

# модуль calendar

1. Модуль `calendar`
2. Атрибуты `day_name`, `day_abbr`, `month_name`, `month_abbr`
3. Функции `setfirstweekday()`, `firstweekday()`
4. Функции `isleap()`, `leapdays()`
5. Функции `weekday()`, `monthrange()`, `monthcalendar()`
6. Функции `month()`, `calendar()`, `prmonth()`, `prcal()`

**Аннотация.** Урок посвящен модулю `calendar`, который содержит полезные типы данных и функции для работы с календарем.

## Модуль calendar

По умолчанию модуль `calendar` следует григорианскому календарю, где понедельник является первым днем недели (имеет номер 0), а воскресенье — последним днем недели (имеет номер 6). В отличие от уже изученных модулей `datetime` и `time`, которые также предоставляют функции, связанные с календарем, модуль `calendar` предоставляет основные функции, связанные с **отображением и манипулированием календарями**.

Прежде чем использовать модуль `calendar`, его необходимо подключить с помощью выражения:

```
import calendar
```

## Атрибуты модуля calendar

В отличие от функций, которые выполняют определенную работу, в модуле `calendar` есть полезные атрибуты, которые возвращают константные (общепринятые) значения, полезные при решении практических задач.

### Атрибут day\_name

Атрибут `calendar.day_name` возвращает итерируемый объект, содержащий названия дней недели на английском языке.

Приведенный ниже код:

```
import calendar

for name in calendar.day_name:
    print(name)
```

ВЫВОДИТ:

```
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
```



Обратите внимание, что при обращении к атрибуту мы не ставим скобки, которые ставим при вызове функции.

Для локализации на русский язык мы используем код:

```
import calendar, locale

locale.setlocale(locale.LC_ALL, 'ru_RU.UTF-8')

for name in calendar.day_name:
    print(name)
```

который выводит:

```
понедельник
вторник
среда
четверг
пятница
суббота
воскресенье
```



Обратите внимание, что на русском языке названия дней недели выводятся с маленькой буквы. Для того чтобы сделать первую букву заглавной, можно использовать строковый метод `title()`.

Для преобразования итерируемого объекта в список мы используем следующий код:

```
import calendar

names = list(calendar.day_name)
print(names)
```

который выводит:

```
['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
```

## Атрибут day\_abbr

Атрибут `calendar.day_abbr` возвращает итерируемый объект, содержащий сокращенные названия дней недели.

Приведенный ниже код:

```
import calendar, locale

for name in calendar.day_abbr:
    print(name)

locale.setlocale(locale.LC_ALL, 'ru_RU.UTF-8')

for name in calendar.day_abbr:
    print(name)
```

ВЫВОДИТ:

```
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Пн
Вт
Ср
Чт
Пт
Сб
Вс
```



Обратите внимание, что на русском языке сокращенные названия дней недели выводятся с большой буквы.

## Атрибут month\_name

Атрибут `calendar.month_name` возвращает итерируемый объект, содержащий названия месяцев года.

Приведенный ниже код:

```
import calendar, locale

english_names = list(calendar.month_name)
print(english_names)

locale.setlocale(locale.LC_ALL, 'ru_RU.UTF-8')

russian_names = list(calendar.month_name)
print(russian_names)
```

ВЫВОДИТ:

```
['', 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December']
['', 'Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль', 'Август',
'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь']
```



Обратите внимание, что атрибут `month_name` соответствует обычному соглашению, что январь – это месяц номер 1, поэтому список имеет длину в 13 элементов, первый из которых – пустая строка.

## Атрибут `month_abbr`

Атрибут `calendar.month_abbr` возвращает итерируемый объект, содержащий сокращенные названия месяцев года.

Приведенный ниже код:

```
import calendar, locale

english_names = list(calendar.month_abbr)
print(english_names)

locale.setlocale(locale.LC_ALL, 'ru_RU.UTF-8')

russian_names = list(calendar.month_abbr)
print(russian_names)
```

ВЫВОДИТ:

```
['', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',
'Dec']
['', 'янв', 'фев', 'мар', 'апр', 'май', 'июн', 'июл', 'авг', 'сен', 'окт', 'ноя',
'дек']
```

## Атрибуты номеров дней недели

Для получения номеров дней недели можно использовать атрибуты: `MONDAY`, `TUESDAY`, ..., `SUNDAY`.

Приведенный ниже код:

```
import calendar

print(calendar.MONDAY)
print(calendar.TUESDAY)
print(calendar.WEDNESDAY)
print(calendar.THURSDAY)
print(calendar.FRIDAY)
print(calendar.SATURDAY)
print(calendar.SUNDAY)
```

ВЫВОДИТ:

```
0
1
2
3
4
5
6
```

## Функции модуля calendar

Модуль `calendar` содержит множество полезных функций. Приведем основные из них.

### Функция `setfirstweekday()`

По умолчанию в модуле `calendar` понедельник является первым днем недели (имеет номер 0), а воскресенье – последним днем недели (имеет номер 6).

Функция `setfirstweekday()` позволяет изменить поведение по умолчанию и устанавливает заданный день недели в качестве начала недели.

Например, чтобы установить первый будний день воскресенье, мы используем код:

```
import calendar

calendar.setfirstweekday(calendar.SUNDAY)    # эквивалентно
calendar.setfirstweekday(6)
```



На практике следует использовать константы `calendar.MONDAY`, `calendar.TUESDAY`, ..., `calendar.SUNDAY`, а не значения `0`, `1`, ..., `6`.

### Функция `firstweekday()`

Функция `firstweekday()` возвращает целое число, означающее день недели, установленное в качестве начала недели.

Приведенный ниже код:

```
import calendar

print(calendar.firstweekday())
calendar.setfirstweekday(calendar.SUNDAY)
print(calendar.firstweekday())
```

ВЫВОДИТ:

```
0
6
```

## Функция isleap()

В курсе "Поколение Python": курс для начинающих мы решали задачу, в которой требовалось проверить високосность года. Напомним, что год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400. Модуль `calendar` содержит функцию `isleap()`, которая осуществляет нужную проверку.

Приведенный ниже код:

```
import calendar

print(calendar.isleap(2020))
print(calendar.isleap(2021))
```

ВЫВОДИТ:

```
True
False
```

## Функция leapdays()

Функция `leapdays(y1, y2)` возвращает количество високосных лет в диапазоне от `y1` до `y2` (исключая), где `y1` и `y2` – годы.

Приведенный ниже код:

```
import calendar

print(calendar.leapdays(2020, 2025))
```

ВЫВОДИТ:

```
2
```

так как в нужном диапазоне [2020; 2025) находятся два високосных года: 2020 и 2024.



Эта функция работает для диапазонов, охватывающих столетнюю эпоху.

## Функция `weekday()`

Функция `weekday(year, month, day)` возвращает день недели в виде целого числа (где 0 – понедельник, 6 – воскресенье) для заданной даты. Аргументы функции `year` – год начиная с 1970, `month` – месяц в диапазоне 1 – 12, `day` – число в диапазоне 1 – 31.

Приведенный ниже код:

```
import calendar

print(calendar.weekday(2021, 9, 1))    # среда
print(calendar.weekday(2021, 9, 2))    # четверг
```

ВЫВОДИТ:

```
2
3
```

## Функция `monthrange()`

Функция `monthrange(year, month)` возвращает день недели первого дня месяца и количество дней в месяце в виде кортежа для указанного года `year` и месяца `month`.

Приведенный ниже код:

```
import calendar

print(calendar.monthrange(2022, 1))    # январь 2022 года
print(calendar.monthrange(2021, 9))    # сентябрь 2021 года
```

ВЫВОДИТ:

```
(5, 31)
(2, 30)
```

## Функция `monthcalendar()`

Функция `monthcalendar(year, month)` возвращает матрицу, представляющую календарь на месяц. Каждая строка матрицы представляет неделю.

Приведенный ниже код:

```
import calendar

print(*calendar.monthcalendar(2021, 9), sep='\n')
```


ВЫВОДИТ:

```
[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, 0, 0, 0]
```

Обратите внимание на то, что дни, которые не входят в указанный месяц, представлены нулями. При этом каждая неделя начинается с понедельника, если не установлено другое функцией `setfirstweekday()`.

## Функция `month()`

Функция `month(year, month, w=0, l=0)` возвращает календарь на месяц в многострочной строке. Аргументами функции являются: `year` (год), `month` (месяц), `w` (ширина столбца даты) и `l` (количество строк, отводимые на неделю).

 Аргументы `w` и `l` имеют значения по умолчанию, поэтому их можно не передавать явно при вызове функции.

Приведенный ниже код:

```
import calendar

print(calendar.month(2021, 9))
print(calendar.month(2021, 10))
print(calendar.month(2021, 9, w=3))
print(calendar.month(2021, 9, l=2))
print(calendar.month(2021, 9, w=5, l=2))
```

ВЫВОДИТ:



### September 2021

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			

### October 2021

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

### September 2021

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			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		

### September 2021

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			

### September 2021

Mon	Tue	Wed	Thu	Fri	Sat	Sun
		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			

# Функция calendar()

Функция `calendar(year, w=2, l=1, c=6, m=3)` возвращает календарь на весь год в виде многострочной строки. Аргументами функции являются: `year` (год), `w` (ширина столбца даты) и `l` (количество строк, отводимые на неделю), `c` (количество пробелов между столбцом месяца), `m` (количество столбцов).



Аргументы `w`, `l`, `c`, `m` имеют значения по умолчанию, поэтому их можно не передавать явно при вызове функции.

Приведенный ниже код:

```
import calendar

print(calendar.calendar(2021))
```

ВЫВОДИТ:

2021

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	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

March

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				

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		

May

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						

June

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				

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	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					

September

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			

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	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					

December

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		

Приведенный ниже код:

```
import calendar, locale

locale.setlocale(locale.LC_ALL, 'ru_RU.UTF-8')

print(calendar.calendar(2022, m=4))
```

ВЫВОДИТ:



# Функции prmonth(), prcal()

Функция `prmonth(theyear, themonth, w=0, l=0)` печатает календарь на месяц, возвращенный функцией `month(theyear, themonth, w=0, l=0)`.

Функция `prcal(year, w=0, l=0, c=6, m=3)` печатает календарь на весь год, возвращенный функцией `calendar(year, w=0, l=0, c=6, m=3)`.

Следующий код:

```
import calendar

calendar.prmonth(2021, 9)
calendar.prcal(2021)
```

эквивалентен коду:

```
import calendar

print(calendar.month(2021, 9))
print(calendar.calendar(2021))
```

и выводит:

## September 2021

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			

2021

## 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	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

## March

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				

## 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		

## May

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						

## June

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				

## 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	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					

## September

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			

## 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	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					

## December

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		

# Примечания

**Примечание 1.** Официальная документация по модулю `calendar` [тут](#).

**Примечание 2.** Документация на русском языке [тут](#).

**Примечание 3.** Объекты, доступные по атрибутам `day_name`, `day_abbr`, `month_name` и `month_abbr`, поддерживают индексацию.

Приведенный ниже код:

```
import calendar

print(calendar.day_name[1])
print(calendar.day_abbr[1])
print(calendar.month_name[1])
print(calendar.month_abbr[1])
```

ВЫВОДИТ:

```
Tuesday
Tue
January
Jan
```