

Python速通指南

```
# 定义变量（动态类型）
x = 5
y = "Hello, World!"
z = 3.14

# 运算
a = 10
b = 3
c = a + b
c = a - b
c = a * b
c = a / b    # 浮点除法
c = a // b   # 整除
c = a % b
c = a**b    # 乘方
c = a**0.5  # 平方根(浮点)
c = a & b   # 按位与
c = a | b   # 按位或
c = a ^ b   # 按位异或
c = ~a      # 按位取反
c = a << 1
c = a >> 1

c += 1
c -= 1
# 没有c++或c--， 使用c += 1或c -= 1
c *= 2
c /= 2
c //= 2
# 等等

# 比较
c = a == b
c = a != b
c = a < b
c = a <= b
c = a > b
c = a >= b
c = a < b < c  # 链式比较

# 逻辑运算
```

```
c = a and b
c = a or b
c = not a

print(pow(2, 100000, 998244353)) # 自带快速幂
print(pow(2, -1, 998244353)) # 自带逆元
print(2**1000) # 自带高精度

# tuple
t = (1, 2, 3)
t = (1, "apple", 3.14, True)

# print
print(x)
print("Hello, World!")
print(f"Value of x: {x}, y: {y}, z: {z}") # f-string格式化
print("no endl", end="") # 不换行输出

# input
s = input() # 输入一行字符串
n = int(input()) # 输入一行一个整数
a, b = map(int, input().split()) # 输入一行多个整数, 空格分隔
a = list(map(int, input().split())) # 输入一行数组

# 条件语句
if a > b:
    print("a is greater than b")

if a > b:
    print("a is greater than b")
elif a < b:
    print("a is less than b")
else:
    print("a is equal to b")

if a > b:
    pass # 占位符
elif a < b:
    print("a is less than b")

# 循环
for i in range(5): # for(int i = 0; i < 5; i++)
    print(i)

for i in range(1, 6): # for(int i = 1; i <= 5; i++)
```

```
print(i)

for i in range(1, 10, 2): # for(int i = 1; i < 10; i += 2)
    print(i)

while a > 0: # while(a > 0)
    print(a)
    a -= 1

for i in range(5):
    if i == 2:
        continue
    if i == 4:
        break
    print(i)

# 列表（数组）
a = [1, 2, 3, 4, 5]
b = ["apple", "banana", "cherry"]
c = [1, "apple", 3.14, True]

a = [0] * 100 # 初始化一个长度为100的列表
b = [i + 1 for i in range(100)] # 列表推导式
c = [[0] * 100 for _ in range(100)] # 初始化一个100x100的二维列表

# 列表操作
b = [0, 1, 2] + [3, 4, 5]
b = [0, 1, 2] * 3

print(a)
print(a[0])
print(a[1:3]) # 切片，下标从0开始，左闭右开
print(a[-1]) # 访问最后一个元素
print(a[-2:]) # 访问倒数第二个及之后的元素
print(a[:3]) # 访问前三个元素
print(a[::-1]) # 反转列表

a.append(3) # v.push_back(3);
a.index(3) # find(v.begin(), v.end(), 3);
a.insert(0, 0) # v.insert(0, 0);
a.remove(3) # v.erase(3);
a.pop() # v.pop_back();
a.pop(0) # v.pop_front();
a.sort() # sort(v.begin(), v.end());
b = sorted(a)
```

```
a.reverse() # reverse(v.begin(), v.end());
a.count(3) # count(v.begin(), v.end(), 3);
a.clear() # 清空列表
del a[0] # 删除第一个元素

if 5 in a:
    print("5 is in the list")

t = tuple(a) # 转换为元组
a = list(t) # 转换为列表

# 字符串操作同理
s = "Hello, World!"
s2 = s + s
s2 += s * 10
print(s[0]) # 访问第一个字符
print(s[7:12]) # 切片，访问从第8到第12个字符
print(s[-1]) # 访问最后一个字符
print(s.lower()) # 转小写
print(s.upper()) # 转大写
print(s.replace("World", "Python")) # 替换字符串
print(s[::-1]) # 反转字符串
print(s.split(", ")) # 分割字符串
print(s.strip()) # 去除首尾空格

if "a" in s:
    print("'a' is in the string")

# set
s = set()
s = {5, 4, 3, 2, 1}
s.add(6) # st.insert(6);
s.remove(3) # st.erase(3);
s.discard(4) # if(st.count(4)) st.erase(4);
print(s)

if 5 in s: # if(st.count(5))
    print("5 is in the set")

for i in s: # for(auto i : st)
    print(i)

# dict
d = dict()
d = {"name": "Alice", "age": 30}
```

```
d["city"] = "New York" # d["city"] = "New York";
d["age"] = 31 # d["age"] = 31;
print(d["name"])

d.get("name", 0) # d.count("name") ? d["name"] : 0;
d.pop("age") # if(d.count("age")) d.erase("age");

if "name" in d: # if(d.count("name"))
    print("Name exists in the dictionary")

# 注意: 字典顺序是插入顺序
for key, value in d.items():
    print(f'{key}: {value}')

for key, value in sorted(d.items()):
    print(f'{key}: {value}')

for k in d:
    print(k, d[k])

for v in d.values():
    print(v)

# 函数
def fun():
    print("This is a function")
    return

def add(a, b):
    return a + b

# 文件
with open("file.txt", "w") as f:
    f.write("Hello, World!\n")
    f.write("This is a test file.\n")

with open("file.txt") as f:
    content = f.read()
    print(content)

with open("file.txt") as f:
    for line in f.readlines():
```

```
print(line.strip()) # 去除行末换行符

# import
import math

print(math.pi)
print(math.log(10))
print(math.log2(8))
print(math.gcd(10, 20))
print(math.lcm(10, 20))
print(math.factorial(5))
print(math.comb(5, 2)) # 组合数
print(math.perm(5, 2)) # 排列数
print(math.sin(math.pi / 2))
print(math.asin(1))
print(math.dist([1, 2], [4, 6])) # 计算两点间的距离
print(math.isqrt(10)) # floor(sqrt(10))

import random

random.seed(42)
print(random.randint(1, 10)) # 随机整数 [1, 10]
print(random.choice([1, 2, 3, 4, 5])) # 随机选择一个元素
print(random.sample([1, 2, 3, 4, 5], 3)) # 随机选择3个元素
print(random.uniform(1, 10)) # 随机浮点数 [1.0, 10.0]
print(random.random()) # 随机浮点数 [0.0, 1.0]
print(random.shuffle([1, 2, 3, 4, 5])) # 打乱列表顺序

import itertools

print(list(itertools.permutations([1, 2, 3]))) # 全排列
print(list(itertools.combinations([1, 2, 3], 2))) # 组合
print(list(itertools.combinations_with_replacement([1, 2, 3], 2))) # 带重复的组合
print(list(itertools.product([1, 2], [3, 4]))) # 笛卡尔积
print(list(itertools.product([1, 2], repeat=4)))

import functools

@functools.lru_cache(maxsize=None) # 自动记忆化
def fib(n):
    if n <= 1:
        return n
    return fib(n - 1) + fib(n - 2)
```

```
print(fib(100))

import os

os.system("echo Hello, World!") # 执行系统命令

import subprocess

subprocess.run(["std.exe"], stdin=open("input.txt"),
stdout=open("output.txt", "w")) # 执行外部程序
p = subprocess.run(["std.exe"], input="Hello, World!", text=True) #  
直接传入字符串作为输入
print(p.stdout) # 输出结果
```