

## Copilot 是微软旗下的open ai技术

<https://github.com/features/copilot/>

### 我愿称之为 – 编程界的实时搜索引擎

Copilot 能够帮助开发者在 Visual Studio、Visual Studio Code、Neovim 和 JetBrains IDE 等开发环境中写代码时自动提供建议。GitHub Copilot 支持 Python、JavaScript、TypeScript、Java、Ruby、Php 和 Go 等编程语言。可以根据上下文自动写代码，包括文档字符串、注释、函数名称、代码，只要用户给出提示，就可以写出完整的函数

### 与ChatGPT 的区别

ChatGPT 目前属于对话型AI工具，能够提供解决思路，能够像和人类一样对话交流（最后我们会演示下ChatGPT <https://chat.openai.com/>）

Copilot 更适合辅助编程，他会观测你的项目上下文来分析你下一步要进行的工作

### 使用注释开发代码

会根据个人开发风格，自动提示开发代码，支持多种代码实现方案，可选择自己喜欢的

### 基础校验

```
1  /**
2   * 判断一个字符串是否为数字的方法
3   */
4
```

### 常规算法

```
1  /**
2   * 冒泡排序
3   */
4
```

### 冒泡排序/Java

```
1  /**
2   * 冒泡排序
```

```

3      */
4      public static void bubbleSort(int[] arr) {
5          int temp = 0;
6          for (int i = 0; i < arr.length - 1; i++) {
7              for (int j = 0; j < arr.length - 1 - i; j++) {
8                  if (arr[j] > arr[j + 1]) {
9                      temp = arr[j];
10                     arr[j] = arr[j + 1];
11                     arr[j + 1] = temp;
12                 }
13             }
14         }
15     }
16

```

## 快速生成辅助函数

```

1      /**
2      * 计算两个日期之间相差的天数 方法
3      */
4

```

## 算法

```

1      /**
2      * 无重复字符串的最长子串
3      */
4

```

## 自动识别当前运行环境

### Laravel

```

1  //获取客户端IP
2  $ip = request()->getClientIp();
3

```

### Yii2

```
1 //打印客户端IP
2 dd(Yii::$app->request->userIP);
3
```

## 多语言自动生成

```
1     return [
2         'en' => [
3             'language',
4             'package',
5         ],
6         'cn' => [
7             '语言',
8             '包裹',
9         ],
10        'jp' => [
11            '言語',
12            'パッケージ',
13        ],
14        'ko' => [
15            '언어',
16            '패키지',
17        ],
18    ];
19
```

## 自动生成前端代码

### 前端语义化开发

```
1     //使用jquery 弹出 当前日期
2     var date = new Date();
3     var year = date.getFullYear();
4     var month = date.getMonth()+1;
5     var day = date.getDate();
6     var week = date.getDay();
7     var weekArr = ['日', '一', '二', '三', '四', '五', '六'];
8     var hour = date.getHours();
```

```

9         var minute = date.getMinutes();
10        var second = date.getSeconds();
11        var time = year+'年'+month+'月'+day+'日'+ ' 星期'+weekArr[week]+'
'+hour+':'+minute+':'+second;
12        alert(time);
13

```

```

1        //在 #submit 点击时 使用 layer 提示 "你没有权限"
2        $("#submit").click(function(){
3            layer.msg('你没有权限');
4        });
5

```

## 生成一个五彩斑斓的黑

这个是弄着玩的

```

1 //在页面中央画一个 五彩斑斓的黑色圆形
2 var canvas = document.getElementById("canvas");
3 var context = canvas.getContext("2d");
4 context.beginPath();
5 context.arc(250,250,100,0,Math.PI*2,true);
6 context.closePath();
7 var gradient = context.createRadialGradient(250,250,0,250,250,100);
8 gradient.addColorStop(0,"#fff");
9 gradient.addColorStop(0.1,"#f00");
10 gradient.addColorStop(0.2,"#0f0");
11 gradient.addColorStop(0.3,"#00f");
12 gradient.addColorStop(0.4,"#ff0");
13 gradient.addColorStop(0.5,"#0ff");
14 gradient.addColorStop(0.6,"#f0f");
15 gradient.addColorStop(0.7,"#f00");
16 gradient.addColorStop(0.8,"#0f0");
17 gradient.addColorStop(0.9,"#00f");
18 gradient.addColorStop(1,"#fff");
19 context.fillStyle = gradient;
20 context.fill();
21

```

## 生成五子棋

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8  <canvas id="canvas" width="500" height="500"></canvas>
9  <script>
10     //生成五子棋
11     var canvas = document.getElementById("canvas");
12     var context = canvas.getContext("2d");
13     var width = canvas.width;
14     var height = canvas.height;
15     var r = 15;
16     var me = true;
17     var over = false;
18     var chessBoard = [];
19     for (var i = 0; i < 15; i++) {
20         chessBoard[i] = [];
21         for (var j = 0; j < 15; j++) {
22             chessBoard[i][j] = 0;
23         }
24     }
25     //画棋盘
26     var drawChessBoard = function () {
27         context.strokeStyle = "#BFBFBF";
28         for (var i = 0; i < 15; i++) {
29             context.moveTo(15 + i * 30, 15);
30             context.lineTo(15 + i * 30, 435);
31             context.stroke();
32             context.moveTo(15, 15 + i * 30);
33             context.lineTo(435, 15 + i * 30);
34             context.stroke();
35         }
36     };
37     //画棋子
38     var oneStep = function (i, j, me) {
```

```
39     context.beginPath();
40     context.arc(15 + i * 30, 15 + j * 30, r, 0, 2 * Math.PI);
41     context.closePath();
42     var gradient = context.createRadialGradient(15 + i * 30 + 2, 15 + j *
30 - 2, r, 15 + i * 30 + 2, 15 + j * 30 - 2, 0);
43     if (me) {
44         gradient.addColorStop(0, "#0A0A0A");
45         gradient.addColorStop(1, "#636766");
46     } else {
47         gradient.addColorStop(0, "#D1D1D1");
48         gradient.addColorStop(1, "#F9F9F9");
49     }
50     context.fillStyle = gradient;
51     context.fill();
52 };
53 //赢法数组
54 var wins = [];
55 for (var i = 0; i < 15; i++) {
56     wins[i] = [];
57     for (var j = 0; j < 15; j++) {
58         wins[i][j] = [];
59     }
60 }
61 var count = 0;
62 //所有横线赢法
63 for (var i = 0; i < 15; i++) {
64     for (var j = 0; j < 11; j++) {
65         for (var k = 0; k < 5; k++) {
66             wins[i][j + k][count] = true;
67         }
68         count++;
69     }
70 }
71 //所有竖线赢法
72 for (var i = 0; i < 15; i++) {
73     for (var j = 0; j < 11; j++) {
74         for (var k = 0; k < 5; k++) {
75             wins[j + k][i][count] = true;
76         }
77         count++;

```

```
78     }
79 }
80 //所有斜线赢法
81 for (var i = 0; i < 11; i++) {
82     for (var j = 0; j < 11; j++) {
83         for (var k = 0; k < 5; k++) {
84             wins[i + k][j + k][count] = true;
85         }
86         count++;
87     }
88 }
89 //所有反斜线赢法
90 for (var i = 0; i < 11; i++) {
91     for (var j = 14; j > 3; j--) {
92         for (var k = 0; k < 5; k++) {
93             wins[i + k][j - k][count] = true;
94         }
95         count++;
96     }
97 }
98 //赢法统计数组
99 var myWin = [];
100 var computerWin = [];
101 for (var i = 0; i < count; i++) {
102     myWin[i] = 0;
103     computerWin[i] = 0;
104 }
105 //点击事件
106 canvas.onclick = function (e) {
107     if (over) {
108         return;
109     }
110     if (!me) {
111         return;
112     }
113     var x = e.offsetX;
114     var y = e.offsetY;
115     var i = Math.floor(x / 30);
116     var j = Math.floor(y / 30);
117     if (chessBoard[i][j] == 0) {
```

```
118         oneStep(i, j, me);
119         chessBoard[i][j] = 1;
120         for (var k = 0; k < count; k++) {
121             if (wins[i][j][k]) {
122                 myWin[k]++;
123                 computerWin[k] = 6;
124                 if (myWin[k] == 5) {
125                     window.alert("你赢了");
126                     over = true;
127                 }
128             }
129         }
130         if (!over) {
131             me = !me;
132             computerAI();
133         }
134     }
135 };
136 //计算机AI
137 var computerAI = function () {
138     var myScore = [];
139     var computerScore = [];
140     var max = 0;
141     var u = 0, v = 0;
142     for (var i = 0; i < 15; i++) {
143         myScore[i] = [];
144         computerScore[i] = [];
145         for (var j = 0; j < 15; j++) {
146             myScore[i][j] = 0;
147             computerScore[i][j] = 0;
148         }
149     }
150     for (var i = 0; i < 15; i++) {
151         for (var j = 0; j < 15; j++) {
152             if (chessBoard[i][j] == 0) {
153                 for (var k = 0; k < count; k++) {
154                     if (wins[i][j][k]) {
155                         if (myWin[k] == 1) {
156                             myScore[i][j] += 200;
157                         } else if (myWin[k] == 2) {
```



```

158         myScore[i][j] += 400;
159     } else if (myWin[k] == 3) {
160         myScore[i][j] += 2000;
161     } else if (myWin[k] == 4) {
162         myScore[i][j] += 10000;
163     }
164     if (computerWin[k] == 1) {
165         computerScore[i][j] += 220;
166     } else if (computerWin[k] == 2) {
167         computerScore[i][j] += 420;
168     } else if (computerWin[k] == 3) {
169         computerScore[i][j] += 2100;
170     } else if (computerWin[k] == 4) {
171         computerScore[i][j] += 20000;
172     }
173 }
174 }
175 if (myScore[i][j] > max) {
176     max = myScore[i][j];
177     u = i;
178     v = j;
179 } else if (myScore[i][j] == max) {
180     if (computerScore[i][j] > computerScore[u][v]) {
181         u = i;
182         v = j;
183     }
184 }
185 if (computerScore[i][j] > max) {
186     max = computerScore[i][j];
187     u = i;
188     v = j;
189 } else if (computerScore[i][j] == max) {
190     if (myScore[i][j] > myScore[u][v]) {
191         u = i;
192         v = j;
193     }
194 }
195 }
196 }

```

```
197     }
198     oneStep(u, v, false);
199     chessBoard[u][v] = 2;
200     for (var k = 0; k < count; k++) {
201         if (wins[u][v][k]) {
202             computerWin[k]++;
203             myWin[k] = 6;
204             if (computerWin[k] == 5) {
205                 window.alert("计算机赢了");
206                 over = true;
207             }
208         }
209     }
210     if (!over) {
211         me = !me;
212     }
213 };
214 //画棋子
215 var oneStep = function (i, j, me) {
216     context.beginPath();
217     context.arc(15 + i * 30, 15 + j * 30, 13, 0, 2 * Math.PI);
218     context.closePath();
219     var gradient = context.createRadialGradient(15 + i * 30 + 2, 15 + j *
30 - 2, 13, 15 + i * 30 + 2, 15 + j * 30 - 2, 0);
220     if (me) {
221         gradient.addColorStop(0, "#0a0a0a");
222         gradient.addColorStop(1, "#636766");
223     } else {
224         gradient.addColorStop(0, "#d1d1d1");
225         gradient.addColorStop(1, "#f9f9f9");
226     }
227     context.fillStyle = gradient;
228     context.fill();
229 };
230 //画棋盘
231 var drawChessBoard = function () {
232     for (var i = 0; i < 15; i++) {
233         context.moveTo(15 + i * 30, 15);
234         context.lineTo(15 + i * 30, 435);
235         context.stroke();
```

```

236         context.moveTo(15, 15 + i * 30);
237         context.lineTo(435, 15 + i * 30);
238         context.stroke();
239     }
240 };
241 drawChessBoard();
242 </script>
243 </body>
244 </html>
245

```

## 实际业务开发

### 文件检索 (智能封装)

```

1     public function actionAliPay(){
2         //递归查询指定文件夹下是否包含某个字符串 方法
3         $path = '/Users/ligen/Downloads/aliPay';
4         $this->findFile($path,'al');
5     }
6     function findFile($path,$str){
7         $handle = opendir($path);
8         while (($file = readdir($handle)) !== false) {
9             if ($file != '.' && $file != '..') {
10                 $cur_path = $path . DIRECTORY_SEPARATOR . $file;
11                 if (is_dir($cur_path)) {
12                     $this->findFile($cur_path,$str);
13                 } else {
14                     if (strpos($file,$str) !== false) {
15                         echo $cur_path . PHP_EOL;
16                     }
17                 }
18             }
19         }
20         closedir($handle);
21     }
22

```

## 语义化ORM

```
1      $page = 1;
2      $pageSize = 10;
3      //分页查询员工模型信息 并使用 em_qiye_wechat_city_id 字段排序 降序
4      $employeeModelList = EmployeeModel::find()-
>orderBy(['em_qiye_wechat_city_id'=>SORT_DESC])->offset(($page-1)*$pageSize)-
>limit($pageSize)->all();
5
```

## 业务排序

```
1      $array = [
2          '技术研发中心',
3          'IT研发中心',
4          '总裁办'
5      ];
6      //中文数组排序
7      $array = array_map(function ($v) {
8          return iconv('UTF-8', 'GBK', $v);
9      }, $array);
10
```

## 用户行为分析

### 根据当前系统上下文，自动分析代码编程风格，提供多种代码方案

这个东西有点玄乎，AI可能会根据你的代码风格，项目编程风格，猜测你的下一步，不太好演示  
。。。更多特性请自行体验

## ChatGTP

### 实验

1. 如何在Linux下 搜索指定文件夹下包含关键字的文件
2. 女朋友为什么会生气
3. Mysql 分表后如何分页查询
4. 如何处理婆媳关系

## 讨论：目前的AI真的会让程序员失业吗？

备注

Copilot 分为 免费版和付费版 ， 免费版仅限学生认证用户使用  
新用户，可免费使用60天，  
付费用户每月10\$， 每年100\$