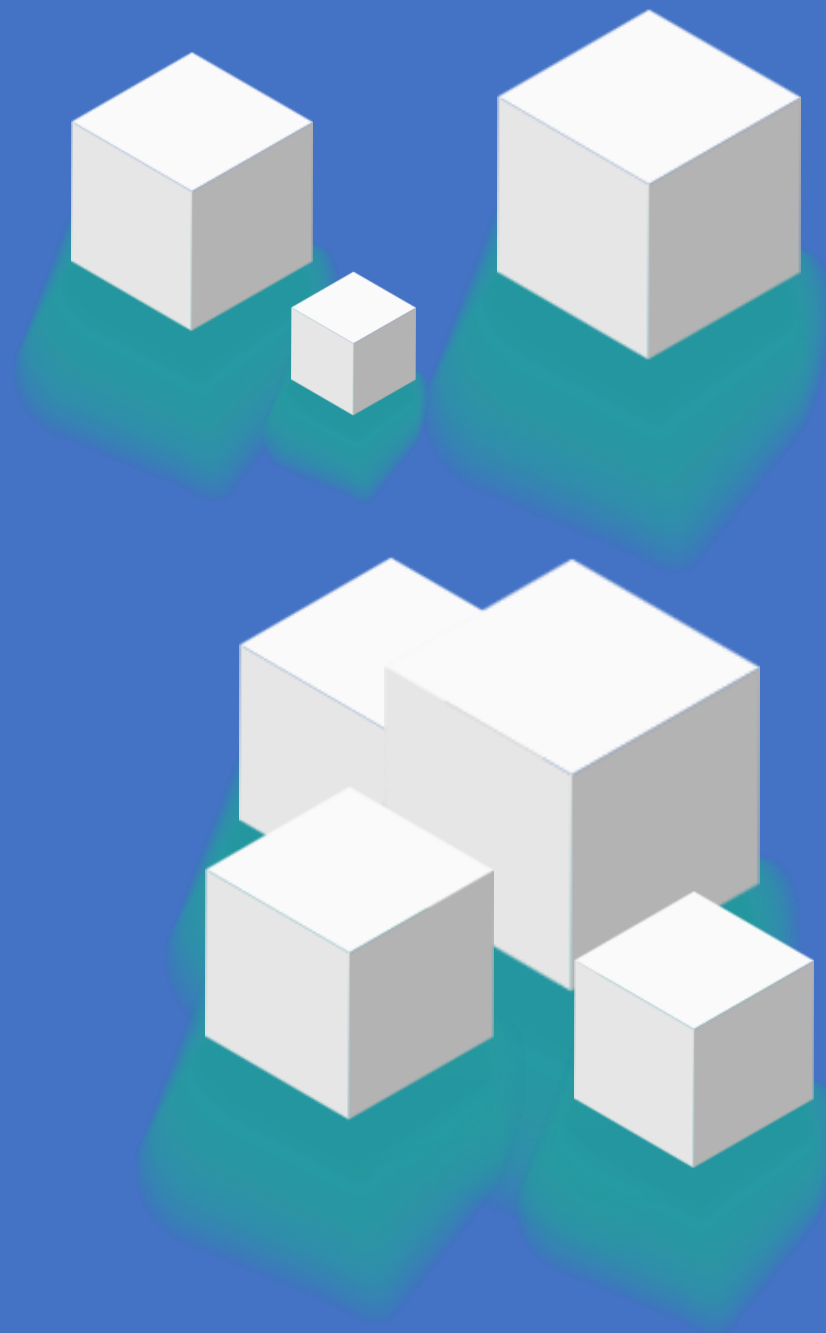
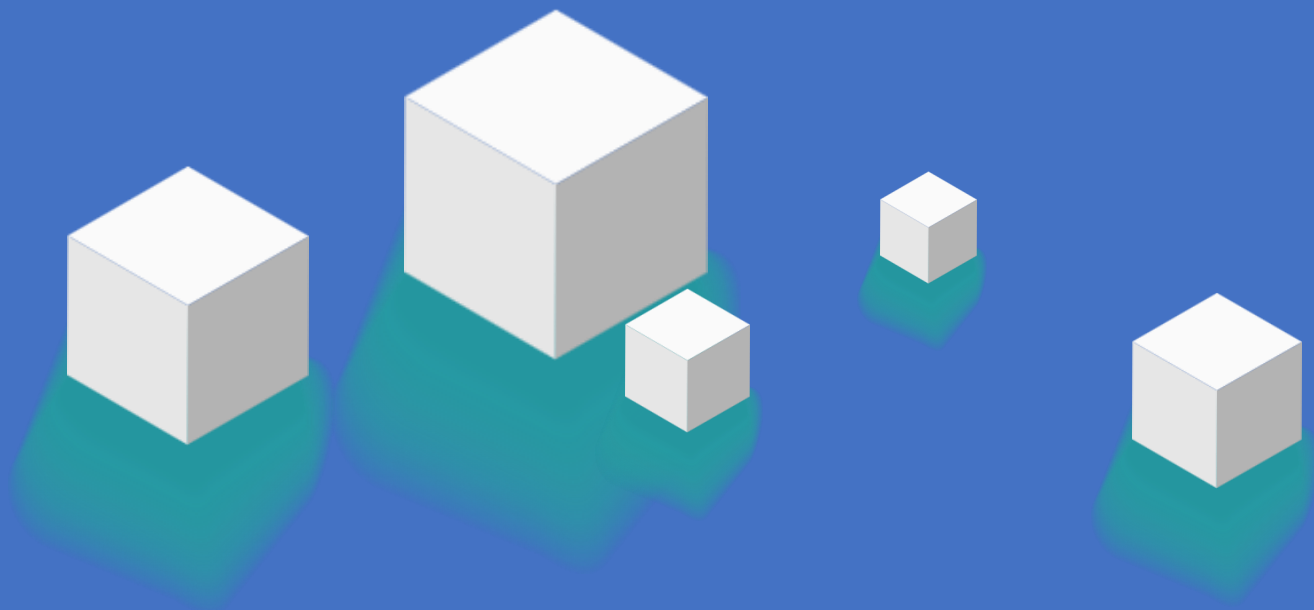


# Python基础语法



# Python基础语法与AI

# Python是数据分析的首选语言

**Thinking:** 如何选择数据分析语言？

- Python是首选的数据分析语言
- 在数据分析/数据科学领域中占有率70%
- 有强大的生态（社区+工具）

科学计算：Sklearn, Numpy, Pandas

人工智能：Tensorflow, PyTorch

网络爬虫：Scrapy, Request, BeautifulSoup

运筹优化：ortools, pulp

Python生态强大，代码简洁

相对其他语言Python更好上手，浙江高考将Python列为可选科目之一

Worldwide, Jun 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	31.6 %	+4.3 %
2		Java	17.67 %	-2.4 %
3		Javascript	8.02 %	-0.2 %
4		C#	6.87 %	-0.4 %
5		PHP	6.02 %	-0.9 %
6		C/C++	5.69 %	-0.2 %
7		R	3.86 %	-0.1 %
8		Objective-C	2.5 %	-0.3 %
9		Swift	2.24 %	-0.1 %
10	↑	TypeScript	1.86 %	+0.2 %

# Python基础语法

---

学习Python可以从以下3个维度掌握

- 基础语法

输入，输出，条件判断，循环语句，注释，引用包，函数定义

- 数据结构

列表、元组、字典、集合

- 常用分析工具

Numpy, Pandas

# Python基础语法

---

- 输入输出

```
name = input("What's your name?")
```

```
sum = 100+100
```

```
print ('hello', name)
```

```
print ('sum', sum)
```

# Python基础语法

---

- 条件判断 if ... else ...

```
score = 95
```

```
if score >= 90:
```

```
    print('Excellent')
```

```
else:
```

```
    if score < 60:
```

```
        print('Fail')
```

```
    else:
```

```
        print('Good Job')
```

# Python基础语法

---

- 循环语句 for ... in

```
sum = 0
```

```
for number in range(11):
```

```
    sum = sum + number
```

```
print(sum)
```

- 循环语句 while

```
sum = 0
```

```
number = 1
```

```
while number < 11:
```

```
    sum = sum + number
```

```
    number = number + 1
```

```
print(sum)
```

# Python基础语法

---

- 注释

```
# -*- coding: utf-8 -*-
```

```
'''
```

这是多行注释，用三个单引号

这是多行注释，用三个单引号

这是多行注释，用三个单引号

```
'''
```

- 引用模块/包: import

# 引用一个或多个包

```
import module_name1,module_name2
```

# 导入包中指定模块

```
from package_name import module_name
```

- 函数定义 def

```
def addone(score):
```

```
    return score + 1
```

```
print(addone(99))
```



# Python数据结构

---

数据类型：列表、元组、字典、集合

- 列表： []

```
lists = ['a','b','c']
```

- 元组 (tuple)

```
tuples = ('tupleA','tupleB')
```

- 字典 {dictionary}

```
score = {'guanyu':95,'zhangfei':96}
```

- 集合： set

```
s = set(['a', 'b', 'c'])
```

# Python数据结构（列表）

---

# 列表

```
lists = ['zhangfei','guanyu','liubei']
```

# 列表中添加元素

```
lists.append('dianwei')
```

```
print(lists)
```

```
print(len(lists))
```

# 在指定位置添加元素

```
lists.insert(0,'diaochan')
```

# 删除末尾元素

```
lists.pop()
```

```
print(lists)
```

# 相关网站和资源

Codewars 是一个在线编程挑战平台，难度适中，适合进行Python基础语法练习

→ ↻ 🔍 codewars.com/kata/53af2b8861023f1d88000832/train/python

Kata Training

Heads Up! Unlock the full potential of your account – confirm your email to enable the full dojo experience!

8 kyu Are You Playing Banjo?

☆ 882 🏆 178 📈 92% of 11,417 🗳️ 72,979 of 191,324 👤 MRodalgaard

🚩 1 Issue Reported

Instructions Output

Create a function which answers the question "Are you playing banjo?".  
If your name starts with the letter "R" or lower case "r", you are playing banjo!

The function takes a name as its only argument, and returns one of the following strings:

```
name + " plays banjo"
name + " does not play banjo"
```

Names given are always valid strings.

STRINGS FUNDAMENTALS

Python 3.11




VIM EMACS

Solution

```
1 def are_you_playing_banjo(name):
2     # Implement me!
3     return name
```

Sample Tests

```
1 import codewars_test as test
2
3 try:
4     from solution import areYouPlayingBanjo as are_you_playing_banjo
5 except ImportError:
6     from solution import are_you_playing_banjo
7
8 @test.describe("Fixed Tests")
9 def basic_tests():
```



Thank You  
Using data to solve problems