External Text Wrap

If you want to improve and upgrade your classes, chances are that what you need can be done with magic methods. Dunders in Python provide a wide range of functionalities, from type conversion to attribute managing.

In this topic, we've covered dunders that are useful when you want to introduce a little bit of math in your classes. A big advantage of magic methods is that you don't have to define them if you don't need it. But if you do, they're there[1] to make your life easier! Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our identity, not equality, and since they have different id numbers[2]. Two complex numbers are considered equal if and only.

1 Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our program. This is because all numbers are different objects: the code checks them for identity, not equality, and since they have different id numbers, they are not considered the same.

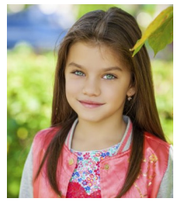
2 In this topic, we've covered dunders that are useful when you want to introduce a little bit of math in your classes. A big advantage of magic methods is that you don't have to define them if you don't need it. But if you do, they're there to make your life easier!

Internal Text Wrap

In this topic, we've covered dunders that are useful when you want to introduce a little bit of math in your classes. A big advantage of magic methods is that you don't have to define them if you don't need it. But if you do, they're there to make your life easier!

In this topic, we've covered dunders that are useful when you want to introduce a little bit of math in your classes. A big advantage of magic methods is that you don't have to define them if you don't need it. But if you do, they're there[1] to make your life easier! Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our identity, not equality, and since they have different id numbers[2] . Two complex numbers

1 Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our program. This is because all numbers are different objects: the code checks them for identity, not equality, and since they have different id numbers, they are not considered the same. Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our program. This is because all numbers are different objects: the code checks them for identity, not equality, and since they have different id numbers, they are not considered the same. Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our program. This is because all numbers are different objects: the code checks them for identity, not equality, and since they have different id numbers, they are not considered the same. Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal.



2 Two complex numbers are considered equal if and only if their real parts and imaginary parts are respectively equal. But, as we can see above, this is not how it worked in our program.