# 二 字符串

练习2-1

```
message = "This is a message"
print(message)
```

```
This is a message
```

练习2-2

```
message = "This is a message"
print(message)
message = "This is a another message"
print(message)
```

```
This is a message
This is a another message
```

---

```
print("Hello World!")
```

```
Hello World!
```

```
name = "ada lovelace"
print(name.title())
```

```
Ada Lovelace
```

```
print(name.upper())
print(name.lower())
```

```
ADA LOVELACE
ada lovelace
```

```
first_name = "ada"
last_name = "lovelace"
full_name = f"{first_name} {last_name}"
print(full_name)
```

```
ada lovelace
```

```
first_name = "ada"
last_name = "lovelace"
full_name = f"{first_name} {last_name}"
print(f"Hello, {full_name.title()}!")
```

```
Hello, Ada Lovelace!
```

```
full_name = "{} {}".format(first_name, last_name)
print(full_name)
```

```
ada lovelace
```

```
print("Python")
print("\tPython")
```

```
Python
    Python
```

```
favourite_language = "Python  "
```

```
favourite_language
```

```
'Python  '
```

```
favourite_language.rstrip()
```

```
'Python'
```

```
favourite_language
```

```
'Python  '
```

```
favourite_language.strip()
```

```
'Python'
```

---

练习2-3

```
message = f"Hello {name}, would you like to learn some Python today?"
name = "TobinMeng"
print(message)
```

```
Hello ada lovelace, would you like to learn some Python today?
```

练习2-4

```
name = "debin meng"
print(name.lower())
print(name.upper())
print(name.title())
```

```
debin meng
DEBIN MENG
Debin Meng
```

练习2-5

```
print('Albert Einstein once said, "A person who never made a mistake never tried
anything new."')
```

```
Albert Einstein once said, "A person who never made a mistake never tried
anything new."
```

练习2-6

```
famous_name = "Albert Einstein"
message = f'{famous_name}  once said, "A person who never made a mistake never
tried anything new."'
print(message)
```

```
Albert Einstein  once said, "A person who never made a mistake never tried
anything new."
```

练习2-7

```
name = "  \t debin meng \n \t "
print(name)
print(name.lstrip())
print(name.rstrip())
print(name.strip())
```

```
     debin meng

debin meng

     debin meng
debin meng
```

练习2-8

```
print(5 + 3)
print(10 - 2)
print(4 * 2)
print(16 / 2)
```

```
8
8
8
8.0
```

练习2-9

```
a = 10
message = f"{a}是我最喜欢的数字"
print(message)
```

```
10是我最喜欢的数字
```

练习2-10

```
#这是一条注释
print("Hello!")
```

```
Hello!
```

```
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
```

```
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

---

# 三 列表

```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles)
```

```
['trek', 'cannondale', 'redline', 'specialized']
```

```
print(bicycles[0])
```

```
trek
```

```
print(bicycles[0].title())
```

```
Trek
```

```
print(bicycles[1])
print(bicycles[3])
```

```
cannondale
specialized
```

```
print(bicycles[-1])
```

```
specialized
```

```
message = f"My first bicycle was a {bicycles[0].title()}"
print(message)
```

```
My first bicycle was a Trek
```

---

练习3-1

```
names = ["Sam", "Amy", "Tom"]
print(names[0])
print(names[1])
print(names[2])
```

```
Sam
Amy
Tom
```

练习3-2

```
message = f"Hello my friend {names[0].title()}"
print(message)
message = f"Hello my friend {names[1].title()}"
print(message)
message = f"Hello my friend {names[2].title()}"
print(message)
```

```
Hello my friend Sam
Hello my friend Amy
Hello my friend Tom
```

练习3-3

```
print("略")
```

```
略
```

---

```
motorcycles = ["honda", "yamaha", "suzuki"]
print(motorcycles)
motorcycles[0] = "ducati"
print(motorcycles)
```

```
['honda', 'yamaha', 'suzuki']
['ducati', 'yamaha', 'suzuki']
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
print(motorcycles)
motorcycles.append('ducati')
print(motorcycles)
```

```
['honda', 'yamaha', 'suzuki']
['honda', 'yamaha', 'suzuki', 'ducati']
```

```
motorcycles = []
motorcycles.append("honda")
motorcycles.append("yamaha")
motorcycles.append("suzuki")
print(motorcycles)
```

```
['honda', 'yamaha', 'suzuki']
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
motorcycles.insert(0, "ducati")
print(motorcycles)
```

```
['ducati', 'honda', 'yamaha', 'suzuki']
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
print(motorcycles)

del motorcycles[0]
print(motorcycles)
```

```
['honda', 'yamaha', 'suzuki']
['yamaha', 'suzuki']
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
print(motorcycles)

popped_motorcycle = motorcycles.pop()
print(motorcycles)
print(popped_motorcycle)
```

```
['honda', 'yamaha', 'suzuki']
['honda', 'yamaha']
suzuki
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
last_owned = motorcycles.pop()

message = f"The last motorcycle I owned was a {last_owned.title()}."
print(message)
```

```
The last motorcycle I owned was a Suzuki.
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
first_motorcycle = motorcycles.pop(0)

message = f"The first motorcycle I owned was a {first_motorcycle.title()}."
print(message)
```

```
The first motorcycle I owned was a Honda.
```

```
motorcycles = ["honda", "yamaha", "suzuki", "ducati"]
print(motorcycles)

motorcycles.remove("ducati")
print(motorcycles)
```

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki']
```

```
motorcycles = ["honda", "yamaha", "suzuki", "ducati"]
print(motorcycles)

too_expensive = "ducati"
motorcycles.remove(too_expensive)
print(motorcycles)
print(f"\nA {too_expensive.title()} is too expensive for me.")
```

```
['honda', 'yamaha', 'suzuki', 'ducati']
['honda', 'yamaha', 'suzuki']

A Ducati is too expensive for me.
```

---

练习3-4

```
names = ["Mao", "Deng", "Zhou"]
message = f"{names[0]}, 来吧! "
print(message)
message = f"{names[1]}, 来吧! "
print(message)
message = f"{names[2]}, 来吧! "
print(message)
```

```
Mao, 来吧!
Deng, 来吧!
Zhou, 来吧!
```

练习3-5

```
names = ["Mao", "Deng", "Zhou"]
aname = "Mao"
print(f"{aname} can not come")
names.remove(aname)
names.append("Liu")
print(f"The new list is {names}")
```

```
Mao can not come
The new list is ['Deng', 'Zhou', 'Liu']
```

练习3-6

```
names = ["Mao", "Deng", "Zhou"]
aname = "Mao"
print(f"{aname} can not come")
names.remove(aname)
names.append("Liu")
print(f"The new list is {names}")
print("\nI find a bigger table now")
names.insert(0, "Wang")
names.insert(2, "Jiang")
names.append("Xi")
print(f"The new list is {names}")
```

```
Mao can not come
The new list is ['Deng', 'Zhou', 'Liu']

I find a bigger table now
The new list is ['Wang', 'Deng', 'Jiang', 'Zhou', 'Liu', 'Xi']
```

练习3-7

```
names = ["Mao", "Deng", "Zhou"]
aname = "Mao"
print(f"{aname} can not come")
names.remove(aname)
names.append("Liu")
print(f"The new list is {names}")
print("\nI find a bigger table now")
```

```
names.insert(0, "Wang")
names.insert(2, "Jiang")
names.append("Xi")
print(f"The new list is {names}")
print("\nThe table can not arrive")

aname = names.pop()
print(f"So sorry {aname}, you can not come.")
aname = names.pop()
print(f"So sorry {aname}, you can not come.")
aname = names.pop()
print(f"So sorry {aname}, you can not come.")
aname = names.pop()
print(f"So sorry {aname}, you can not come.\n")

print(f"{names[0]}, you still can arrive")
print(f"{names[1]}, you still can arrive")

del names[0]
del names[0]
print(f"The new list is {names}")
```

```
Mao can not come
The new list is ['Deng', 'Zhou', 'Liu']

I find a bigger table now
The new list is ['Wang', 'Deng', 'Jiang', 'Zhou', 'Liu', 'Xi']

The table can not arrive
So sorry Xi, you can not come.
So sorry Liu, you can not come.
So sorry Zhou, you can not come.
So sorry Jiang, you can not come.

Wang, you still can arrive
Deng, you still can arrive
The new list is []
```

---

```
cars = ["bwm", "audi", "toyota", "subaru"]
cars.sort()
print(cars)
```

```
['audi', 'bwm', 'subaru', 'toyota']
```

```
cars = ["bwm", "audi", "toyota", "subaru"]
cars.sort(reverse=True)
print(cars)
```

```
['toyota', 'subaru', 'bwm', 'audi']
```

```
cars = ["bwm", "audi", "toyota", "subaru"]

print("Here is the original list:")
print(cars)

print("\nHere is the sorted list:")
print(sorted(cars))

print("\nHere is the original list again:")
print(cars)
```

```
Here is the original list:
['bwm', 'audi', 'toyota', 'subaru']

Here is the sorted list:
['audi', 'bwm', 'subaru', 'toyota']

Here is the original list again:
['bwm', 'audi', 'toyota', 'subaru']
```

```
cars = ["bwm", "audi", "toyota", "subaru"]

print("Here is the original list:")
print(cars)

print("\nHere is the sorted list:")
print(sorted(cars, reverse=True))

print("\nHere is the original list again:")
print(cars)
```

```
Here is the original list:
['bwm', 'audi', 'toyota', 'subaru']

Here is the sorted list:
['toyota', 'subaru', 'bwm', 'audi']

Here is the original list again:
['bwm', 'audi', 'toyota', 'subaru']
```

```
cars = ["bwm", "audi", "toyota", "subaru"]
print(cars)

cars.reverse()
print(cars)
```

```
['bwm', 'audi', 'toyota', 'subaru']
['subaru', 'toyota', 'audi', 'bwm']
```

```
cars = ["bwm", "audi", "toyota", "subaru"]
len(cars)
```

```
4
```

练习3-8

```
places = ["beijign", "shanghai", "shenzhen", "xian", "guilin"]
print(places)
print(sorted(places))
print(places)
print(sorted(places, reverse=True))
print(places)
places.reverse()
print(places)
places.sort()
print(places)
places.sort(reverse=True)
print(places)
```

```
['beijign', 'shanghai', 'shenzhen', 'xian', 'guilin']
['beijign', 'guilin', 'shanghai', 'shenzhen', 'xian']
['beijign', 'shanghai', 'shenzhen', 'xian', 'guilin']
['xian', 'shenzhen', 'shanghai', 'guilin', 'beijign']
['beijign', 'shanghai', 'shenzhen', 'xian', 'guilin']
['guilin', 'xian', 'shenzhen', 'shanghai', 'beijign']
['beijign', 'guilin', 'shanghai', 'shenzhen', 'xian']
['xian', 'shenzhen', 'shanghai', 'guilin', 'beijign']
```

练习3-9

```
names = ["Mao", "Deng", "Zhou"]
aname = "Mao"
print(f"{aname} can not come")
names.remove(aname)
names.append("Liu")
print(f"The new list is {names}")
print("\nI find a bigger table now")
names.insert(0, "Wang")
names.insert(2, "Jiang")
names.append("Xi")
print(f"The new list is {names}")
print(f"The number of list is {len(names)}")
```

```
Mao can not come
The new list is ['Deng', 'Zhou', 'Liu']

I find a bigger table now
The new list is ['Wang', 'Deng', 'Jiang', 'Zhou', 'Liu', 'Xi']
The number of list is 6
```

练习3-10

```
print("略")
```

```
略
```

```
motorcycles = ["honda", "yamaha", "suzuki"]
print(motorcycles[3])
```

```
---------------------------------------------------------------------------

IndexError                                Traceback (most recent call last)

<ipython-input-49-5f506ef75c18> in <module>()
      1 motorcycles = ["honda", "yamaha", "suzuki"]
----> 2 print(motorcycles[3])
```

```
IndexError: list index out of range
```