```python
In [1]: class Apple:
            def __init__(self, color, flavor):
                self.color = color
                self.flavor = flavor
            def __str__(self):
                return "This apple is {} and its flavor is {}".format(self.color, self.flavor)
```

```python
In [2]: jonagold = Apple("red", "sweet")
        print(jonagold)
```

```
This apple is red and its flavor is sweet
```

```python
In [3]: help(Apple)
```

```
Help on class Apple in module __main__:

class Apple(builtins.object)
 |  Apple(color, flavor)
 |
 |  Methods defined here:
 |
 |  __init__(self, color, flavor)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  __str__(self)
 |      Return str(self).
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

```python
In [4]: def to_seconds(hours,minutes,seconds):
            """Return the amount of seconds in the given hours, minutes and seconds."""
            return hours*3600+minutes*60+seconds
```

```python
In [5]: help (to_seconds)
```

```
Help on function to_seconds in module __main__:

to_seconds(hours, minutes, seconds)
    Return the amount of seconds in the given hours, minutes and seconds.
```

```python
In [10]: class ClassName:
             def method_name(self, other_parameters):
                 body_of_method
```

```python
In [14]: class Animal:
             sound=""
             def __init__(self, name):
                 self.name=name
             def speak(self):
                 print("{sound} I'm {name}! {sound}".format(name=self.name, sound=self.sound))

         class Cat(Animal):
             sound="Meow!"

         myLuna=Cat("Luna")
         myLuna.speak()
```

```
Meow! I'm Luna! Meow!
```

```python
In [15]: class Animal:
             sound=""
             def __init__(self, name):
                 self.name=name
             def speak(self):
                 print("{sound} I'm {name}! {sound}".format(name=self.name, sound=self.sound))

         class Cat(Animal):
             sound="Meow!"

         myLuna=Cat("Luna")
         myLuna.speak()

         class Cow(Animal):
             sound="Mooo"

         myCow=Cow("Milky")
         myCow.speak()
```

```
Meow! I'm Luna! Meow!
Mooo I'm Milky! Mooo
```

```python
In [16]: class Clothing:
             material=""
             def __init__(self,name):
                 self.name=name
             def checkmaterial(self):
                 print("This {} is made of {}".format(self.name,self.material))

         class Shirt(Clothing):
             material="Cotton"

         polo=Shirt("Polo")
         polo.checkmaterial()
```

```
This Polo is made of Cotton
```

```python
In [17]: import random
         random.randint(1,10)
```

```
Out[17]: 10
```

```python
In [18]: random.randint(1,10)
```

```
Out[18]: 6
```

```python
In [19]: random.randint(1,10)
```

```
Out[19]: 7
```

```python
In [21]:
```

```
  Cell In[21], line 4
    class 'datetime.datetime'
          ^
SyntaxError: invalid syntax
```

```python
In [ ]:
```