



Magic Methods



Magic methods

- Magic methods là các phương thức có tên bắt đầu và kết thúc với 2 dấu _
- Magic methods được thực thi tự động bởi trình thông dịch Python khi sử dụng object với các cú pháp hay built-in functions khác nhau
- Lợi ích lớn nhất của magic methods là cung cấp cú pháp đơn giản để các đối tượng (kiểu tùy chỉnh) hoạt động giống như những kiểu dữ liệu và với các hàm định nghĩa sẵn trong Python



Comparison magic methods

Magic methods	Hành vi	Ví dụ
<code>__eq__(self, other)</code>	So sánh ==	<code>obj1 == obj2</code>
<code>__ne__(self, other)</code>	So sánh !=	<code>obj1 != obj2</code>
<code>__lt__(self, other)</code>	So sánh <	<code>obj1 < obj2</code>
<code>__gt__(self, other)</code>	So sánh >	<code>obj1 > obj2</code>
<code>__le__(self, other)</code>	So sánh <=	<code>obj1 <= obj2</code>
<code>__ge__(self, other)</code>	So sánh >=	<code>obj1 >= obj2</code>



Numeric magic methods

Magic methods	Hành vi	Ví dụ
<code>__pos__(self)</code> , <code>__neg__(self)</code>	Toán tử đơn +, -	<code>+obj</code> , <code>-obj</code>
<code>__abs__(self)</code>	Gọi hàm <code>abs()</code>	<code>abs(obj)</code>
<code>__round__(self, n)</code>	Gọi hàm <code>round()</code>	<code>round(obj, 2)</code>
<code>__floor__(self)</code>	Gọi hàm <code>math.floor()</code>	<code>math.floor(obj)</code>
<code>__ceil__(self)</code>	Gọi hàm <code>math.ceil()</code>	<code>math.ceil(obj)</code>
<code>__trunc__(self)</code>	Gọi hàm <code>math.trunc()</code>	<code>math.trunc(obj)</code>



Numeric magic methods

Magic methods	Hành vi	Ví dụ
<code>__add__(self, other)</code>	Toán tử +	<code>obj1 + obj2</code>
<code>__sub__(self, other)</code>	Toán tử -	<code>obj1 - obj2</code>
<code>__mul__(self, other)</code>	Toán tử *	<code>obj1 * obj2</code>
<code>__truediv__(self, other)</code>	Toán tử /	<code>obj1 / obj2</code>
<code>__floordiv__(self, other)</code>	Toán tử //	<code>obj1 // obj2</code>
<code>__mod__(self, other)</code>	Toán tử %	<code>obj1 % obj2</code>



Numeric magic methods

Magic methods	Hành vi	Ví dụ
<code>--radd__(self, other)</code>	Toán tử +	<code>obj2 + obj1</code>
<code>--rsub__(self, other)</code>	Toán tử -	<code>obj2 - obj1</code>
<code>--rmul__(self, other)</code>	Toán tử *	<code>obj2 * obj1</code>
<code>--rtruediv__(self, other)</code>	Toán tử /	<code>obj2 / obj1</code>
<code>--rfloordiv__(self, other)</code>	Toán tử //	<code>obj2 // obj1</code>
<code>--rmod__(self, other)</code>	Toán tử %	<code>obj2 % obj1</code>



Numeric magic methods

Magic methods	Hành vi	Ví dụ
<code>__iadd__(self, other)</code>	Toán tử +=	<code>obj1 += 1</code>
<code>__isub__(self, other)</code>	Toán tử -=	<code>obj1 -= 1</code>
<code>__imul__(self, other)</code>	Toán tử *=	<code>obj1 *= 1</code>
<code>__itruediv__(self, other)</code>	Toán tử /=	<code>obj1 /= 1</code>
<code>__ifloordiv__(self, other)</code>	Toán tử //=	<code>obj1 //= 1</code>
<code>__imod__(self, other)</code>	Toán tử % =	<code>obj1 %= 1</code>



Type conversion magic methods

Magic methods	Hành vi	Ví dụ
<code>__int__(self)</code>	Chuyển thành số nguyên	<code>int(obj)</code>
<code>__float__(self)</code>	Chuyển thành số thực	<code>float(obj)</code>
<code>__bool__(self)</code>	Chuyển thành kiểu bool	<code>bool(obj)</code>
<code>__repr__(self)</code>	Chuyển thành chuỗi	<code>repr(obj)</code>
<code>__str__(self)</code>	Chuyển thành chuỗi	<code>str(obj)</code> , <code>print(obj)</code>



Custom container magic methods

Magic methods	Hành vi	Ví dụ
<code>__len__(self)</code>	Gọi hàm len	<code>len(obj)</code>
<code>__getitem__(self, key)</code>	Truy cập giá trị	<code>obj["name"]</code>
<code>__setitem__(self, key, value)</code>	Đặt giá trị	<code>obj["name"] = "Ba"</code>
<code>__iter__(self)</code>	Sử dụng vòng lặp	<code>for x in obj</code>
<code>__reversed__(self)</code>	Đảo ngược	<code>reversed(obj)</code>