

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python？

02 我会怎样带你学 Python？

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码？

第 2 章 通用语言特性

05 数据的名字和种类——变量和类型

06 一串数据怎么存——列表和字符串

07 不只有一条路——分支和循环

08 将代码放进盒子——函数

09 知错能改——错误处理、异常机制

10 定制一个模子——类

11 更大的代码盒子——模块和包

12 练习——密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数 [最近阅读](#)

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编程

15 Python大法初体验：内置函数

更新时间：2019-09-18 10:40:42



“

虚心使人进步，骄傲使人落后。

——毛泽东

”

Python 中内置有很多常用的函数，这些函数无需从模块中导入，可直接使用。由于内置函数有六七十个之多，故这里不一一介绍，只介绍一些最常用的。

数据类型相关

| 内置函数 | 功能 | 示例 | 示例结果 |
|----------------------|-------------------------|----------------------------------|---------------------------------------|
| <code>dict()</code> | 将参数转换为字典类型 | <code>dict(a=1, b=2, c=3)</code> | <code>{'a': 1, 'b': 2, 'c': 3}</code> |
| <code>float()</code> | 将字符串或数字转换为浮点型 | <code>float('0.22')</code> | <code>0.22</code> |
| <code>int()</code> | 将字符串或数字转换为整数型 | <code>int(1.23)</code> | <code>1</code> |
| <code>list()</code> | 将元组、字符串等可迭代对象转换为列表 | <code>list('abc')</code> | <code>['a', 'b', 'c']</code> |
| <code>tuple()</code> | 将列表、字符串等可迭代对象转换为元组 | <code>tuple([1, 2, 3])</code> | <code>(1, 2, 3)</code> |
| <code>set()</code> | 1.创建空集合；2.将可迭代对象转换为列表集合 | <code>set('abc')</code> | <code>{'b', 'a', 'c'}</code> |
| <code>str()</code> | 将参数转换为字符串 | <code>str(3.14)</code> | <code>'3.14'</code> |
| <code>bytes()</code> | 将参数转换为字节序列 | <code>bytes(4)</code> | <code>b'\x00\x00\x00\x00'</code> |

| | |
|--|--|
| 目录 | 似。 |
| 第 1 章 入门准备 | |
| 01 开篇词：你为什么要学 Python？ | 可迭代对象：如列表、元组、字符串、集合、字典等。关于可迭代对象的细节见下一章节。 |
| 02 我会怎样带你学 Python？ | |
| 03 让 Python 在你的电脑上安家落户 | |
| 04 如何运行 Python 代码？ | |
| 第 2 章 通用语言特性 | 数值计算相关 |
| 05 数据的名字和种类—变量和类型 | |
| 06 一串数据怎么存—列表和字符串 | |
| 07 不只有一条路—分支和循环 | |
| 08 将代码放进盒子—函数 | |
| 09 知错能改—错误处理、异常机制 | |
| 10 定制一个模子—类 | |
| 11 更大的代码盒子—模块和包 | |
| 12 练习—密码生成器 | |
| 第 3 章 Python 进阶语言特性 | |
| 13 这么多的数据结构（一）：列表、元祖、字符串 | |
| 14 这么多的数据结构（二）：字典、集合 | |
| 15 Python大法初体验：内置函数 最近阅读 | |
| 16 深入理解下迭代器和生成器 | |
| 17 生成器表达式和列表生成式 | |
| 18 把盒子升级为豪宅：函数进阶 | |
| 19 让你的模子更好用：类进阶 | |
| 20 从小独栋升级为别墅区：函数式编程 | |

| | |
|--|---|
| 目录 | |
| 第 1 章 入门准备 | |
| 01 开篇词：你为什么要学 Python ？ | True >>> bool(0) False >>> bool([1, 2, 3]) True >>> bool([]) False >>> bool('abc') True >>> bool(' ') False |
| 02 我会怎样带你学 Python ？ | |
| 03 让 Python 在你的电脑上安家落户 | |
| 04 如何运行 Python 代码 ？ | |
| 第 2 章 通用语言特性 | |
| 05 数据的名字和种类—变量和类型 | >>> all(['a' , 1, [1]]) True >>> all(['a' , 0, [1]]) False |
| 06 一串数据怎么存—列表和字符串 | |
| 07 不只有一条路—分支和循环 | |
| 08 将代码放进盒子—函数 | |
| 09 知错能改—错误处理、异常机制 | >>> any([' ' , 0, []]) False >>> any(['a' , 0, []]) True |
| 10 定制一个模子—类 | |
| 11 更大的代码盒子—模块和包 | |
| 12 练习—密码生成器 | |
| 第 3 章 Python 进阶语言特性 | |
| 13 这么多的数据结构（一）：列表、元祖、字符串 | |
| 14 这么多的数据结构（二）：字典、集合 | |
| 15 Python大法初体验：内置函数 最近阅读 | |
| 16 深入理解下迭代器和生成器 | |
| 17 生成器表达式和列表生成式 | |
| 18 把盒子升级为豪宅：函数进阶 | |
| 19 让你的模子更好用：类进阶 | |
| 20 从小独栋升级为别墅区：函数式编程 | |

IO 相关

IO 即输入输出。

| 内置函数 | 功能 |
|----------------------|---------------------------------|
| <code>input()</code> | 从标准输入中读取字符串 |
| <code>print()</code> | 将内容写入标准输出中 |
| <code>open()</code> | 打开一个文件。之后便可以对文件做读写操作。详见 IO 操作章节 |

元数据相关

| 内置函数 | 功能 |
|---------------------------|--|
| <code>type()</code> | 获取对象的类型 |
| <code>isinstance()</code> | 判断对象是否是某个类（或其子类）的对象 |
| <code>dir()</code> | 获取类或对象中的所有方法和属性；无参数时获取当前作用域下的所有名字 |
| <code>id()</code> | 返回一个对象的唯一标识。在我们所使用的 CPython 中这个唯一标识实际为该对象在内存中的地址 |

| | |
|---|--|
| <div><div>← 慕课专栏</div><div>☰ 你的第一本Python基础入门书 / 15 Python大法初体验：内置函数</div></div> | |
| 目录 | |
| 第 1 章 入门准备 | |
| 01 开篇词：你为什么要学 Python ？ | <pre>>>> numbers = [1, 2, 3] >>> type(numbers) <class 'list' ></pre> |
| 02 我会怎样带你学 Python ？ | <p>isinstance() 示例：</p> |
| 03 让 Python 在你的电脑上安家落户 | <pre>>>> numbers = [1, 2, 3] >>> isinstance(numbers, list) True >>> isinstance(numbers, str) False</pre> |
| 04 如何运行 Python 代码？ | <p>也可以把多个类型放在元组中，其中一个与对象的类型相符即为 True，若无相符则为 False。</p> <p>如：</p> |
| 第 2 章 通用语言特性 | |
| 05 数据的名字和种类—变量和类型 | <pre>>>> numbers = [1, 2, 3] >>> isinstance(numbers, (list, str)) True</pre> |
| 06 一串数据怎么存—列表和字符串 | |
| 07 不只有一条路—分支和循环 | |
| 08 将代码放进盒子—函数 | |
| 09 知错能改—错误处理、异常机制 | |
| 10 定制一个模子—类 | |
| 11 更大的代码盒子—模块和包 | |
| 12 练习—密码生成器 | |
| 第 3 章 Python 进阶语言特性 | |
| 13 这么多的数据结构（一）：列表、元祖、字符串 | <pre>>>> dir(list) ['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']</pre> |
| 14 这么多的数据结构（二）：字典、集合 | |
| 15 Python大法初体验：内置函数 最近阅读 | |
| 16 深入理解下迭代器和生成器 | <p>id() 示例：</p> |
| 17 生成器表达式和列表生成式 | <pre>>>> number = 1 >>> id(number) 4411695232</pre> |
| 18 把盒子升级为豪宅：函数进阶 | |
| 19 让你的模子更好用：类进阶 | |
| 20 从小独栋升级为别墅区：函数式编程 | |

| | |
|---|--|
| <div>← 慕课专栏</div> | <div>≡ 你的第一本Python基础入门书 / 15 Python大法初体验：内置函数</div> |
| <div>目录</div> | <div>4417622792</div> |
| <div>第 1 章 入门准备</div> | <div>help()</div> |
| <div>01 开篇词：你为什么要学 Python？</div> | <div>解释器交互模式下获取某个函数、类的帮助信息，非常实用。</div> |
| <div>02 我会怎样带你学 Python？</div> | <div>比如查看内置函数 <code>any()</code> 的用法：</div> |
| <div>03 让 Python 在你的电脑上安家落户</div> | <div></div> |
| <div>04 如何运行 Python 代码？</div> | <div>>>> <code>help(any)</code> # 只需使用函数名字</div> |
| <div>第 2 章 通用语言特性</div> | <div>将显示出 <code>any()</code> 的帮助信息：</div> |
| <div>05 数据的名字和种类—变量和类型</div> | <div>Help on built-in function any in module builtins:</div> |
| <div>06 一串数据怎么存—列表和字符串</div> | <div><code>any(iterable, /)</code> Return True if bool(x) is True for any x in the iterable.</div> |
| <div>07 不只有一条路—分支和循环</div> | <div>If the iterable is empty, return False.</div> |
| <div>08 将代码放进盒子—函数</div> | <div>(END)</div> |
| <div>09 知错能改—错误处理、异常机制</div> | <div>按下 <code>q</code> 键退出上述界面。</div> |
| <div>10 定制一个模子—类</div> | <div></div> |
| <div>11 更大的代码盒子—模块和包</div> | <div>对于这个章节中的内置函数，如果你有不清楚的地方，便可以用 <code>help()</code> 来查看使用说明。</div> |
| <div>12 练习—密码生成器</div> | <div>sorted()</div> |
| <div>第 3 章 Python 进阶语言特性</div> | <div>对可迭代对象中的数据进行排序，返回一个新的列表。</div> |
| <div>13 这么多的数据结构（一）：列表、元祖、字符串</div> | <div>>>> <code>numbers = (4, 5, 2, 8, 9, 1, 0)</code> >>> <code>sorted(numbers)</code> [0, 1, 2, 4, 5, 8, 9]</div> |
| <div>14 这么多的数据结构（二）：字典、集合</div> | <div></div> |
| <div>15 Python大法初体验：内置函数 最近阅读</div> | <div>通过参数 <code>reverse=True</code> 指定倒序：</div> |
| <div>16 深入理解下迭代器和生成器</div> | <div>>>> <code>numbers = (4, 5, 2, 8, 9, 1, 0)</code> >>> <code>sorted(numbers, reverse=True)</code> [9, 8, 5, 4, 2, 1, 0]</div> |
| <div>17 生成器表达式和列表生成式</div> | <div></div> |
| <div>18 把盒子升级为豪宅：函数进阶</div> | <div>通过参数 <code>key</code> 指定排序时所使用的字段：</div> |
| <div>19 让你的模子更好用：类进阶</div> | <div></div> |
| <div>20 从小独栋升级为别墅区：函数式编程</div> | <div></div> |

| | |
|---|---|
| <div>← 慕课专栏</div> <div>☰ 你的第一本Python基础入门书 / 15 Python大法初体验：内置函数</div> | |
| 目录 | |
| 第 1 章 入门准备 | |
| 01 开篇词：你为什么要学 Python ？ | |
| 02 我会怎样带你学 Python ？ | |
| 03 让 Python 在你的电脑上安家落户 | |
| 04 如何运行 Python 代码 ？ | |
| 第 2 章 通用语言特性 | |
| 05 数据的名字和种类—变量和类型 | |
| 06 一串数据怎么存—列表和字符串 | |
| 07 不只有一条路—分支和循环 | |
| 08 将代码放进盒子—函数 | |
| 09 知错能改—错误处理、异常机制 | |
| 10 定制一个模子—类 | |
| 11 更大的代码盒子—模块和包 | |
| 12 练习—密码生成器 | |
| 第 3 章 Python 进阶语言特性 | |
| 13 这么多的数据结构（一）：列表、元祖、字符串 | |
| 14 这么多的数据结构（二）：字典、集合 | |
| 15 Python大法初体验：内置函数 最近阅读 | <pre>州', '020']] >>> sorted(codes, key=lambda x: x[1]) [(' 北京 ', '010'), (' 广州 ', '020'), (' 上海 ', '021'), (' 成都 ', '028')]</pre> |
| 16 深入理解下迭代器和生成器 | <p>说明：指定 key 排序需要用到 lambda 表达式。有关 lambda 表达式的内容将在函数式编程章节中介绍。</p> |
| 17 生成器表达式和列表生成式 | <h3>range()</h3> <p>获取一个整数序列。可指定起始数值，结束数值，增长步长。</p> <p>在 for 循环中想要指定循环次数时非常有用。</p> <ul style="list-style-type: none">指定起始数值和结束数值，获取一个连续的整数序列 <pre>for i in range(2, 6): print(i)</pre> |
| 18 把盒子升级为豪宅：函数进阶 | <pre>>>> for i in range(2, 6): ... print(i) ... 2 3 4 5</pre> |
| 19 让你的模子更好用：类进阶 | <p>注意，生成的数值范围为左开右闭区间，即不包括所指定的结束数值。</p> <ul style="list-style-type: none">只指定结束数值，此时起始数值默认为 0 |
| 20 从小独栋升级为别墅区：函数式编程 | <pre>>>> for i in range(4): ... print(i) ... 0 1 2 3</pre> <ul style="list-style-type: none">指定步长（第三个参数） |

← 慕课专栏

你的第一本Python基础入门书 / 15 Python大法初体验：内置函数

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python ？

02 我会怎样带你学 Python ？

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码 ？

第 2 章 通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改—错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数 最近阅读

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编程

… print(i)

…

3

6

9

12

扩展：`range()` 返回的并不是容器，而是可迭代对象，关于可迭代对象的细节见下一章节

← 14 这么多的数据结构（二）：字典、集合

16 深入理解下迭代器和生成器 →

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

！

目前暂无任何讨论

千学不如一看，千看不如一练

www.imooc.com/read/46/article/824

7/7