



Introdução à linguagem Python

Estrutura de decisão

```
5 if(condiçãoPrimaria):  
6     print("Condição verdadeira")  
7 elif(condiçãoSecundaria):  
8     print("Condição secundaria verdadeira")  
9 else:  
10    print("Caso contrario")
```

```
17 temperatura = 26
18
19 if(temperatura < 10):
20     print("Frio")
21 elif(temperatura > 25):
22     print("Calor")
23 else:
24     print("Normal")
```

```
13 for numero in range(1, 10):  
14     print(numero)  
15
```

```
12 for letra in "ABCD":  
13     print(letra)  
14
```

Usando o laço de repetição faça um programa que mostre a tabuada do 7

```
26 for numero in range(1, 11):  
27     print(7, "x" ,numero, " = ", numero*7)  
28
```

```
16 def minhaFuncao(parametro):  
17     print(parametro)  
18  
19 minhaFuncao("Esse e o parametro da função")  
20
```

Faça uma função que recebe um numero como parâmetro e mostra a tabuada do numero recebido

```
13 def minhaFuncao(parametro):
14     for numero in range(1, 11):
15         print(parametro, "x", numero, "=", parametro * numero)
16
17 minhaFuncao(2)
18
```

Faça uma função que recebe uma idade como parâmetro e mostra se é maior de idade ou não

```
23 def funcaoMaior(idade):  
24     if(idade >= 18):  
25         print("Maior de idade")  
26     else:  
27         print("Menor de idade")  
28  
29 funcaoMaior(20)
```


Faça uma função que recebe uma idade como parâmetro e mostra se é maior de idade ou não

```
1 arquivo = open("./meu_arquivo.txt", "w")
2
3 arquivo.write("Meu primeiro arquivo criado com python")
4 arquivo.close()
5
```

```
6 arquivo = open("./meu_arquivo.txt", "r")
7 conteudo = arquivo.read()
8 print(conteudo)
```

Faça um programa que crie um arquivo e grave uma sequência de nucleotídeos nesse arquivo

<http://dontpad.com/bioevol/2019/seq>

```
1 arquivo = open("./arquivoteste.txt", "w")
2
3 #http://dontpad.com/bioevol/2019/seq
4 sequencia = ["C","G","G","C","T","T","T","G","G","T","G","A",
5             "C","T","C","T","A","G","A","T","A","A","C","C","T","C","G",
6             "G","G","C","C","G","A","T","C","G","C","A","C","G","C","C",
7             "C","C","C","C","G","T","G","G","C","G","G","C","G","A","C",
8             "G","A","C","C","C","A","T","T","C","G","A","A","C"]
9
10 for letra in sequencia:
11     arquivo.write(letra)
12
13 arquivo.close()
14
```

Faça um programa que crie um arquivo e grave uma sequência de nucleotídeos nesse arquivo substituindo o que não for "T" e "C" por "-"

<http://donthpad.com/bioevol/2019/seq>

```
1 arquivo = open("./arquivoteste.txt", "w")
2
3 #http://donthpad.com/bioevol/2019/seq
4 sequencia = ["C","G","G","C","T","T","T","G","G","T","G","A","C",
5
6 for letra in sequencia:
7     if(letra == "C"):
8         arquivo.write(letra)
9     elif(letra == "T"):
10        arquivo.write(letra)
11    else:
12        arquivo.write("-")
13
14 arquivo.close()
15
```

```
1 import math
2
3 raiz = math.sqrt(144)
4 print(raiz)
5
```

math — Mathematical functions x +

Python Software Foundation [US] | <https://docs.python.org/3/library/math.html>

Apps G f N t v o e o s n news Mestrado React Native

Python » English » 3.7.4rc1 » Documentation » The Python Standard Library » Numeric and Mathematical Modules »

math — Mathematical functions

This module provides access to the mathematical functions defined by the C standard.

These functions cannot be used with complex numbers; use the functions of the same name from the `cmath` module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

Number-theoretic and representation functions

math.ceil(x)
Return the ceiling of *x*, the smallest integer greater than or equal to *x*. If *x* is not a float, delegates to *x*.`__ceil__()`, which should return an *Integral* value.

math.copysign(x, y)
Return a float with the magnitude (absolute value) of *x* but the sign of *y*. On platforms that support signed zeros, `copysign(1.0, -0.0)` returns `-1.0`.

math.fabs(x)
Return the absolute value of *x*.

math.factorial(x)
Return *x* factorial as an integer. Raises *ValueError* if *x* is not integral or is negative.

math.floor(x)
Return the floor of *x*, the largest integer less than or equal to *x*. If *x* is not a float, delegates to *x*.`__floor__()`, which should return an *Integral* value.

Table of Contents

- math — Mathematical functions
 - Number-theoretic and representation functions
 - Power and logarithmic functions
 - Trigonometric functions
 - Angular conversion
 - Hyperbolic functions
 - Special functions
 - Constants

Previous topic
[numbers — Numeric abstract base classes](#)

Next topic
[cmath — Mathematical functions for complex numbers](#)

This Page
[Report a Bug](#)
[Show Source](#)

math — Mathematical functions — x

Python Software Foundation [US] | <https://docs.python.org/3/library/math.html>

Apps G f N t v e o s u g v p x news Mestrado React Native

Return the square root of x .

Trigonometric functions

`math.acos(x)`
Return the arc cosine of x , in radians.

`math.asin(x)`
Return the arc sine of x , in radians.

`math.atan(x)`
Return the arc tangent of x , in radians.

`math.atan2(y, x)`
Return `atan(y / x)`, in radians. The result is between `-pi` and `pi`. The vector in the plane from the origin to point `(x, y)` makes this angle with the positive X axis. The point of `atan2()` is that the signs of both inputs are known to it, so it can compute the correct quadrant for the angle. For example, `atan(1)` and `atan2(1, 1)` are both `pi/4`, but `atan2(-1, -1)` is `-3*pi/4`.

`math.cos(x)`
Return the cosine of x radians.

`math.hypot(x, y)`
Return the Euclidean norm, `sqrt(x*x + y*y)`. This is the length of the vector from the origin to point `(x, y)`.

`math.sin(x)`
Return the sine of x radians.

`math.tan(x)`
Return the tangent of x radians.

Angular conversion

`math.degrees(x)`
Convert angle x from radians to degrees.

`math.radians(x)`
Convert angle x from degrees to radians.

Hyperbolic functions

BioPython

BioPython é uma biblioteca para facilitar o desenvolvimento de aplicações para bioinformática utilizando a linguagem Python

Pagina: www.biopython.org

Criado por: Champman B, Chang J

Surgiu em: 2000

Mantido por: Open Bioinformatics Foundation





Python Tools for
Computational
Molecular Biology

Documentation

Download

Mailing lists

News

Biopython Contributors

Scriptcentral

Source Code

GitHub project

Biopython version 1.73

© 2019. All rights
reserved.

Biopython itself.

[HTML](#) | [PDF](#)

- The **Biopython Tutorial and Cookbook** contains the bulk of Biopython documentation. It provides information to get you started with Biopython, in addition to specific documentation on a number of modules

[HTML](#) | [PDF](#)

- API documentation for Biopython modules is generated directly from source code comments using [Epydoc](#).

[HTML](#)

- [Wiki documentation](#)
 - [Seq](#) and [SeqRecord](#) objects
 - [Bio.SeqIO](#) - sequence input/output
 - [Bio.AlignIO](#) - alignment input/output
 - [Bio.PopGen](#) - population genetics
 - [Bio.PDB](#) - structural bioinformatics
 - Biopython's [BioSQL interface](#)
- Documentation for the Biopython interfaces to BioSQL cover installing Python database adaptors and basic usage of BioSQL.

[HTML](#) | [PDF](#)

- Cookbook-style documentation:
 - [Cookbook documentation](#) (on the wiki).
 - The Biopython Structural Bioinformatics FAQ (i.e. how to use the Bio.PDB module).

```
1 import Bio
2
```

```
1 from Bio.Seq import Seq
2
3 minha_primeira_seq = Seq("ATG")
4 print(minha_primeira_seq)
5
```

```
1 >titulo_da_sequencia
2 CGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGAT
3
```

```
1 from Bio import SeqIO
2
3 for fasta in SeqIO.parse("fastaexemplo.fasta", "fasta"):
4     print(fasta.id)
5     print(fasta.seq)
6
```

NCBI Resources How To

Nucleotide Nucleotide Advanced

The Nucleotide database will include EST and GSS sequences in early 2019. [Read more.](#)

FASTA

Human 18S ribosomal RNA

GenBank: X03205.1

[GenBank](#) [Graphics](#)

```
>X03205.1 Human 18S ribosomal RNA
TACCTGGTTGATCCTGCCAGTAGCATATGCTTGTCTCAAAGATTAAGCCATGCATGTCTAAGTACGCACG
GCCGGTACAGTGAACTGCGAATGGCTCATTAAATCAGTTATGGTTCTTTGGTCGCTCGCTCCTCTCCC
ACTTGGATAACTGTGGTAATTCTAGAGCTAATACATGCCGACGGGCGCTGACCCCTTCGCGGGGGGGAT
GCGTGCAATTTATCAGATCAAAACCAACCCGGTCAGCCCCCTCTCCGGCCCCGGCCGGGGGGGGGCGCCGG
CGGCTTTGGTGACTCTAGATAACCTCGGGCCGATCGCACGCCCCCGTGCGCGGCGACGACCCATTCTGAAC
GTCTGCCCTATCAACTTTCGATGGTAGTCGCCGTGCCATACCATGGTGACCACGGGTGACGGGGAATCAGG
GTTTCGATTCGGGAGAGGGAGCCTGAGAAACGGCTACCACATCCAAGGAAGGCAGCAGGCGCGCAATTAC
CCACTCCCGACCCGGGGAGGTAGTGACGAAAAATAACAATACAGGACTCTTTCGAGGCCCTGTAATTGGA
ATGAGTCCACTTTAAATCCTTTAACGAGGATCCATTGGAGGGCAAGTCTGGTGCCAGCAGCCGCGTAAT
TCCAGCTCCAATAGCGTATATTAAGTTGCTGCAGTTAAAAAGCTCGTAGTTGGATCTTGGGAGCGGGCG
GGCGGTCCGCGCGAGGCGAGCCACCGCCCGTCCCGCCCTTGCCCTCTCGCGCCCCCTCGATGCTCTT
AGCTGAGTGTCGCGGGGGCCGAAGCGTTTACTTTGAAAAAATTAGAGTGTTCAAAGCAGGCCGAGCC
GCCTGGATACGCGAGCTAGGAATAATGGAATAGGACCGCGGTTCTATTTGTTGGTTTTCGGAACGTAGG
CCATGATTAAGAGGGACGGCCGGGGGCAATTCGATTGCGCGCTAGAGGTGAAATTCCTGGACCGGCGCA
AGACGGACAGAGCGAAAGCATTTGCCAAGAATGTTTTCTTAATCAAGAACGAAAGTCGGAGGTTGAA
GACGATCAGATACCGTCGTAGTTCGACCATAAACGATGCCGACCGGCGATGCGGCGGCGTTATTCCTCAT
GACCCGCCGGGCGCTTCCGGGAAACCAAGTCTTTGGGTTCCGGGGGAGTATGGTTGCAAAGCTGAAA
CTTAAGGAATTGACGGAAGGGCACCAAGGAGTGGAGCCTGCGGCTTAATTTGACTCAACACGGGAAA
CCTCACCCGGCCGGACACGGACAGGATTGACAGATTGATAGCTCTTCTCGATTCCGTGGGTGGTGGTG
CATGGCGTTCTTAGTTGGTGAGCGATTTGTCTGGTTAATTCGATAACGAACGAGACTCTGGCATGCT
AACTAGTTACGCGACCCCCGAGCGGTGCGCTCCCCCACTTCTTAGAGGGACAAGTGGCGTTTACGCCAC
CCGAGATTGAGCAATAACAGGTCTGTGATGCCCTTAGATGTCGGGGCTGCACGCGCGCTACACTGACTG
GCTCAGCGTGTGCTACCTACGCCGGCAGGCGCGGGTAACCCGTTGAACCCCATTCGTGATGGGGATCG
GGGATTGCAATTATTCCTCATGAACGAGGAATCCAGTAAGTGCGGGTCATAAGCTTGCGTTGATTAAAG
TCCCTGCCCTTTGTACACACCGCCGTCGCTACTACCGATTGGATGGTTAGTGAGGCCCTCGGATCGGC
CCCGCCGGGGTCGGCCACGGCCCTGGCGGAGCGCTGAGAAGACGGTCGAACCTTGACTATCTAGAGGAAG
TAAAGTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATT
```

NCBI Resources How To

Nucleotide Nucleotide Advanced

The Nucleotide database will include EST and GSS sequences in early 2019. [Read more.](#)

FASTA

Escherichia coli strain U 5/41 16S ribosomal RNA, partial sequence

NCBI Reference Sequence: NR_024570.1

[GenBank](#) [Graphics](#)

>NR_024570.1 Escherichia coli strain U 5/41 16S ribosomal RNA, partial sequence
 AGTTTGATCATGGCTCAGATTGAACGCTGGCGGCAGGCCTAACACATGCAAGTCGAACGGTAACAGGAAG
 CAGCTTGCTGCTTTGCTGACGAGTGGCGGACGGGTAGTAATGTCTGGGAAACTGCCTGATGGAGGGGGA
 TAACTACTGGAACGGTAGCTAATACCGCATAACGTCGCAAGCACAAGAGGGGGACCTTAGGGCCTCTT
 GCCATCGGATGTGCCAGATGGGATTAGCTAGTAGGTGGGGTAACGGCTCACCTAGGCGACGATCCCTAG
 CTGGTCTGAGAGGATGACCAGCAACACTGGAAGTGAACACGGTCCAGACTCCTACGGGAGGCAGCAGTG
 GGGAAATTGCACAATGGGCGCAAGCCTGATGCAGCCATGCNGCGTGTATGAAGAAAGGCCTTCGGGTTGT
 AAAGTACTTTTCAGCGGGGAGGAAGGGAGTAAAGTTAATACCTTTGCTCATTGACGTTACCCGCAGAAGAA
 GCACCGGCTAACTCCGTGCCAGCAGCCGCGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGC
 GTAAAGCGCACGCAGGCGGTTTGTAAAGTCAGATGTGAAATCCCCGGGCTCAACCTGGGAAGTGCATCTG
 ATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAAATCCAGGTGTAGCGGTGAAATGCGTAGAGATCT
 GGAGGAATACCGGTGGCGAAGGCGGCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGC
 AAACAGGATTAGATACCCTGGTAGTCCACGCCGTAAACGATGTCGACTTGGAGGTTGTGCCCTTGAGGCG
 TGGCTTCCGGANNTAACGCGTTAAGTCGACCGCCTGGGAGTACGGCCGCAAGGTTAAAGTCAAAATGAA
 TTGACGGGGGCGCACAGCGGTGGAGCATGTGGTTAATTCGATGCAACGCGAAGAACCTTACCTGGTC
 TTGACATCCACGGAAGTTTTCAGAGATGAGAAATGTGCCCTCGGGAACCGTGAGACAGGTGCTGCATGGCT
 GTCGTGAGCTCGTGTGTGAAATGTTGGGTTAAGTCCCGCAACGAGCGCAACCTTATCCTTTGTTGCCA
 GCGGTCGGGCCGGGAAGTCAAGGAGACTGCCAGTGATAAAGTGGAGGAAGGTGGGGATGACGTCAAGTC
 ATCATGGCCCTTACGACAGGGCTACACACGTGCTACAATGGCGCATACAAAGAGAAGCGACCTCGCGAG
 AGCAAGCGGACCTCATAAAGTCGCTGCTAGTCCGGATTGGAGTCTGCAACTCGACTCCATGAAGTCGGAA
 TCGCTAGTAATCGTGATCAGAAATGCCACGGTGAATACGTTCCCGGGCTTGTACACACCGCCGTCACA
 CCATGGGAGTGGGTTGCAAAAGAAGTAGGTAGCTTAACCTCGGGAGGGCG