



PYTHON-PALETTE

@yeseuly.park – 24.03.08 (1st)



To Do List

■ Installation

- *Python*
- *VSCode*
- *Anaconda*
- *GitHub*

■ Development

- *variable*
- *function*
- *If*
- *for / while*
- *assignment*

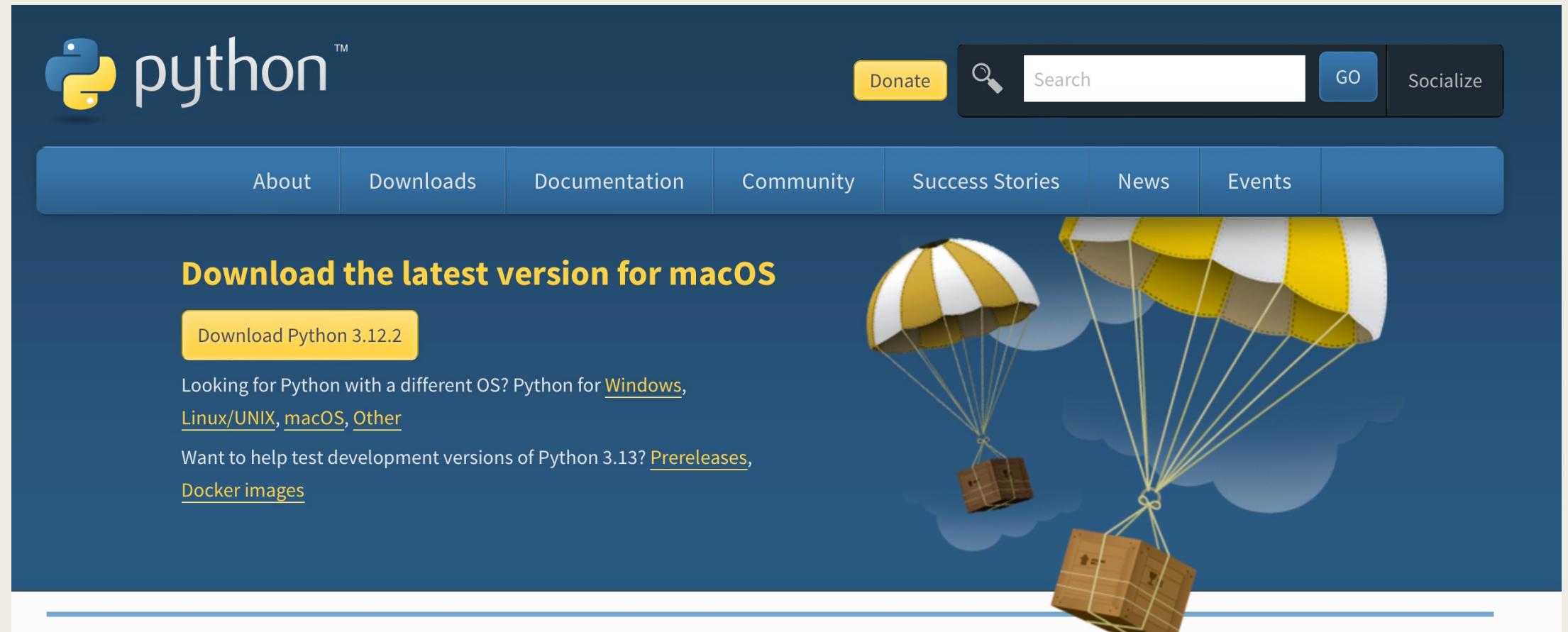
Installation

- Environmental Set-up

- Python
- VSCode
- Anaconda
- GitHub

Python – latest version 3.12.2

<https://www.python.org/downloads/>



The screenshot shows the Python.org website with a dark blue header. On the left is the Python logo. To its right is a navigation bar with links: About, Downloads, Documentation, Community, Success Stories, News, and Events. Further right are a 'Donate' button, a search bar with a 'GO' button, and a 'Socialize' link. The main content area has a dark blue background. On the left, it says 'Download the latest version for macOS' in yellow, followed by a yellow button 'Download Python 3.12.2'. Below this are links for other operating systems and pre-releases. On the right is a large illustration of two yellow and white striped parachutes carrying cardboard boxes.

python™

Donate

Search

GO

Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for macOS

Download Python 3.12.2

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [macOS](#), [Other](#)

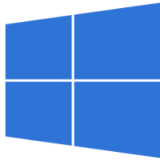
Want to help test development versions of Python 3.13? [Prereleases](#),
[Docker images](#)

VScode – check your OS (dependancies)

<https://code.visualstudio.com/download>


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows
Windows 10, 11


User Installer	x64	Arm64
System Installer	x64	Arm64
.zip	x64	Arm64
CLI	x64	Arm64



↓ .deb
Debian, Ubuntu

↓ .rpm
Red Hat, Fedora, SUSE

.deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		
CLI	x64	Arm32	Arm64

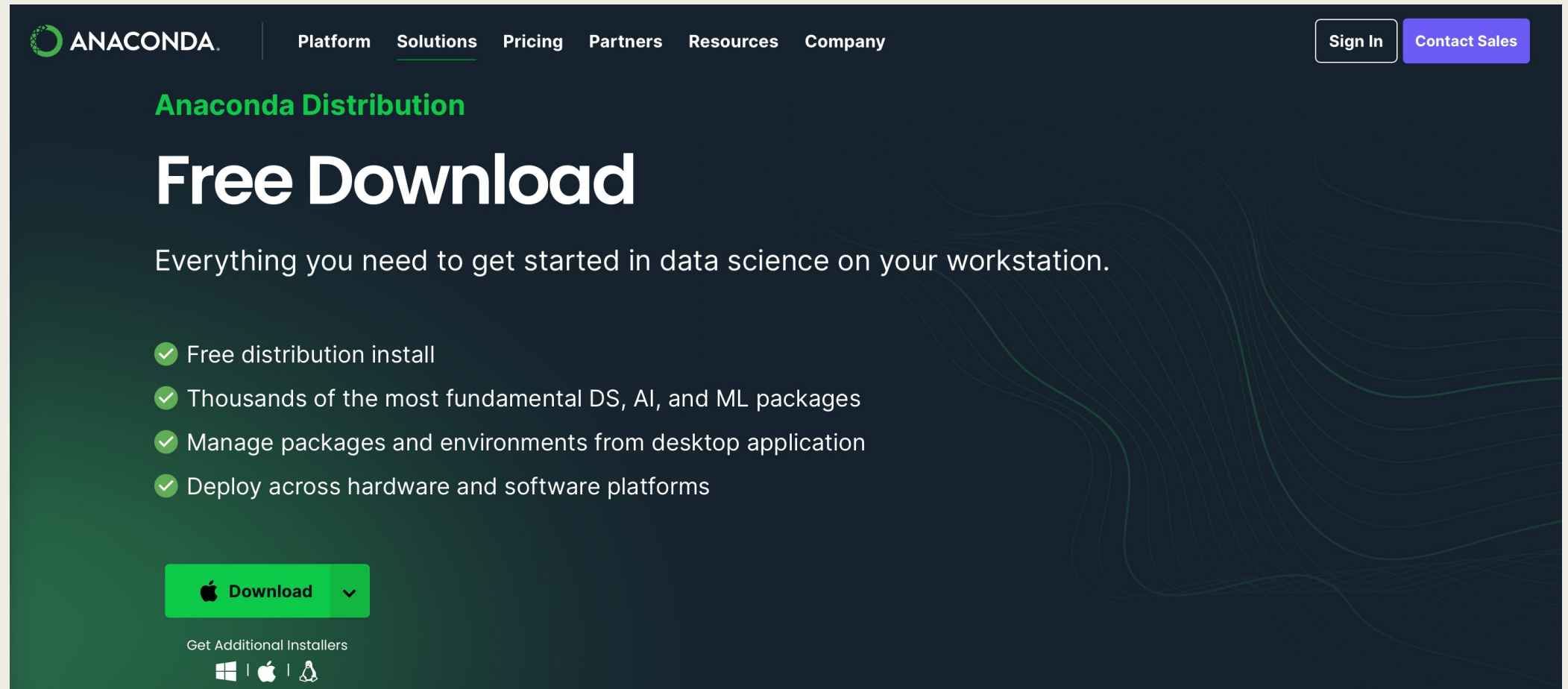


↓ Mac
macOS 10.15+

.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	

Anaconda – check your OS (dependancies)

<https://www.anaconda.com/download>



The screenshot shows the Anaconda website's download page. The header features the Anaconda logo and navigation links: Platform, Solutions (highlighted), Pricing, Partners, Resources, and Company. On the right, there are 'Sign In' and 'Contact Sales' buttons. The main content area has a dark blue background with green wavy lines. It features the text 'Anaconda Distribution' in green, followed by 'Free Download' in large white letters. Below this is the tagline 'Everything you need to get started in data science on your workstation.' and a list of four benefits, each preceded by a green checkmark. At the bottom, there is a green 'Download' button with a dropdown arrow, and a section titled 'Get Additional Installers' with icons for Windows, macOS, and Linux.

ANACONDA

Platform Solutions Pricing Partners Resources Company

Sign In Contact Sales

Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Download

Get Additional Installers

Windows | macOS | Linux

Anaconda – check your OS (dependancies)

Powered by the most recommended and trusted open-source packages



Data Exploration and Transformation

- pandas
- Intake
- NumPy
- Dask
- Apache Airflow



Visualization

- Matplotlib
- seaborn
- Plotly
- Bokeh
- HoloViews




AI and Machine Learning

- scikit-learn
- TensorFlow
- Keras
- PyTorch
- XGBoost

VScode – workspace set-up

1. Extension Installation



Python v2024.2.1
Microsoft  microsoft.com |  114,386
IntelliSense (Pylance), Linting, Debugging...

[Disable](#)  [Uninstall](#)  [Switch to Pre-Release](#)

This extension is enabled globally.


[DETAILS](#) [FEATURE CONTRIBUTIONS](#) [CHANGELOG](#) [EXTENSION PACK](#) [F](#)



Python extension for Visual Studio Code



A [Visual Studio Code](#) extension with rich support for the [Python language](#) (for all [actively supported versions](#) of the language: ≥ 3.7), including features such as IntelliSense (Pylance), linting,

Categories

- Programming Languages
- Linters
- Debuggers
- Formatters
- Data Science
- Machine Learning




Jupyter v2024.1.1
Microsoft  microsoft.com |  75,540,
Jupyter notebook support, interactive pr...


[Disable](#)  [Uninstall](#)  [Switch to Pre-Release](#)



This extension is enabled globally.

[DETAILS](#) [FEATURE CONTRIBUTIONS](#) [CHANGELOG](#) [RUNTIME STATUS](#)

Extension Pack (4)



Jupyter Keymap
Jupyter keymaps for notebooks


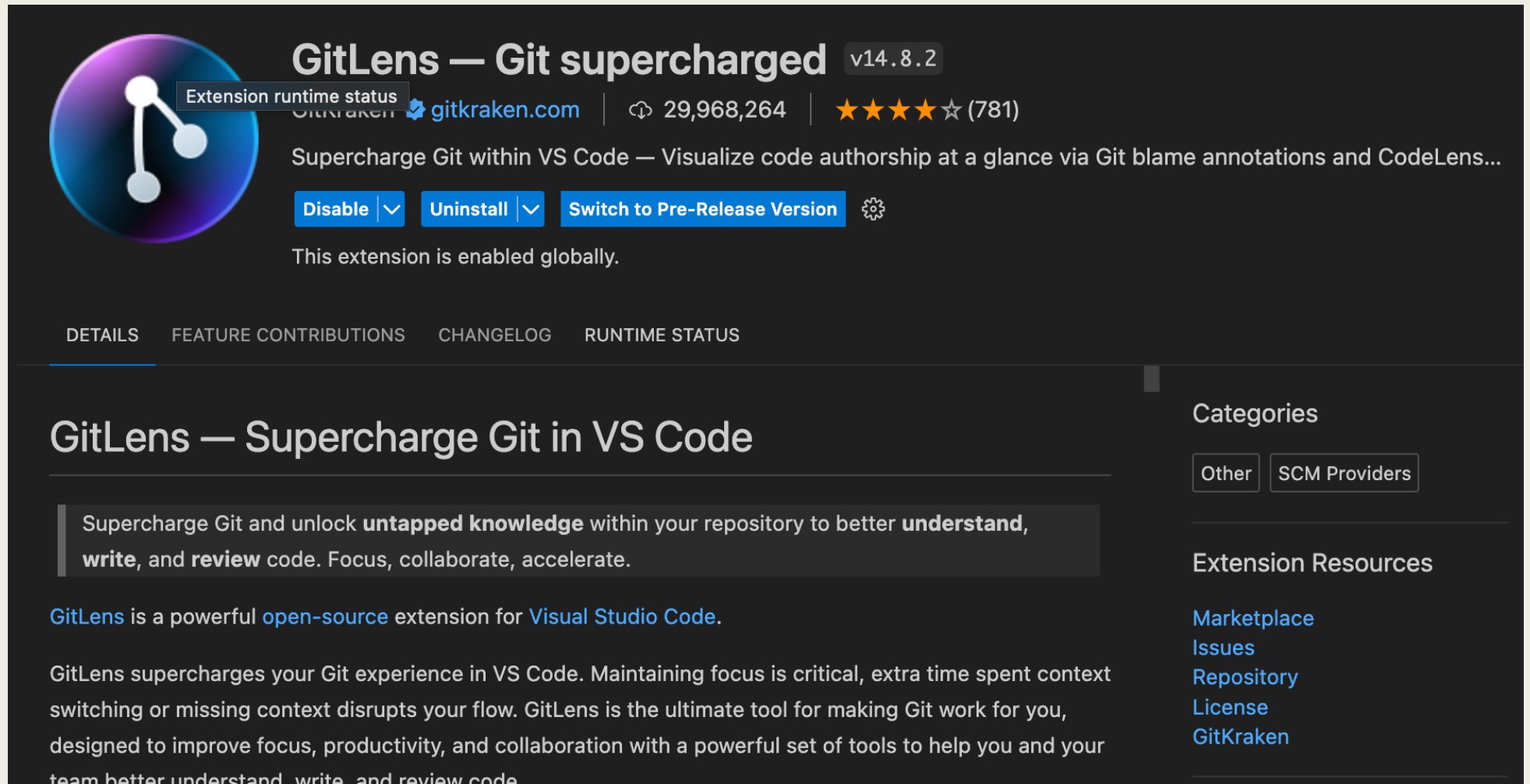
Jupyter Keymap v1.1.2
 57.1M |  4
Jupyter keymaps for notebooks

Categories

- Extension Packs
- Data Science
- Machine Learning
- Notebooks
- Visualization

VScode – workspace set-up

1. Extension Installation



The screenshot shows the GitLens extension page in VS Code. At the top, there's a header with the GitLens logo (a blue circle with a white Git branching diagram), the title "GitLens — Git supercharged" in white, and the version "v14.8.2" in a grey box. Below the title, it says "gitkraken.com" with a GitHub icon, "29,968,264" downloads, and a 4.5-star rating from 781 reviews. A description reads: "Supercharge Git within VS Code — Visualize code authorship at a glance via Git blame annotations and CodeLens...". There are three buttons: "Disable" (with a dropdown arrow), "Uninstall" (with a dropdown arrow), and "Switch to Pre-Release Version" (with a gear icon). Below these buttons, it says "This extension is enabled globally.".

Below the header, there's a navigation bar with four tabs: "DETAILS" (selected), "FEATURE CONTRIBUTIONS", "CHANGELOG", and "RUNTIME STATUS".

The main content area has the title "GitLens — Supercharge Git in VS Code" in white. Below it, a dark grey box contains the text: "Supercharge Git and unlock **untapped knowledge** within your repository to better **understand**, **write**, and **review** code. Focus, collaborate, accelerate."

Below this, it says "GitLens is a powerful [open-source](#) extension for [Visual Studio Code](#)."

Further down, it says: "GitLens supercharges your Git experience in VS Code. Maintaining focus is critical, extra time spent context switching or missing context disrupts your flow. GitLens is the ultimate tool for making Git work for you, designed to improve focus, productivity, and collaboration with a powerful set of tools to help you and your team better understand, write, and review code."

On the right side, there's a "Categories" section with two buttons: "Other" and "SCM Providers". Below that, there's an "Extension Resources" section with links: "Marketplace", "Issues", "Repository", "License", and "GitKraken".

Vscode – workspace set-up

2. Gitlens

The screenshot displays the VS Code interface for a workspace named 'Python-palette'. The Explorer sidebar on the left shows a file tree with folders 'diary', 'environment', 'examples', 'homework', and 'materials', and files 'test.py' and 'README.md'. The main editor area shows the 'README.md' file with the following content:

```
1 # Python-palette
2 - __author: @yeseuly.park
3 - __date: since 24.03.08
4 - __contents: Basic Python
5
6 ## 1) Environment
7 - Python: 3.12.2(latest version: 24.03.08)
8   - Packages
9     : To be continued..
10 - Conda Environment (Anaconda)
11   You, 60 minutes ago • Uncommitted changes
12
13 ## 2) Contents
14 - Python Grammar
15 - Data Processing
16 - Data Analysis
```

The right sidebar shows a preview of the 'README.md' file, displaying the rendered content:

Python-palette

- __author: @yeseuly.park
- __date: since 24.03.08
- __contents: Basic Python

1) Environment

- Python: 3.12.2(latest version: 24.03.08)
 - Packages : To be continued..
- Conda Environment (Anaconda)

2) Contents

- Python Grammar
- Data Processing

The bottom panel shows the Gitlens extension interface. The 'COMMIT GRAPH: PYTHON-PALETTE' view displays a commit history with a single commit 'Initial commit' by 'yeseuly' on the 'main' branch. The 'COMMIT GRAPH DETAILS' panel shows the commit message 'Work in progress' and the commit hash '13c32ed'. The 'Initial commit' button is visible. The 'AUTOLINKS' panel shows 0 found links. The 'FILES CHANGED' panel shows 1 file changed: 'README.md'.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

✓ **COMMIT GRAPH: PYTHON-PALETTE** ✨



Python-palette



main



↻ Fetch



Try t



Search commits (↑↓ for history), e.g. "Updates dependencie .ab_ Aa No results



BRANCH / TAG

GRAPH

COMMIT MESSAGE

AUTHOR



Work in progress + 1



main



Initial commit

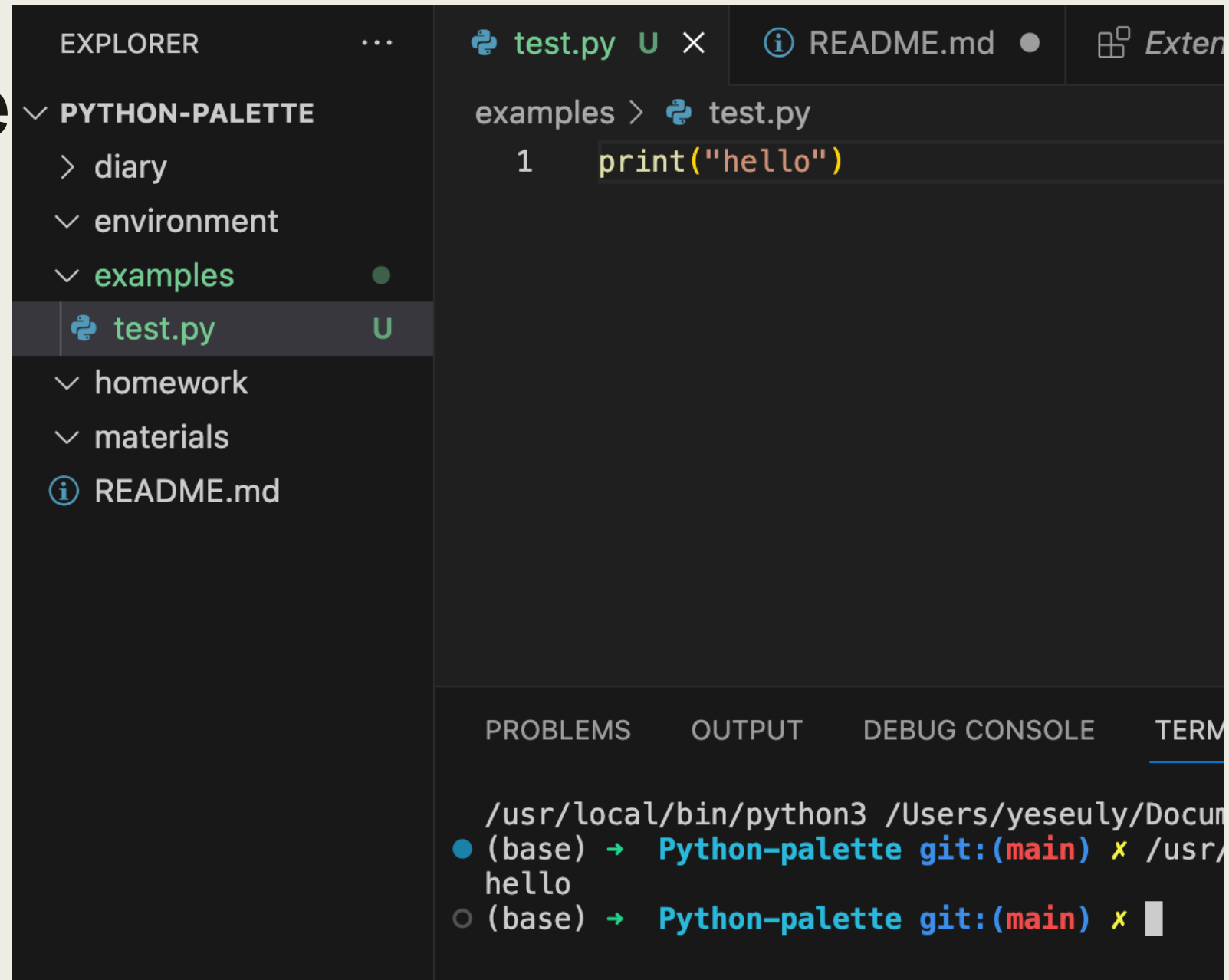
yeseuly



1

VScode

3. Conda



The screenshot displays the VS Code interface with a dark theme. On the left, the Explorer sidebar shows a project structure with folders 'diary', 'environment', 'examples', 'homework', and 'materials', and files 'test.py' and 'README.md'. The 'examples' folder is expanded, and 'test.py' is selected. The main editor area shows the content of 'test.py', which contains a single line of Python code: `print("hello")`. The bottom panel shows the 'TERMINAL' view, which displays the output of a command executed in a Conda environment. The terminal text is as follows:

```
/usr/local/bin/python3 /Users/yeseuly/Docum
● (base) → Python-palette git:(main) x /usr/
hello
○ (base) → Python-palette git:(main) x
```

GitHub – make your own ID

<https://github.com/yeseuly/Python-palette>

The screenshot shows the GitHub interface for a repository named 'Python-palette' owned by 'yeseuly'. The repository is public and has 0 stars, 0 forks, and 1 watch. The main branch is 'main'. The repository contains a single commit by 'yeseuly' with the message 'Initial commit' and a file named 'README.md'. The README content is 'Python-palette'. The right sidebar shows the 'About' section with the text 'No description, website, or topics provided.' and the 'Releases' section with the text 'No releases published' and a link to 'Create a new release'.

Python-palette Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

yeseuly Initial commit 13c32ed · 1 hour ago 1 Commits

README.md Initial commit 1 hour ago

README

Python-palette

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published

[Create a new release](#)

Development

- Basic Python

- variable
- function
- If
- for / while
- assignment

Variables – definition

Grammar

→ “Variable Name = Variable Value”

```
a = 3  
print(a)
```

3

```
a = 3  
b = 5  
c = a + b  
print(c)
```

8

Variables – int/float

```
a = 3  
b = 2.5
```

```
print(type(a))
```

```
<class 'int'>
```

```
print(type(b))
```

```
<class 'float'>
```

```
a = 7  
b = 2  
  
print(a + b)
```

```
9
```

```
print(a - b)
```

```
5
```

```
print(a % b)
```

```
1
```


Variables – string

```
a = '안녕 파이썬'  
print(a)
```

안녕 파이썬

```
print(type(a))
```

```
<class 'str'>
```

```
a = 1  
b = '1'  
c = 1.0  
d = "1.0"
```

```
print(type(a))  
print(type(b))  
print(type(c))  
print(type(d))
```

```
<class 'int'>  
<class 'str'>  
<class 'float'>  
<class 'str'>
```

Variables – string

```
a = '안녕 파이썬'  
b = 1  
print(a + b)
```

```
a = '안녕'  
b = '파이썬'  
print(a + b)
```

안녕파이썬

```
a = '안녕'  
b = '파이썬'  
c = ' '  
print(a + c + b)
```

안녕 파이썬

TypeError

Traceback (most recent **call last**)

<ipython-input-5-16efbab39763> in <module>

1 a = '안녕 파이썬'

2 b = 1

----> 3 print(a + b)

TypeError: can **only** concatenate str (**not** "int") **to** str

Variables – string

```
a = 3
b = 5
c = 8
print(a + ' 더하기 ' + b + '는 ' + c + '이에요~')
```

```
a = 3
print(type(str(a)))
```

```
<class 'str'>
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-14-caeac29d5648> in <module>
      2 b = 5
      3 c = 8
----> 4 print(a + ' 더하기 ' + b + '는 ' + c + '이에요~')
```

TypeError: unsupported operand **type**(s) **for** +: 'int' and 'str'

Variables – string

```
var = '안녕하세요. 반갑습니다.'
```

```
print(var[0])  
print(var[2])  
print(var[8])
```

안
하
갑

- 안:0
- 녕:1
- 하:2
- 세:3
- 요:4
- .:5
- 공백:6
- 반:7
- 갑:8
- 습:9
- 니:10
- 다:11
- .:12

Function – declaration

```
1  >>> def 함수이름(인자1, 인자2, ...):  
2  ...     # 함수의 본문  
3  ...     # return 반환값  
4  
5  >>> 변수 = 함수이름(인자1, 인자2, ...) #함수 호출
```

```
1  >>> def add(a, b):  
2  ...     result = a + b  
3  ...     return result  
4  
5  >>> print(add(3, 4)) # 출력 : 7
```

```
1  >>> def print_hello():  
2  ...     print("Hello, World!")  
3  
4  >>> print_hello() # 출력 : Hello, World!
```

IF – declaration

```
1  if 조건식 1:
2      코드 블록 1
3  elif 조건식 2:
4      코드 블록 2
5  ...
6  elif 조건식 n:
7      코드 블록 n
8  else:
9      코드 블록 n+1
```

```
1  x = 5
2  if x > 0:
3      print("양수입니다")
```

```
1  x = 0
2  if x > 0:
3      print("양수입니다")
4  elif x == 0:
5      print("0입니다")
6  else:
7      print("음수입니다")
```

```
1  x = 5
2  if x > 0:
3      print("양수입니다")
4  else:
5      print("음수입니다")
```

IF – declaration

```
1  fruits = ["apple", "banana", "cherry"]
2
3  if "apple" in fruits:
4      print("apple is in the fruits list")
5  else:
6      print("apple is not in the fruits list")
```

IF – declaration

```
1  numbers = [1, 2, 3, 4, 5]
2
3  if 6 in numbers:
4      print("6 is in the numbers list")
5  elif 7 in numbers:
6      print("7 is in the numbers list")
7  else:
8      print("6 and 7 are not in the numbers list")
```


FOR – declaration

```
1  >>> for 변수 in 범위:  
2      >>>     수행할 문장1  
3      >>>     수행할 문장2
```

```
1  #range를 이용한 반복문  
2  >>> for i in range(1, 11):  
3      >>>     print(i)  
4  
5  #리스트를 이용한 반복문  
6  >>> fruits = ["사과", "바나나", "포도"]  
7  >>> for fruit in fruits:  
8      >>>     print(fruit)  
9  
10 #문자열을 이용한 반복문  
11 >>> text = "파이썬"  
12 >>> for character in text:  
13 >>>     print(character)
```

FOR – declaration

```
1  >>> for 변수1 in 반복 가능한 객체 1:
2  >>>     for 변수2 in 반복 가능한 객체 2:
3  >>>         실행할 코드
```

```
1  >>> for i in range(2, 10):
2  >>>     for j in range(1, 10):
3  >>>         print(f"{i} * {j} = {i * j}")
4  >>>     print()
```

WHILE – declaration

```
1  >>> while 조건1:
2  >>>     while 조건2:
3  >>>         실행할 코드
4  >>>     조건1에 관련된 변수 변경
```

```
1  >>> i = 2
2
3  >>> while i < 10:
4  >>>     j = 1
5  >>>     while j < 10:
6  >>>         print(f"{i} * {j} = {i * j}")
7  >>>         j += 1
8  >>>     print()
9  >>>     i += 1
```

```
1  >>> n = 1
2
3  >>> while True:
4  >>>     if n > 10:
5  >>>         break
6  >>>     print(n)
7  >>>     n += 1
```

Assignment 1