Programming: Python

报告人：张诏月

Programming

Way of structuring sets of instructions to enable a computer to perform a certain task

There are many programming languages

- C
- C++
- Java
- Perl
- **Python** Data processing, Text processing, Mathematic computation
- R Statistical analysis, Plot
- Matlab

python集成开发环境(ide)

PyCharm

Anaconda是一个运行环境，且兼容多个python版本

Jupyter是一个数据图形化GUI,组合文本，图像和代码

Google Colab https://colab.research.google.com/notebooks/intro.ipynb

Basic

注释 #

操作符

+	a + b	a plus b
-	a - b	a minus b
*	a * b	a times b
/	a / b	a divided by b
%	a % b	Remainder of a divided by b
**	a ** b	a to the power of b
==	a == b	a is equal to b
!=	a != b	a is not equal to b
>	a > b	a is greater than b
>=	a >= b	a is greater than or equal to b
<	a < b	a is smaller than b
<=	a <= b	a is smaller than or equal to b

变量和赋值

变量：字符开头，可包含字母、数字、_，大小写敏感

一些特殊单词如print def for 不能作为变量名

= 赋值

标准数据类型

- **数字** eg. 0, 1, -3.14
- **字符串** eg. ”apple”, ”0123”, ’ATTTAACGCGCGAA’
- **列表** eg. ['abcd', 786, 2.23, 'runoob', 70.2]
- 元组 eg. (‘abcd’, 786 , 2.23, ‘runoob’, 70.2) 不可修改
- 集合 eg. {‘abcd’, 786 , 2.23, ‘runoob’, 70.2} 集合运算
- **字典** eg. course = {“Database”:“Huang”,“Python”:“Zhang”,“Statistic”:“Sun”} 无序

缩进　四个空格/Tab

基本语句

- 赋值
- 输入输出语句　　input, print
- 条件判断语句　　if-elif-else
- 循环语句　　for-in, while, break/continue
- 异常处理语句　　try-except

文件读写 open('file','mode')
read/readline/readlines/write/close

函数
```
def function_name(args):
    code
    return result
```

调用包 from module import function as alias
包是含有Python模块的文件夹
常用的包/模块：os, sys, time, pandas, re

正则表达式
```
re.search(pattern, string)
pattern.findall(string[, pos[, endpos]])
re.sub(pattern, repl, string, count=0)
```

文件读写

函数

调用包

正则表达式

\	将下一个字符标记为一个特殊字符、或一个原义字符、或一个 向后引用、或一个八进制转义符。例如，'n' 匹配字符 "n"。'\n' 匹配一个换行符。序列 '\\' 匹配 "\" 而 "\(" 则匹配 "("。
^	匹配输入字符串的开始位置。
$	匹配输入字符串的结束位置。
*	**匹配前面的子表达式零次或多次。例如，zo* 能匹配 "z" 以及 "zoo"。**
+	**匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"。**
?	**匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 或 "does" 。**
{n}	匹配确定的 n 次。例如，'o{2}' 匹配 "food" 中的两个 o。
{n,}	至少匹配n 次。例如，'o{2,}' 匹配 "foooood" 中的所有 o。
{n,m}	最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中的前三个 o。
?	非贪婪模式：尽可能少的匹配所搜索的字符串，默认的贪婪模式则尽可能多的匹配所搜索的字符串。例如，对于字符串 "oooo"，'o+?' 将匹配单个 "o"，而 'o+' 将匹配所有 'o'。
.	匹配除换行符（\n、\r）之外的任何单个字符。要匹配包括 '\n' 在内的任何字符，"**(.|\n)**"。
x|y	**匹配 x 或 y。**
[xyz]	**字符集合。匹配所包含的任意一个字符。例如， '[abc]' 可以匹配 "plain" 中的 'a'。**
[^xyz]	**负值字符集合。匹配未包含的任意字符。例如， '[^abc]' 可以匹配 "plain" 中的'p'、'l'、'i'、'n'。**
[a-z]	**字符范围。匹配指定范围内的任意字符。例如，'[a-z]' 可以匹配 'a' 到 'z' 范围内的任意小写字母字符。**
[^a-z]	**负值字符范围。匹配任何不在指定范围内的任意字符。例如，'[^a-z]' 可以匹配任何不在 'a' 到 'z' 范围内的任意字符。**
\d	**匹配一个数字字符。等价于 [0-9]。**
\D	**匹配一个非数字字符。等价于 [^0-9]。**
\f	匹配一个换页符。等价于 \x0c 和 \cL。
\n	匹配一个换行符。等价于 \x0a 和 \cJ。
\r	匹配一个回车符。等价于 \x0d 和 \cM。
\s	**匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [\f\n\r\t\v]。**
\S	**匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。**
\t	匹配一个制表符。等价于 \x09 和 \cI。
\v	匹配一个垂直制表符。等价于 \x0b 和 \cK。
\w	**匹配字母、数字、下划线。等价于'[A-Za-z0-9_]'。**
\W	**匹配非字母、数字、下划线。等价于 '[^A-Za-z0-9_]'。**
\xn	匹配 n，其中 n 为十六进制转义值。十六进制转义值必须为确定的两个数字长。例如，'\x41' 匹配 "A"。'\x041' 则等价于 '\x04' & "1"。正则表达式中可以使用 ASCII 编码。
\b	匹配一个单词边界，也就是指单词和空格间的位置。例如， 'er\b' 可以匹配"never" 中的 'er'，但不能匹配 "verb" 中的 'er'。
\B	匹配非单词边界。'er\B' 能匹配 "verb" 中的 'er'，但不能匹配 "never" 中的 'er'。

Exercise

1. 计算1+3+5+…+49，并输出结果

2. 将DNA序列’ATCGGACGT’ 转化为相应的RNA序列

3. 对于exercise1.fasta文件(https://github.com/ZhaoyueZhang/Lin-Group-Bioinformatics-seminar)
   ①提取文件中序列Gene_Name并输出
   ②计算序列中核酸二联体(AA, AG, …,TT)的频率并输出到文件中