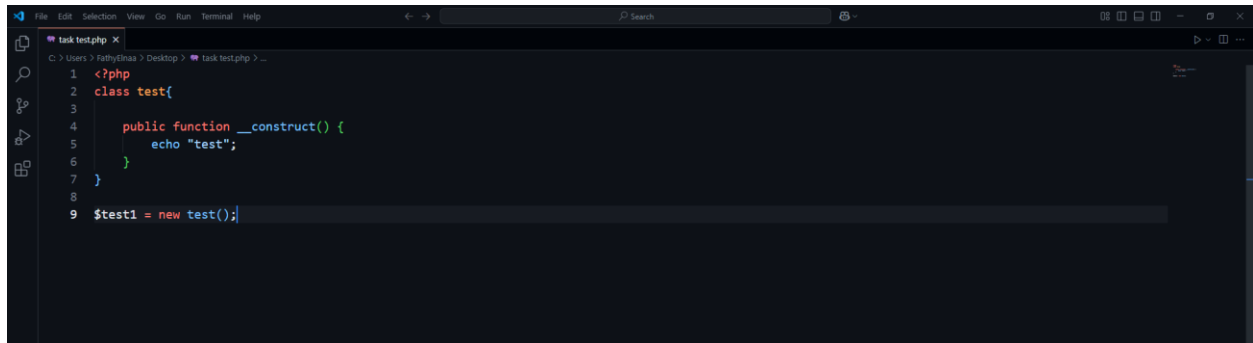


magic methods

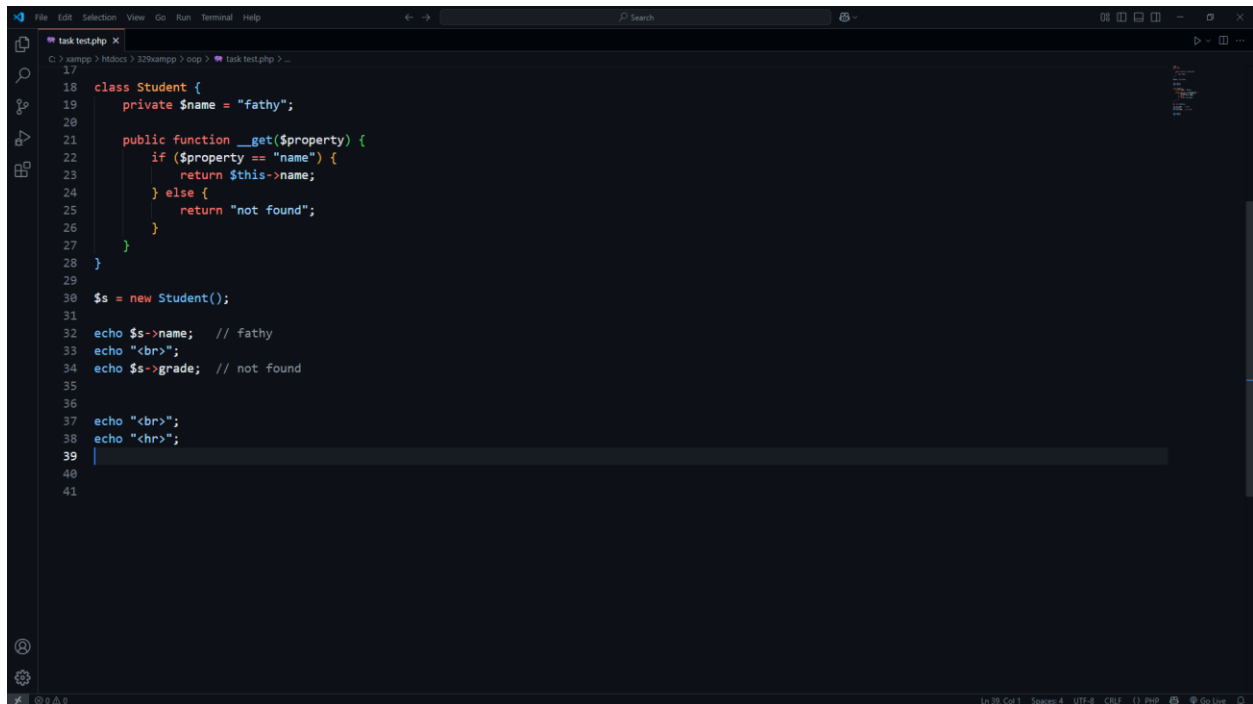
هي دوال خاصة في الـ برمجة الكائنية في php، تبدأ بـ (_)، ويتم استدعاؤها بشكل تلقائي في مواقف معينة

1- __construct(): دي دالة البناء، بتشتغل أول ما يتعمل كائن (object) من الكلاس



```
1 <?php
2 class test{
3
4     public function __construct() {
5         echo "test";
6     }
7 }
8
9 $test1 = new test();
```

2- __get(\$property): بتشتغل لما تحاول توصل لخاصية مش معرفة أو مش عامة



```
17
18 class Student {
19     private $name = "fathy";
20
21     public function __get($property) {
22         if ($property == "name") {
23             return $this->name;
24         } else {
25             return "not found";
26         }
27     }
28 }
29
30 $s = new Student();
31
32 echo $s->name; // fathy
33 echo "<br>";
34 echo $s->grade; // not found
35
36
37 echo "<br>";
38 echo "<hr>";
39
40
41
```

3- __set(\$property, \$value): بتشتغل لما تحاول تعدل خاصية مش معرفة أو محمية

```
task test.php X
C:\xampp\htdocs\325uampp > oop > task test.php > ...
14 echo <pre>;
15 echo <hr>;
16
17
18 class Student {
19     private $name;
20
21     public function __set($property, $value) {
22         if ($property == "name") {
23             $this->name = $value;
24             echo "save $value<br>";
25         } else {
26             echo "ممنوع تعديل الخاصية دي <br>";
27         }
28     }
29
30     public function showName() {
31         return $this->name;
32     }
33 }
34
35 $s = new Student();
36 $s->name = "fathy";
37 echo $s->showName(). " <br>";
38 $s->age = 20;
39
40
41
42 echo <br>;
43 echo <hr>;
44
45
46
47
48
```

4- __call(\$method, \$args): لما تحاول تستدعي دالة مش موجودة

```
task test.php X
C:\xampp\htdocs\325uampp > oop > task test.php > PHP interpreter > call > __call
14 class call {
15     public function __call($name, $arguments) {
16         echo $name ;
17         echo <pre>;
18         print_r($arguments);
19     }
20 }
21
22 $m = new call();
23 $m->sayHello("علي", 25);
24
25 echo <br>;
26 echo <hr>;
```

-5 call static : اقدر استدعاء دالة أو خاصية ثابتة (static) باستخدام معامل النطاق ::دون الحاجة إلى إنشاء كائن من الكلاس

```
71
72
73 class sta_call{
74     public static function area($num){
75         echo $num*$num;
76     }
77 }
78
79 sta_call::area(10);
80
81 echo "<br>";
82
83 class sta_call2 {
84     public static function say_hello() {
85         return "Hello from static";
86     }
87
88     public static function call_hello() {
89         return self::say_hello();
90     }
91 }
92
93 echo sta_call2 ::call_hello();
94
95 echo "<br>";
96 echo "<br>";
```

-6 clone: باستخدام clone اقدر انشئ كائن جديد مستقل له نفس الخصائص، ولكن يمكن تعديله دون التأثير على الكائن الأصلي

```
97
98
99 class Person {
100     public $name;
101 }
102
103 $person1 = new Person();
104 $person1->name = "fathy";
105
106 $person2 = clone $person1;
107 $person2->name = "mohamed";
108
109 echo $person1->name; // fathy
110 echo $person2->name; // mohamed
111
112
113
114
115 echo "<br>";
116 echo "<br>";
```

7- `__toString()` : عندما تريد تحويل كائن (Object) إلى سلسلة نصية (String) ، فإن الطريقة الصحيحة لذلك هي استخدام الدالة السحرية `__toString()` داخل الكلاس.

```
task test.php X
C:\xampp\htdocs> cd c:\xampp\oop > task test.php > ...
113
114
115 echo "<div>";
116 echo "</div>";
117
118 class User {
119     private $name;
120
121     public function __construct($name) {
122         $this->name = $name;
123     }
124
125     public function __toString() {
126         return "User name: " . $this->name;
127     }
128 }
129
130 $user = new User("Fathy");
131 echo $user; // User name: Fathy
132
133
134 echo "<div>";
135 echo "</div>";
```

8- `__invoke()` : تسمح لك إنك تتعامل مع الكائن كأنه دالة، يعني تنادي عليه بالأقواس () كأنك بتستدعي function

```
task test.php X
C:\xampp\htdocs> cd c:\xampp\oop > task test.php > ...
133
134
135 class sum {
136     public function __invoke($a, $b) {
137         return $a + $b;
138     }
139 }
140
141 $sum = new sum();
142 echo $sum(5, 7); // Output: 12
143
144
145 echo "<div>";
146 echo "</div>";
```