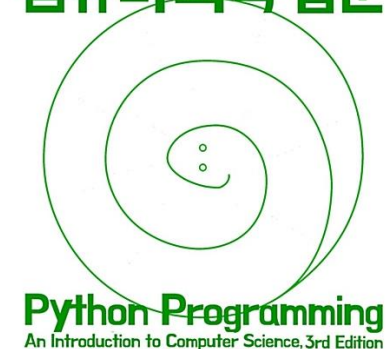


파이썬 기본 문법 학습

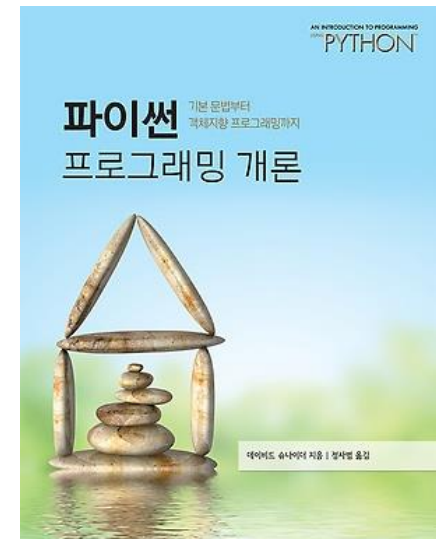
학습 자료

- 두 책자의 소스를 학습
 - 파이썬으로 시작하는 컴퓨터 과학 입문
 - 파이썬 프로그래밍 개론
 - (An Introduction to Programming using Python)
- 웹
 - Realpython.com
 - <https://realpython.com/>
 - Github.com/realpython
 - 파이썬 자습서(튜토리얼)
 - <https://docs.python.org/ko/3/tutorial/index.html>
 - 점프 투 파이썬
 - <https://wikidocs.net/book/1>
 - 파이썬 코딩 도장
 - <https://dojang.io/course/view.php?id=7>
 - 파이썬 생활 코딩
 - <https://opentutorials.org/course/1750/9609>

파이썬으로 시작하는 컴퓨터 과학 입문



프로그래밍인사이드

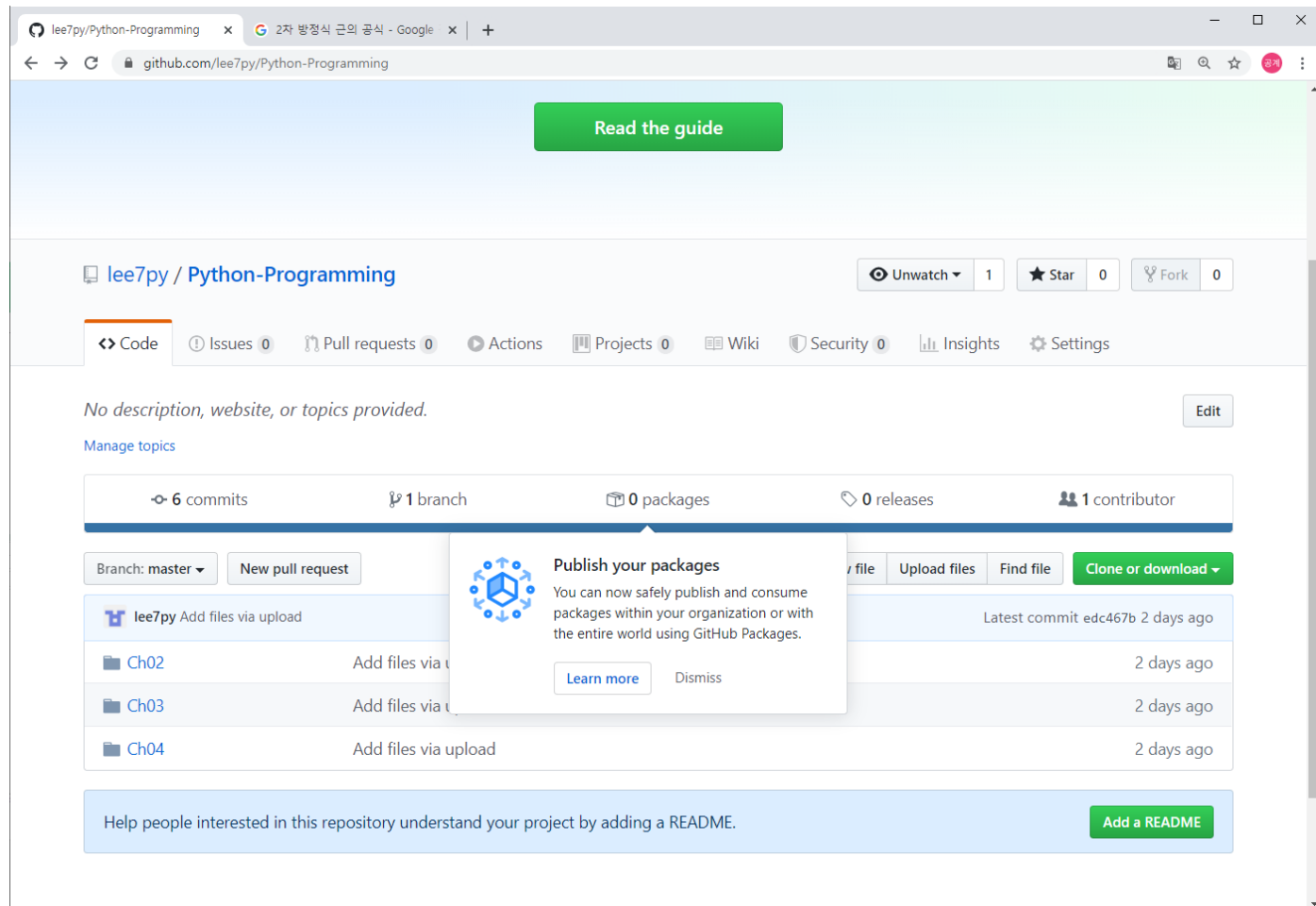


코딩 시작 전

- Github 계정 만들기
 - 구글이나 원드라이브 계정 만들기
- Github 로그인

파이썬 기본 코딩 연습

- [Github.com/lee7py](https://github.com/lee7py/Python-Programming)



깃허브 저장소

- **Repository**

- <https://github.com/lee7py/Python-Programming/>

The screenshot shows the GitHub interface for the repository 'lee7py / Python-Programming'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security' (0), 'Insights', and 'Settings'. The main content area shows the file path 'Python-Programming / Ch02 / 2-1-1.py' and a 'Jump to' dropdown. Below the path, there is a section for 'Add files via upload' with a commit hash 'f327446' and a timestamp '6 days ago'. A '1 contributor' section is also visible. At the bottom, the file content is displayed, showing two lines of Python code: 'print(3 + 2, 3 - 2, 3 * 2)' and 'print(8 / 2, 8 ** 2, 2 * (3 + 4))'. The file statistics show '2 lines (2 sloc)' and '63 Bytes'. There are buttons for 'Raw', 'Blame', and 'History' next to the code.

lee7py / Python-Programming

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Branch: master Python-Programming / Ch02 / 2-1-1.py / <> Jump to Find file Copy path

lee7py Add files via upload f327446 6 days ago

1 contributor

2 lines (2 sloc) 63 Bytes Raw Blame History

```
1 print(3 + 2, 3 - 2, 3 * 2)
2 print(8 / 2, 8 ** 2, 2 * (3 + 4))
```

깃허브 저장소

- Repository

- <https://github.com/lee7py/Pythonpgm-JM>

The screenshot displays the GitHub interface for the repository `lee7py / Pythonpgm-JM`. At the top, there are buttons for `Unwatch` (1), `Star` (0), and `Fork` (0). Below this is a navigation bar with links for `Code`, `Issues` (0), `Pull requests` (0), `Actions`, `Projects` (0), `Wiki`, `Security` (0), `Insights`, and `Settings`.

The main content area shows the file path `Pythonpgm-JM / chapter01 / chaos.py` with a `Jump to` dropdown. Below the path, it indicates `lee7py` as the contributor and `174beb9` as the commit hash, dated `6 days ago`. A `1 contributor` link is also present.

The file content is displayed in a code editor with 13 lines (9 sloc) and 274 Bytes. The code is as follows:

```
1 # File: chaos.py
2 # A simple program illustrating chaotic behavior.
3
4
5 def main():
6     print("This program illustrates a chaotic function")
7     x = eval(input("Enter a number between 0 and 1: "))
8     for i in range(10):
9         x = 3.9 * x * (1 - x)
10        print(x)
11
12
13 main()
```




Realpython.com

- 기본 자료형
 - <https://realpython.com/python-data-types/>
- 문자열 형태의 정수를 정수 자료형으로 변환
 - <https://realpython.com/convert-python-string-to-int/>
- 리스트와 튜플
 - <https://realpython.com/python-lists-tuples/>
 - <https://realpython.com/courses/lists-tuples-python/>
- 문자열
 - <https://realpython.com/python-strings/>

Realpython basic python 연습

- 책 연습

- <https://github.com/realpython/python-basics-exercises>

```
Executable File | 26 lines (20 sloc) | 609 Bytes | Raw | Blame | History |   
```

```
1 # 3.2 - Screw Things Up
2 # Solutions to review exercises
3
4
5 # Exercise 1
6 # The following line won't run because of a syntax error
7 print("hi)
8
9 # We didn't close the double quotes at the end of the string.
10 # The line above needed to have been:
11 # print("hi")
12
13
14 # Exercise 2
15 ''' The following lines won't run properly,
16     even if the syntax error in the line above is corrected,
17     because of a run-time error '''
18 print(hello)
19
20 # We meant to print the string "hello";
21 # a variable named 'hello' doesn't exist yet.
22 #
23 # This line could have been:
24 #
25 # my_string = "hello"
26 # print(my_string)
```




How to Convert a Python String to int

by Alex Ronquillo ⌚ Sep 18, 2019 💬 4 Comments 🏷️ basics python

 Tweet  Share  Email

Table of Contents

- Representing Integers in Python
- Converting a Python String to an int
- Converting a Python int to a String
- Conclusion

https://realpython.com/python-strings/

- 검색 realpython string

The screenshot shows the Real Python website with the article 'String Manipulation' open. The page has a dark blue header with the Real Python logo and navigation links. The main content area is white with a light blue sidebar on the right. The article text explains string operators, specifically the '+' operator for concatenation. A code block shows Python code for string concatenation and printing. The sidebar contains a 'Table of Contents' with links to 'String Manipulation', 'bytes Objects', and 'Conclusion'. There are also social media share buttons, a 'Recommended Video Course' section, and a 'Watch Now' button for a video course. At the bottom, there is a green 'Improve Your Python' button and a small red notification icon.

Strings and Character Data in Python

realpython.com/python-strings/

Real Python Start Here Learn Python Store More Search Join Sign-In

Stuck at home? Enjoy free courses, on us →

String Manipulation

The sections below highlight the operators, methods, and functions that are available for working with strings.

String Operators

You have already seen the operators `+` and `*` applied to numeric operands in the tutorial on [Operators and Expressions in Python](#). These two operators can be applied to strings as well.

The `+` Operator

The `+` operator concatenates strings. It returns a string consisting of the operands joined together, as shown here:

```
Python >>> s = 'foo'
>>> t = 'bar'
>>> u = 'baz'

>>> s + t
'foobar'
>>> s + t + u
'foobarbaz'

>>> print('Go team' + '!!!')
Go team!!!
```

The `*` Operator

The `*` operator creates multiple copies of a string. If `s` is a string and `n` is an integer, either of the following expressions returns a string consisting of `n` concatenated copies of `s`:

```
s * n
```

Table of Contents

- String Manipulation
- bytes Objects
- Conclusion

Tweet Share Email

Recommended Video Course

Strings and Character Data in Python

High Quality Python Video Courses Watch Now

Improve Your Python