bits. The `float`
ereted differently in order to facilitate
with text?

othing but a list of these numbers.
en knows how to interpret these numbers as characters. Maybe the most well-known
ping is ASCII [6, 321], which, however, contained only latin characters, punctuation marks,

69

tion of π using the method of

```
\u2248(pi).")
, i.e.,  $e=6$ .
radius is also 1.
48(e * s / 2).")
```

```
the side length.
48(e * s / 2).")
```

```
the side length.
48(e * s / 2).")
```

```
the side length.
48(e * s / 2).")
```

```
the side length.
48(e * s / 2).")
```

```
48(e * s / 2).")
```

ven in Listing 4.3.

```
89793.
1058285412302498.
132628613281237.
.139350203046872.
14103195089053.
3.1414524722853443.
```

(4.1)

(4.2)

o type the numbers and com-
into a program, as illustrated
the side length to `s = 1`, still
`s == 2`, which is equivalent to
`- sqrt(4 - (s ** 2)))` have
approximated value of π as
 \approx via the escapes `\u03c0` and
d prints the greek character π

in the `stdout` in Listing 4.4). Either way, since Equations 4.1 and 4.2 are always the same, we can simply copy-paste the lines of code for updating `s`, `e`, and printing the approximated value of π several times.

Listing 4.4 shows the standard output stream (`stdout`) produced by this program. Indeed, each new approximation comes closer to π . For 192 edges, we get the approximation 3.1414524722853443.