

# 8

# Best AI Tools for Code Generation

Embrace the Future of Coding





# GitHub Copilot

- AI code completion tool that helps to write code faster and with fewer errors
- Trained on a massive dataset of code
- Generate code in a variety of programming languages, including Python, Java, JavaScript, and C++

The screenshot shows a GitHub Copilot interface integrated into a dark-themed code editor (VS Code). On the left, a sidebar displays a list of recent repositories and a GitHub Copilot section. The main area shows a Python file named 'module.py' with code for parsing expense strings. A tooltip from GitHub Copilot provides documentation for the 'parse\_expenses' function, explaining its purpose, parameters, and examples. The code editor also shows a snippet of test code for the 'TestParseExpenses' class.

```

import datetime
def parse_expenses(expenses_string):
    """
    Parse the list of expenses and return the list of triples (date, amount, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2023-01-02 -34.01 USD
        2023-01-03 2.59 DKK
        2023-01-03 -2.72 EUR
    """
    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split(" ")
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                        float(value),
                        currency))
    return expenses
expenses_data = '''2023-01-02 -34.01 USD
2023-01-03 2.59 DKK
2023-01-03 -2.72 EUR'''
```

Swipe >>>



# ChatGPT

- Exhibits its capability by generating efficient and readable lines of code.
- Optimize the programming process by leveraging pattern analysis in existing code.



I got this when running 'node random-quote-api.js' Error: Cannot find module 'express'



The error message "Cannot find module 'express'" indicates that Node.js is unable to find the required module 'express', which is used to create the web server and handle HTTP requests in the Random Quote Generator API.



To fix this error, you need to install the 'express' module using npm (Node Package Manager). Open your terminal and navigate to the directory where your `random-quote-api.js` file is saved, then run the following command:

Copy code

```
npm install express
```

Swipe >>>



# OpenAI Codex

- Used to generate entire programs from natural language descriptions.
- Trained on a massive dataset of code and can generate code in a variety of programming languages, including Python, Java, JavaScript, and Go.

The screenshot shows a dark-themed code editor with a tab bar at the top containing four files: parse\_expenses.py (selected), write\_sql.go, sentiments.ts, and addresses.rb. The main pane displays the following Python code:

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, value, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2016-01-02 -34.01 USD
9         2016-01-03 2.59 DKK
10        2016-01-03 -2.72 EUR
11    """
12    expenses = []
13    for line in expenses_string.splitlines():
14        if line.startswith("#"):
15            continue
16        date, value, currency = line.split(" ")
17        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
18                         float(value),
19                         currency))
20
21    return expenses
```

Swipe >>>



# Tabnine

- Offers predictive text and code completion
- Suggest the next lines or blocks of code to speed up the development process
- Grasp the context in which the code is written, providing relevant suggestions even in complex development scenarios

```
1 import os
2 import sys
3
4 # Count lines of code in the given directory, separated by file extension
5 def main(directory):
6     line_count = {}
7     for filename in os.listdir(directory):
8         _, ext = os.path.splitext(filename)
9         if ext not in line_count:
10             line_count[ext] = 0
11             for line in open(os.path.join(directory, filename)):
12                 line_count[ext] += 1
13             for ext in line_count.keys():
14                 Tab 12%
15                 ext in line_count.keys    Tab+Tab 12%
16                 filename in os.listdir   Tab+3  2%
17                 ext in                  Tab+4  19%
18                 filename               Tab+5  5%
19
```

Swipe >>>



# Seek

- Used to generate code snippets, functions, and even entire programs from natural language descriptions.
- Trained on a massive dataset of code and can generate code in a variety of programming languages, including Python, Java, JavaScript, and C++.

The screenshot shows the Seek AI interface. On the left, under the heading "Questions", there is a list of six prompts:

- What products had the most sales last month?
- What products had the most sales last month?
- Which customers made the most purchases last quarter?
- What products received the most complaints?
- How many employees started over the last few years?
- Show me our performance from the last five quarters.

On the right, under the heading "Code", is the generated SQL query:

```
select ID, PRODUCT_NAME, SALES from PRODUCT_DATABASE
PUBLISHED_2022 order by SALES desc limit 10;
```

Below the code, under the heading "Data", is a table showing sales data for three products:

Product	Sales
Product 1	\$150,000
Product 2	\$110,400
Product 3	\$100,000

Swipe >>>



# Enzyme

- Designed for frontend web development.
- Used to generate React components, HTML, and CSS from natural language descriptions.
- Build better tools to bring web3 closer to creators

The screenshot shows the Enzyme ENS Dashboard. At the top, there's a navigation bar with tabs like 'Contracts', 'ENS', 'Manager', and 'Contracts'. Below the navigation is a large header with the text 'ENS Dashboard to manage' and a subtext explaining ENS as the DNS blockchain equivalent. To the right of this header are four main sections: 'Commitments' (with a 'Pending' tab), 'Bent Price' (with tabs for 'Price', 'Supply', and 'Demand'), 'Available' (with tabs for 'Name' and 'String'), and 'Transfer Ownership' (with tabs for 'Name' and 'Address'). A large blue and white flame logo is positioned at the bottom left of the dashboard area.

Swipe >>



# Kite

- Aims to assist software developers by providing them with relevant code suggestions in real-time
- Uses Machine Learning models to predict the next line of code
- Popular for its Co-Pilot feature, which assists developers in suggesting entire lines or blocks of code

```
import os
import sys

def count_py_files_in_repos(dirname):
    if os.path.exists(os.path.join(dirname, '.git')):
        count = 0
        for root, dirs, files in os.walk(dirname):
            count += len([f for f in files if f.endswith('.py')])
        print('{} has {} Python files'.format(dirname, count))
    for name in os.listdir(dirname):
        path = os.path.join(dirname, name)
        if os.path.isdir(path):
            cou|
            count_py_files_in_repos...
            count
```

Swipe >>>



# Codota

- Provides intelligent code completion, suggestions, and inline documentation by learning from existing code.
- Analyze code from various repositories and learning patterns.
- Supports various languages and can be integrated into IDEs like IntelliJ IDEA, Android Studio, and Eclipse

```
public static void main(String[] args) throws Exception {
    String filePath = args[0];
    File file = n|
}
    ↗ new File(filePath)
    new
    ↑ null
    ↓ m new File(String pathname)
    ↑ ↗ new File(filePath, String child)
    ↓ m new File(String parent, String child)
    ↓ m new File(File parent, String child)
    m new File(URI uri)
    ↗ new File(System.getProperty("java.io.tmpdir"))
    ↗ new File(String parent, "lib")
```

Swipe >>>