

## PYTHON教程

### 1. 简介

### 2. Python历史

### 3. 安装Python

#### 3.1. Python解释器

### 4. 第一个Python程序

#### 4.1. 使用文本编辑器

#### 4.2. 输入和输出

### 5. Python基础

#### 5.1. 数据类型和变量

#### 5.2. 字符串和编码

#### 5.3. 使用list和tuple

#### 5.4. 条件判断

#### 5.5. 模式匹配

#### 5.6. 循环

#### 5.7. 使用dict和set

### 6. 函数

#### 6.1. 调用函数

#### 6.2. 定义函数

#### 6.3. 函数的参数

#### 6.4. 递归函数

### 7. 高级特性

#### 7.1. 切片

#### 7.2. 迭代

#### 7.3. 列表生成式

#### 7.4. 生成器

#### 7.5. 迭代器

### 8. 函数式编程

### 9. 模块

### 10. 面向对象编程

### 11. 面向对象高级编程

### 12. 错误、调试和测试

### 13. IO编程

### 14. 进程和线程

### 15. 正则表达式

### 16. 常用内建模块

### 17. 常用第三方模块

### 18. 图形界面

### 19. 网络编程

### 20. 电子邮件

### 21. 访问数据库

### 22. Web开发

### 23. 异步IO

### 24. FAQ

### 25. 期末总结

# 调用函数



廖雪峰

资深软件开发工程师，业余马拉松选手。

Python内置了很多有用的函数，我们可以直接调用。

要调用一个函数，需要知道函数的名称和参数，比如求绝对值的函数 `abs`，只有一个参数。可以直接从Python的官方网站查看[文档](#)，也可以在交互式命令行通过 `help(abs)` 查看 `abs` 函数的帮助信息。

调用 `abs` 函数：

```
>>> abs(100)
100
>>> abs(-20)
20
>>> abs(12.34)
12.34
```

调用函数的时候，如果传入的参数数量不对，会报 `TypeError` 的错误，并且Python会明确地告诉你：`abs()` 有且仅有1个参数，但给出了两个：

```
>>> abs(1, 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: abs() takes exactly one argument (2 given)
```

如果传入的参数数量是对的，但参数类型不能被函数所接受，也会报 `TypeError` 的错误，并且给出错误信息：`str` 是错误的参数类型：

```
>>> abs('a')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: bad operand type for abs(): 'str'
```

而 `max` 函数 `max()` 可以接收任意多个参数，并返回最大的那个：

```
>>> max(1, 2)
2
>>> max(2, 3, 1, -5)
3
```

## 数据类型转换

Python内置的常用函数还包括数据类型转换函数，比如 `int()` 函数可以把其他数据类型转换为整数：

```
>>> int('123')
123
>>> int(12.34)
12
>>> float('12.34')
12.34
>>> str(1.23)
'1.23'
>>> str(100)
'100'
>>> bool(1)
True
>>> bool('')
False
```

函数名其实就是指向一个函数对象的引用，完全可以把函数名赋给一个变量，相当于给这个函数起了一个“别名”：

```
>>> a = abs # 变量a指向abs函数
>>> a(-1) # 所以也可以通过a调用abs函数
```

## PYTHON教程

- 1. 简介
- 2. Python历史
- 3. 安装Python
  - 3.1. Python解释器
- 4. 第一个Python程序
  - 4.1. 使用文本编辑器
  - 4.2. 输入和输出
- 5. Python基础
  - 5.1. 数据类型和变量
  - 5.2. 字符串和编码
  - 5.3. 使用list和tuple
  - 5.4. 条件判断
  - 5.5. 模式匹配
  - 5.6. 循环
  - 5.7. 使用dict和set
- 6. 函数
  - 6.1. 调用函数
  - 6.2. 定义函数
  - 6.3. 函数的参数
  - 6.4. 递归函数
- 7. 高级特性
  - 7.1. 切片
  - 7.2. 迭代
  - 7.3. 列表生成式
  - 7.4. 生成器
  - 7.5. 迭代器
- 8. 函数式编程
- 9. 模块
- 10. 面向对象编程
- 11. 面向对象高级编程
- 12. 错误、调试和测试
- 13. IO编程
- 14. 进程和线程
- 15. 正则表达式
- 16. 常用内建模块
- 17. 常用第三方模块
- 18. 图形界面
- 19. 网络编程
- 20. 电子邮件
- 21. 访问数据库
- 22. Web开发
- 23. 异步IO
- 24. FAQ
- 25. 期末总结

## 练习

请利用Python内置的 `hex()` 函数把一个整数转换成十六进制表示的字符串：

```
n1 = 255
n2 = 1000

print(???)
```



## 参考源码

call\_func.py

## 小结

调用Python的函数，需要根据函数定义，传入正确的参数。如果函数调用出错，一定要学会看错误信息，所以英文很重要！

« 函数

定义函数 »



## Comments

Comments loaded. To post a comment, please [Sign In](#)



仰望星空 @ 2025/12/31 07:23:06

```
n1 = 255 n2 = 1000 print(hex(n1)) print(hex(n2))
```



庶民的胜利 @ 2025/10/3 03:34:30

```
a=int(input("请输入需要转化的数字")) b=hex(a) print(f'{a}的16进制转换为{b}')
```



凉雨 @ 2025/11/26 20:16:17

赞



陈年老酒, 越陈越醇! @ 2025/12/28 21:46:35

牛



杰 @ 2025/12/14 23:53:03

直接在python IDE上写就可以了。

```
n1 = 255
n2 = 1000
h1 = hex(n1)
h2 = hex(n2)
```

## PYTHON教程

### 1. 简介

### 2. Python历史

### 3. 安装Python

#### 3.1. Python解释器

### 4. 第一个Python程序

#### 4.1. 使用文本编辑器

#### 4.2. 输入和输出

### 5. Python基础

#### 5.1. 数据类型和变量

#### 5.2. 字符串和编码

#### 5.3. 使用list和tuple

#### 5.4. 条件判断

#### 5.5. 模式匹配

#### 5.6. 循环

#### 5.7. 使用dict和set

### 6. 函数

#### 6.1. 调用函数

#### 6.2. 定义函数

#### 6.3. 函数的参数

#### 6.4. 递归函数

### 7. 高级特性

#### 7.1. 切片

#### 7.2. 迭代

#### 7.3. 列表生成式

#### 7.4. 生成器

#### 7.5. 迭代器

### 8. 函数式编程

### 9. 模块

### 10. 面向对象编程

### 11. 面向对象高级编程

### 12. 错误、调试和测试

### 13. IO编程

### 14. 进程和线程

### 15. 正则表达式

### 16. 常用内建模块

### 17. 常用第三方模块

### 18. 图形界面

### 19. 网络编程

### 20. 电子邮件

### 21. 访问数据库

### 22. Web开发

### 23. 异步IO

### 24. FAQ

### 25. 期末总结

```
print(f"整数n1={n1}, 整数n2={n2}")
print(""*45)
print(f"小写无前缀显示: n1={n1:x}, n2={n2:x}")
print(f"小写带前缀显示: n1={n1:#x}, n2={n2:#x}")
print(f"大写带前缀显示: n1={n1:#X}, n2={n2:#X}")
print(f"转换成十六进制后显示: hex(n1)={h1}, hex(n2)={h2}")
```



卿缘 @ 2025/12/10 01:29:59

延伸一下, 采用input形式拿到数据的话, 就要对数据可转化类型做判断, 浮点数, 整数和其他

```
input_str=input('请输入您要转换为十六进制的数字: ')
# 先尝试转化为整数, 使用try进行判断, 不满足整数的话进行下一步
try:
    num=int(input_str)
    print(f'您输入的数字{num}的十六进制格式为:{hex(num)}')
except ValueError:
    # 尝试转化为浮点数
    try:
        num=float(input_str)
        print(f'您输入数字{num}的十六进制格式为:{num.hex()}')
    # 既不是整数也不是浮点数则提示重新输入
except ValueError:
    print('您输入的数字类型有误, 请重新输入')
```



Rynhow @ 2025/12/8 06:46:48

```
n1 = 255 n2 = 1000 num1 = hex(n1) num2 = hex(n2) print(f'第一个数:{num1},第二个数{num2}')
```



啊, 你以为我傻 @ 2025/11/29 02:55:54

```
n1 = 255 n2 = 1000 print(str(hex(n1))) print(hex(n2))
```



lv @ 2025/11/25 06:19:28

```
a=hex n1 = 255 print(f'print n1:{n1} to hexadecimal:',a(n1)) n2 = 1000 print(f'print n2:{n2} to hexadecimal:',a(n2))
```

```
n1 = 255 print('print n1:%d to hexadecimal:' % n1,a(n1)) n2 = 1000 print('print n2:%d to hexadecimal:' % n2,a(n2))
```

```
#type function,返回变量的类型, 可以看出函数也是一种类型 print(type(a)) print(type(max))
```



Hypersomnia @ 2025/11/23 22:18:25

```
n1 = 255
n2 = 1000
print(hex(n1))
print(hex(n2))
```



Meow @ 2025/11/13 01:35:14

```
def ten_2_sixteen(n):
    print(f'{n} 的十六进制为 {hex(n)}')

n1 = 255
n2 = 1000

ten_2_sixteen(n1)
ten_2_sixteen(n2)
```



梦缺卿 @ 2025/10/15 06:48:27

```
n1=255
n2=1000
```

## PYTHON教程

### 1. 简介

### 2. Python历史

### 3. 安装Python

#### 3.1. Python解释器

### 4. 第一个Python程序

#### 4.1. 使用文本编辑器

#### 4.2. 输入和输出

### 5. Python基础

#### 5.1. 数据类型和变量

#### 5.2. 字符串和编码

#### 5.3. 使用list和tuple

#### 5.4. 条件判断

#### 5.5. 模式匹配

#### 5.6. 循环

#### 5.7. 使用dict和set

### 6. 函数

#### 6.1. 调用函数

#### 6.2. 定义函数

#### 6.3. 函数的参数

#### 6.4. 递归函数

### 7. 高级特性

#### 7.1. 切片

#### 7.2. 迭代

#### 7.3. 列表生成式

#### 7.4. 生成器

#### 7.5. 迭代器

### 8. 函数式编程

### 9. 模块

### 10. 面向对象编程

### 11. 面向对象高级编程

### 12. 错误、调试和测试

### 13. IO编程

### 14. 进程和线程

### 15. 正则表达式

### 16. 常用内建模块

### 17. 常用第三方模块

### 18. 图形界面

### 19. 网络编程

### 20. 电子邮件

### 21. 访问数据库

### 22. Web开发

### 23. 异步IO

### 24. FAQ

### 25. 期末总结

```
print('%d的十六进制:%s,%d的十六进制:%s' % (n1,hex(n1),n2,hex(n2)))
```



Y² @ 2025/11/6 03:21:34

赞



Y² @ 2025/11/6 03:19:46

```
n1=255 n2=1000 print(str(hex(n1))) print(str(hex(n2)))
```



待绝笔墨痕干 @ 2025/10/30 22:24:51

#请利用Python内置的hex()函数把一个整数转换成十六进制表示的字符串: n1 = 255 n2 = 1000

```
print(f'{n1}转化成16进制的值为{hex(n1)}') print(f'{n2}转化成16进制的值为{hex(n2)}')
```



Bacium @ 2025/10/24 22:12:59

# 请利用Python内置的hex()函数把一个整数转换成十六进制表示的字符串:

```
n1 = 255
```

```
n2 = 1000
```

```
def numForamt(num):  
    print(f"{num}转换16进制后:{hex(num)}")
```

```
numForamt(n1)
```

```
numForamt(n2)
```



Sfz @ 2025/10/24 04:12:55

```
n1 = 255
```

```
n2 = 1000
```

```
print(hex(n1),hex(n2))
```



z. @ 2025/10/20 11:23:47

```
n=255 n2=1000 print(f'n的16进制{hex(n)[2:]}\\nn2的16进制{hex(n2)[2:]}')
```



长期素食 @ 2025/10/14 10:05:44

```
n1 = 255
```

```
n2 = 1000
```

```
print("Hex(255) = {0}".format(hex(n1)))
```

```
print("Hex(1000) = {0}".format(hex(n2)))
```



318197375 @ 2025/10/13 05:00:03

```
>>> print(type(max))  
<class 'builtin_function_or_method'  
>>> print(type(m))  
<class 'builtin_function_or_method'  
这里表明，函数在python中也是一个数据类型
```



AADM @ 2025/10/10 00:47:39

在交互式命令行中 help(abs)查看abs函数的帮助信息。

## PYTHON教程

### 1. 简介

### 2. Python历史

### 3. 安装Python

#### 3.1. Python解释器

### 4. 第一个Python程序

#### 4.1. 使用文本编辑器

#### 4.2. 输入和输出

### 5. Python基础

#### 5.1. 数据类型和变量

#### 5.2. 字符串和编码

#### 5.3. 使用list和tuple

#### 5.4. 条件判断

#### 5.5. 模式匹配

#### 5.6. 循环

#### 5.7. 使用dict和set

### 6. 函数

#### 6.1. 调用函数

#### 6.2. 定义函数

#### 6.3. 函数的参数

#### 6.4. 递归函数

### 7. 高级特性

#### 7.1. 切片

#### 7.2. 迭代

#### 7.3. 列表生成式

#### 7.4. 生成器

#### 7.5. 迭代器

### 8. 函数式编程

### 9. 模块

### 10. 面向对象编程

### 11. 面向对象高级编程

### 12. 错误、调试和测试

### 13. IO编程

### 14. 进程和线程

### 15. 正则表达式

### 16. 常用内建模块

### 17. 常用第三方模块

### 18. 图形界面

### 19. 网络编程

### 20. 电子邮件

### 21. 访问数据库

### 22. Web开发

### 23. 异步IO

### 24. FAQ

### 25. 期末总结



Gmarket @ 2025/10/9 10:49:26

```
n1 = 255 n2 = 1000 print(hex(n1),hex(n2))
```



學不懂Fourier @ 2025/10/2 04:45:38

```
n1 = 255 n2 = 1000 a1=hex(n1) a2=hex(n2) print(a1,a2)
```

©liaoxuefeng.com - 微博 - GitHub - License