```
help("")
```

```
Help on class str in module builtins:

class str(object)
 |  str(object='') -> str
 |  str(bytes_or_buffer[, encoding[, errors]]) -> str
 |
 |  Create a new string object from the given object. If encoding or
 |  errors is specified, then the object must expose a data buffer
 |  that will be decoded using the given encoding and error handler.
 |  Otherwise, returns the result of object.__str__() (if defined)
 |  or repr(object).
 |  encoding defaults to sys.getdefaultencoding().
 |  errors defaults to 'strict'.
 |
 |  Methods defined here:
 |
 |  __add__(self, value, /)
 |      Return self+value.
 |
 |  __contains__(self, key, /)
 |      Return bool(key in self).
 |
 |  __eq__(self, value, /)
 |      Return self==value.
 |
 |  __format__(self, format_spec, /)
 |      Return a formatted version of the string as described by format_spec.
 |
 |  __ge__(self, value, /)
 |      Return self>=value.
 |
 |  __getitem__(self, key, /)
 |      Return self[key].
 |
 |  __getnewargs__(...)
 |
 |  __gt__(self, value, /)
 |      Return self>value.
 |
 |  __hash__(self, /)
 |      Return hash(self).
 |
 |  __iter__(self, /)
 |      Implement iter(self).
 |
 |  __le__(self, value, /)
 |      Return self<=value.
 |
 |  __len__(self, /)
 |      Return len(self).
 |
 |  __lt__(self, value, /)
 |      Return self<value.
 |
 |  __mod__(self, value, /)
 |      Return self%value.
 |
 |  __mul__(self, value, /)
```

```
class Fruit:
  def __init__ (self,color,flavor):
    self.color=color
    self.flavor=flavor

class Apple(Fruit):
  pass
class Banana(Fruit):
  pass

granny= Apple("Green","tart")
bob = Banana("Yellow","Sweet")
print(granny.color)
```

```
Green
```

```
class Piglet:

  def speak(self):
    name="hamlet"
    print("Oink! I'm {} Oink!".format(self.name))
```

```
hamlet= Piglet()
hamlet.name="Hamlet"
hamlet.speak()
```

```
Oink! I'm Hamlet Oink!
```

```
class Fruit:
  def __init__ (self, color, flavor):
    self.color=color
    self.flavor=flavor

class Apple(Fruit):
  pass

class Grape(Fruit):
  pass

granny = Apple("green", "tart")
bob = Grape("purple", "Sweet")

print(granny.color)
print(bob.flavor)
```

```
green
Sweet
```

```
class Piglet:
  def speak(self):
    name = "Hamlet"
    print(f"Oink I'm {name}! Oink!")
hamlet = Piglet()
hamlet.speak()
```

```
Oink I'm Hamlet! Oink!
```