



Lập trình Python cơ bản

Bài 6: Number – String – Datetime



Phòng Lập Trình - Mạng

2023

Nội dung

1. Number
2. String
3. Datetime



1. Number

□ Hàm toán học (import math)

Phương thức	Kết quả trả về	Ví dụ
<code>abs(x)</code>	Trị tuyệt đối của số nguyên x	<code>abs(-45)</code> # 45
<code>ceil(x)</code>	Giá trị nguyên nhỏ nhất lớn hơn x	<code>math.ceil(5.25)</code> # 6
<code>fabs(x)</code>	Trị tuyệt đối của số thực x	<code>math.fabs(-15)</code> # 15.0
<code>floor(x)</code>	Giá trị nguyên lớn nhất nhỏ hơn x	<code>math.floor(5.25)</code> # 5
<code>log(x)</code>	Logarit tự nhiên của một số	<code>math.log(5)</code> # 1.6094379124341003
<code>max(x₁, x₂, ..., x_n)</code>	Giá trị lớn nhất trong các tham số x	<code>max(10, 52, 15)</code> # 52
<code>min(x₁, x₂, ..., x_n)</code>	Giá trị nhỏ nhất trong các tham số x	<code>min(10, 52, 15)</code> # 10

1. Number

□ Hàm toán học (import math)

Phương thức	Kết quả trả về	Ví dụ
modf(x)	Tách x thành 2 phần: phần lẻ và phần nguyên. Kết quả trả về đều là kiểu float	math.modf(3.5) # (0.5, 3.0) math.modf(10) # (0.0, 10.0)
pow(x, y)	x^y	math.pow(2, 3) # 8.0 pow(2, 3) # 8
round(x[, n])	Làm tròn số, với n là số lẻ	round(12.25) # 12 round(12.25456, 3) # 12.255 round(12.25456, -1) # 10.0
sqrt(x)	Căn bậc 2 của x ($x>0$)	math.sqrt(8) # 2.8284271247461903
factorial(x)	Giai thừa của x	math.factorial(5) # 120
pi	Hằng số PI	math.pi # 3.141592653589793

1. Number

□ Hàm xử lý số ngẫu nhiên (import random)

Phương thức	Kết quả trả về	Ví dụ
randrange ([start,] stop [,step])	Một số ngẫu nhiên trong khoảng (start, stop, step)	random.randrange(1, 20, 1) # 11 random.randrange(2, 20, 2) # 4 random.randrange(1, 20, 2) # 15 random.randrange(20) # 10
randint(start, stop)	Một số ngẫu nhiên trong khoảng từ start đến stop	random.randint(1, 10) # 5 random.randint(5, 15) # 15
choice(seq)	Một item ngẫu nhiên trong một sequence (list, tuple hoặc string)	random.choice([1, 2, 3, 4, 5, 6]) # 5 random.choice('Hello Python') # P

1. Number

□ Hàm xử lý số ngẫu nhiên (import random)

Phương thức	Kết quả trả về	Ví dụ
sample(range(a,b), quantity)	Trả về quantity số ngẫu nhiên n với a <= n < b và n không trùng nhau	<pre>random.sample(range(1, 11), 5) # [9, 1, 3, 7, 6]</pre> <pre>random.sample(range(1, 20), 8) # [6, 1, 8, 4, 11, 19, 14, 15]</pre>
random()	Một số ngẫu nhiên r trong dãy $0 \leq r < 1$	<pre>random.random() # 0.005090615688038702</pre> <pre>random.random() # 0.9941983026288641</pre>
uniform(x, y)	Một số thực r trong dãy $x \geq r > y$	<pre>random.uniform(10, 20) # 10.063023672916664</pre>

Nội dung

1. Number
2. String
3. Datetime



2. String

□ Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
capitalize()	Kí tự đầu tiên của chuỗi được viết HOA	<pre>chuoi = 'trung tâm tin học' chuoi.capitalize() => Trung tâm tin học</pre>
lower()	Tất cả các ký tự trong chuỗi gốc thành chữ thường	<pre>chuoi = 'Trung Tâm Tin Học' chuoi.lower() => trung tâm tin học</pre>
upper()	Tất cả các ký tự trong chuỗi gốc thành chữ HOA	<pre>chuoi = 'trung tâm tin học' chuoi.upper() => TRUNG TÂM TIN HỌC</pre>
title()	Tất cả các ký tự đầu tiên của mỗi từ trong chuỗi gốc thành chữ HOA	<pre>chuoi = 'trung tâm tin học' chuoi.title() => Trung Tâm Tin Học</pre>

2. String

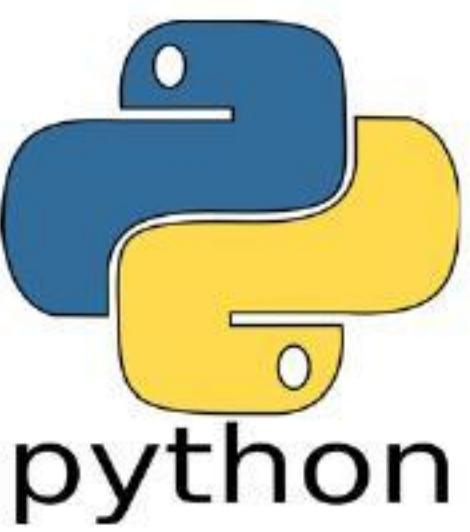
□ Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
center(width[, fillchar])	Tạo bản sao của chuỗi bằng cách điền vào <u>đầu</u> và <u>cuối</u> chuỗi bằng ký tự fillchar để có chiều dài chuỗi = width	s = 'Hello Python' s. center (20, '*') => ****Hello Python****
ljust(width[, fillchar])	Tạo bản sao của chuỗi bằng cách điền vào <u>cuối</u> chuỗi bằng ký tự fillchar để có chiều dài chuỗi = width	s = 'Hello Python' s. ljust (20, '*') => Hello Python*****
rjust(width[, fillchar])	Tạo bản sao của chuỗi bằng cách điền vào <u>đầu</u> chuỗi bằng ký tự fillchar để có chiều dài chuỗi = width	s = 'Hello Python' s. rjust (20, '*') => *****Hello Python

2. String

□ Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
<code>count(value[, start, end])</code>	Số lần xuất hiện của value trong chuỗi.	<pre>chuoi = 'Trung Tâm Tin Học' chuoi.count('Tâm') => 1</pre> <pre>chuoi.count('Tâm', 0, 10) => 1</pre>
<code>find(value[, start, end])</code>	<p>Số lần xuất hiện đầu tiên của value chỉ định.</p> <ul style="list-style-type: none"> - Trả về vị trí index nếu tìm thấy. - Trả về -1 nếu không tìm thấy. 	<pre>chuoi = 'Thành phố Hồ Chí Minh' s.find('phố') => 6</pre> <pre>s.find('c') => -1</pre>



2. String

Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
<code>strip([characters])</code>	Loại bỏ characters ở <u>đầu</u> và <u>cuối</u> chuỗi. Mặc định là loại bỏ khoảng trắng.	<pre>ten_dang_nhap = ' pythoncoban ' ten_dang_nhap.strip() => pythoncoban</pre> <pre>s = 'Lập trình Python thật đơn giản\n' s.strip('\n') => Lập trình Python thật đơn giản</pre>
<code>lstrip([characters])</code>	Loại bỏ characters ở <u>đầu</u> chuỗi. Mặc định là loại bỏ khoảng trắng.	<pre>ten_dang_nhap = ' pythoncoban ' ten_dang_nhap.lstrip() => pythoncoban </pre> <p style="text-align: center;"><small>Khoảng trắng</small></p>
<code>rstrip([characters])</code>	Loại bỏ characters ở <u>cuối</u> chuỗi. Mặc định là loại bỏ khoảng trắng.	<pre>ten_dang_nhap = ' pythoncoban ' ten_dang_nhap.rstrip() => </pre> <p style="text-align: center;"><small>Khoảng trắng</small></p>

2. String

□ Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
<code>replace(old_value, new_value[, count])</code>	Thay thế old_value bằng new_value . Nếu có sử dụng count thì sẽ thay thế count lần.	<pre>chuoi = 'Lập trình Python thật phức tạp' chuoi.replace('phức tạp', 'đơn giản') => Lập trình Python thật đơn giản</pre>
<code>split([separator, maxsplit])</code>	Tạo ra list các phần tử được cắt theo separator (mặc định là khoảng trắng). Tham số maxsplit chỉ định số lần cắt tối đa.	<pre>chuoi1 = 'Thành phố Hồ Chí Minh' chuoi1.split() => ['Thành', 'phố', 'Hồ', 'Chí', 'Minh'] chuoi2 = 'Hồ Chí Minh Hà Nội Đà Nẵng' chuoi2.split(' ') => ['Hồ Chí Minh', 'Hà Nội', 'Đà Nẵng']</pre>

2. String

□ Hàm xử lý chuỗi

Phương thức	Kết quả trả về	Ví dụ
join(iterable)	Nối các đối tượng chuỗi trong iterable với nhau.	<pre>chuoi = ' - ' list_chuoi = ['Hồ Chí Minh', 'Hà Nội', 'Đà Nẵng'] chuoi.join(list_chuoi) => Hồ Chí Minh - Hà Nội - Đà Nẵng</pre>
isdigit()	Trả về True nếu tất cả các ký tự là số, ngược lại là False .	<pre>nam = '2022' nam.isdigit() => True</pre>
isalpha()	Trả về True nếu chuỗi chứa toàn ký tự trong bảng chữ cái, ngược lại là False .	<pre>pwd = 'P@ssword' pwd.isalpha() => False</pre>
isalnum()	Trả về True nếu chuỗi chứa toàn ký tự trong bảng chữ cái hoặc ký số, ngược lại là False .	<pre>pwd = 'Password123' pwd.isalnum() => True</pre>

Nội dung

1. Number
2. String
3. Datetime



3. Datetime

□ Thư viện time (**import time**)

- **time.time()**: Trả về thời gian hiện tại của hệ thống dưới dạng số

```
import time  
seconds = time.time()  
print(seconds)
```

1667282909.2115467



3. Datetime

□ Thư viện time (**import time**)

- **time.ctime(secs)**: Trả về chuỗi thời gian theo giờ địa phương (local time)

```
import time
seconds = 1667282909.2115467
local_time = time.ctime(seconds)
print(local_time)
```

Tue Nov 1 13:08:29 2022

3. Datetime

□ Thư viện time (**import time**)

- **time.localtime(secs)**: Trả về class **time.struct_time** theo giờ địa phương (local time)

```
import time  
seconds = time.time()  
local_time = time.localtime(seconds)  
print(local_time)
```

```
time.struct_time(tm_year=2022, tm_mon=11, tm_mday=1, tm_hour=13, tm_min=34, tm_sec=43,  
tm_wday=1, tm_yday=305, tm_isdst=0)
```

- **time.gmtime(secs)**: Trả về class **time.struct_time** theo giờ UTC (Coordinated Universal Time)

– Giờ hợp nhất quốc tế)

```
gm_time = time.gmtime(seconds)  
print(gm_time)
```

```
time.struct_time(tm_year=2022, tm_mon=11, tm_mday=1, tm_hour=6, tm_min=34, tm_sec=43,  
tm_wday=1, tm_yday=305, tm_isdst=0)
```

3. Datetime

□ Thư viện time (**import time**)

- **time.sleep(secs)**: Hàm sẽ delay chương trình theo số giây truyền vào.

```
import time  
print('Chương trình sẽ bắt đầu sau 3 giây. Vui lòng chờ...')  
time.sleep(3)  
print('Bắt đầu!')
```

Chương trình sẽ bắt đầu sau 3 giây. Vui lòng chờ...
Bắt đầu!

3. Datetime

□ Thư viện time (**import time**)

- **time.asctime(struct_time)**: Kết quả trả về một chuỗi thời gian.

```
import time  
seconds = time.time()  
local_time = time.localtime(seconds)  
print('localtime:', time.asctime(local_time))  
gm_time = time.gmtime(seconds)  
print('gmtime:', time.asctime(gm_time))
```

localtime: Tue Nov 1 14:04:17 2022

gmtime: Tue Nov 1 07:04:17 2022

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.date.today()**: Kết quả trả về ngày hiện tại của hệ thống.

```
import datetime  
hom_nay = datetime.date.today()  
print('Hôm nay:', hom_nay)  
print('Ngày hiện tại:', hom_nay.day)  
print('Tháng hiện tại:', hom_nay.month)  
print('Năm hiện tại:', hom_nay.year)
```

Hôm nay: 2022-11-01
Ngày hiện tại: 1
Tháng hiện tại: 11
Năm hiện tại: 2022

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.date(year, month, day)**: Kết quả trả về ngày theo giá trị **year, month, day** truyền vào nếu hợp lệ. Ngược lại sẽ phát sinh lỗi.

```
import datetime  
print(datetime.date(2022, 11, 1))
```

2022-11-01

```
import datetime  
print(datetime.date(2022, 24, 1))
```

```
Traceback (most recent call last):  
  File "1:/GIANG_DAY/_PORTAL/Chuyen_de  
    print(datetime.date(2022, 24, 1))  
ValueError: month must be in 1..12
```

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.time([hour, minute, second]):** Kết quả trả về thời gian theo giá trị hour, minute, second truyền vào nếu hợp lệ. Ngược lại sẽ phát sinh lỗi. Giá trị mặc định của 3 tham số = 0.

```
import datetime  
print('Time 1:', datetime.time())  
print('Time 2:', datetime.time(16, 30, 55))
```

Time 1: 00:00:00

Time 2: 16:30:55

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.datetime.now()**: Kết quả trả về ngày giờ hiện tại của hệ thống.

```
import datetime
ngay_hien_tai = datetime.datetime.now()
print('Ngày hôm nay:', ngay_hien_tai)
print('Ngày hôm qua:', ngay_hien_tai + datetime.timedelta(-1))
print('Ngày mai:', ngay_hien_tai + datetime.timedelta(1))
```

Ngày hôm nay: 2022-11-01 16:54:37.195009

Ngày hôm qua: 2022-10-31 16:54:37.195009

Ngày mai: 2022-11-02 16:54:37.195009

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.datetime(year, month, day)**: Kết quả trả về ngày giờ theo giá trị **year, month, day** truyền vào nếu hợp lệ. Ngược lại sẽ phát sinh lỗi.
- Ngoài ra, class **datetime()** còn có các tham số tùy chọn (không bắt buộc) về thời gian (time) và múi giờ (timezone) như: **hour, minute, second, microsecond, tzname**

```
import datetime  
print(datetime.datetime(2022, 11, 1))
```

2022-11-01 00:00:00

```
import datetime  
print(datetime.datetime(2022, 24, 10))
```

PS L:\GIANG_DAY_PORTAL\Chuyen_de\LDS1_2022
File "l:/GIANG_DAY/_PORTAL/Chuyen_de/LDS1.
print(datetime.datetime(2022, 24, 10))
ValueError: month must be in 1..12

3. Datetime

□ Thư viện datetime (**import datetime**)

- **datetime.datetime.fromtimestamp(timestamp)**: Kết quả trả về ngày giờ theo giá trị timestamp truyền vào.

```
import time
import datetime
giay = time.time()
ngay_gio = datetime.datetime.fromtimestamp(giay)
print("Số giây:", giay)
print("Ngày giờ:", ngay_gio)
```

Số giây: 1667370509.842569
Ngày giờ: 2022-11-02 13:28:29.842569

3. Datetime

❑ Định dạng chuỗi

Directive	Description	Example
%a	Weekday, short version	Wed
%A	Weekday, full version	Wednesday
%w	Weekday as a number 0-6, 0 is Sunday	3
%d	Day of month 01-31	31
%b	Month name, short version	Dec
%B	Month name, full version	December
%m	Month as a number 01-12	12
%y	Year, short version, without century	18
%Y	Year, full version	2018
%H	Hour 00-23	17
%I	Hour 00-12	05
%p	AM/PM	PM
%M	Minute 00-59	41
%S	Second 00-59	08

3. Datetime

❑ Định dạng chuỗi

Directive	Description	Example
%f	Microsecond 000000-999999	548513
%z	UTC offset	+0100
%Z	Timezone	CST
%j	Day number of year 001-366	365
%U	Week number of year, Sunday as the first day of week, 00-53	52
%W	Week number of year, Monday as the first day of week, 00-53	52
%c	Local version of date and time	Mon Dec 31 17:41:00 2018
%C	Century	20
%x	Local version of date	12/31/18
%X	Local version of time	17:41:00
%%	A % character	%
%G	ISO 8601 year	2018
%u	ISO 8601 weekday (1-7)	1
%V	ISO 8601 weeknumber (01-53)	01

3. Datetime

❑ Định dạng chuỗi

- **datetime_variable.strftime(format)**: Kết quả trả về chuỗi theo định dạng.

```
import datetime  
ngay_hien_tai = datetime.datetime.now()  
print(ngay_hien_tai.strftime('%d-%m-%Y %H:%M:%S'))
```

01-11-2022 17:20:01



3. Datetime

□ Định dạng chuỗi

- **datetime.datetime.strptime(datetime_string, format)**: Phân tích chuỗi thành thời gian.

```
import datetime

ngay_tao = '2022-11-02'
print('Ngày tạo:', datetime.datetime.strptime(ngay_tao, '%Y-%m-%d'))

ngay_sinh = '11 July 1992'
print('Ngày sinh:', datetime.datetime.strptime(ngay_sinh, '%d %B %Y'))
```

Ngày tạo: 2022-11-02 00:00:00

Ngày sinh: 1992-07-11 00:00:00

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.isleap(year)**: Kết quả trả về **True** nếu là năm nhuận, ngược lại là **False**.

```
import calendar
print(calendar.isleap(2022)) # False
print(calendar.isleap(2024)) # True
```

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.month(year, month)**: Kết quả hiển thị calendar theo month, year.

```
import calendar  
print(calendar.month(2022, 11))
```

November 2022						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.calendar(year)**: Kết quả hiển thị calendar của năm truyền vào.

```
import calendar
print("The calendar of year 2023 is : ")
print(calendar.calendar(2023))
```

The calendar of year 2023 is :						
2023						
January						
Mo	Tu	We	Th	Fr	Sa	Su
					1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					
February						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					
March						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
April						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
30	31					
May						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
June						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		
July						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						
August						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
September						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	
October						
Mo	Tu	We	Th	Fr	Sa	Su
					1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					
November						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			
December						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.weekday(year, month, day)**: Kết quả trả về thứ của ngày/tháng/năm với giá trị số
 - 0: Monday
 - 1: Tuesday
 - ...

```
import calendar  
print(calendar.weekday(2023, 1, 1)) # 6
```

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.monthcalendar(year, month)**: Kết quả trả về danh sách các ngày trong tháng, năm được chọn, chia theo từng tuần, với giá trị 0 là ngày không nằm trong tháng, năm.

```
import calendar  
print(calendar.monthcalendar(2023, 1))
```

```
[[0, 0, 0, 0, 0, 0, 1], [2, 3, 4, 5, 6, 7, 8], [9, 10, 11, 12, 13, 14, 15], [16, 17, 18,  
, 19, 20, 21, 22], [23, 24, 25, 26, 27, 28, 29], [30, 31, 0, 0, 0, 0, 0]]
```

3. Datetime

□ Thư viện calendar (**import calendar**)

- **calendar.monthrange(year, month)**: Kết quả trả về hai phần tử
 - Phần tử đầu: là thứ của ngày đầu tiên trong tháng.
 - Phần tử thứ hai: là số ngày trong tháng.

```
import calendar  
print(calendar.monthrange(2023, 2))
```

(2, 28)

3. Datetime

□ Thư viện calendar (**import calendar**)

- **Class calendar.month_name:** Danh sách tháng trong năm

```
import calendar  
danh_sach_thang = calendar.month_name  
for thang in danh_sach_thang:  
    print(thang)
```

January
February
March
April
May
June
July
August
September
October
November
December



3. Datetime

□ Thư viện calendar (**import calendar**)

- **Class calendar.day_name:** Danh sách thứ trong tuần

```
import calendar
danh_sach_thu = calendar.day_name
for thu in danh_sach_thu:
    print(thu)
```

Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TIT TRUNG TÂM TIN HỌC
TTH