Манипуляция датой и временем

Модуль **datetime**:
манипуляции датой и временем

Модуль **datetime**

Модуль **datetime** позволяет манипулировать датой и временем: выполнять арифметические операции, сравнивать даты, выводить дату и время в различных форматах и др. Прежде чем использовать классы из этого модуля, необходимо подключить модуль с помощью инструкции:

```
import datetime
```

Модуль содержит пять классов:

timedelta — дата в виде количества дней, секунд и микросекунд. Экземпляр этого класса можно складывать с экземплярами классов date и datetime. Кроме того, результат вычитания двух дат будет экземпляром класса timedelta;

date — представление даты в виде объекта;

time — представление времени в виде объекта;

datetime — представление комбинации даты и времени в виде объекта;

tzinfo — абстрактный класс, отвечающий за зону времени. За подробной информацией по этому классу обращайтесь к документации по модулю datetime.

Класс **time**

Класс **time**

Класс time из модуля datetime позволяет выполнять операции над значениями времени.
Конструктор класса имеет следующий формат:

time(hour, minute, second, microsecond, tzinfo, fold)

Все параметры являются **необязательными**.

Класс **time**

В параметрах конструктора можно указать следующие диапазоны значений:

hour — часы (число от 0 до 23);
minute — минуты (число от 0 до 59);
second — секунды (число от 0 до 59);
microsecond — микросекунды (число от 0 до 999999);
tzinfo — зона (экземпляр класса tzinfo или значение None);
fold — порядковый номер отметки времени. Значение 0 (используется по умолчанию) обозначает первую отметку, значение 1 — вторую. Введено в Python 3.6 для тех случаев, когда в данной временной зоне практикуется перевод часов с зимнего на летнее время и обратно, в результате чего часы могут дважды в сутки показывать одинаковое время.

Если значения выходят за диапазон, возбуждается исключение **ValueError**

Пример использования конструктора

```
import datetime

t = datetime.time(hour = 21, second=38, minute=12)
print(repr(t))
print(str(t))
```

Результат можно получать используя следующие атрибуты:

hour — часы (число от 0 до 23);
minute — минуты (число от 0 до 59);
second — секунды (число от 0 до 59);
microsecond — микросекунды (число от 0 до 999999);
tzinfo — зона (экземпляр класса tzinfo или значение None);
fold — порядковый номер отметки времени. Значение 0 (используется по умолчанию) обозначает первую отметку, значение 1 — вторую. Введено в Python 3.6

```
t = datetime.time(23, 12, 38, 375000)
print('Час:', t.hour, 'Мин:', t.minute, 'Сек: ', t.second,'мс:', t.microsecond)
```

Над экземплярами класса time нельзя выполнять арифметические операции. Можно только производить сравнения.

```
t1 = datetime.time(23, 12, 38, 375000)
t2 = datetime.time(12, 28, 17)
print('t1 =', t1)
print('t2 =', t2)
print('t1 < t2 =', t1 < t2)
print('t1 > t2 =', t1 > t2)
print('t1 <= t2 =', t1 <= t2)
print('t1 >= t2 =', t1 >= t2)
```

Класс **time** поддерживает следующие методы:

replace(hour, minute, second, microsecond, tzinfo) — возвращает время с обновленными значениями. Значения можно указывать через запятую в порядке следования параметров или присвоить значение через название параметра:

```
import datetime


t = datetime.time(23, 12, 38, 375000)
print(t.replace(10, 52))
print(t.replace(second=21))
```

Класс **time** поддерживает следующие методы:

isoformat() — возвращает время в формате ISO 8601:

```
import datetime

t = datetime.time(23, 12, 38, 375000)
print(t.isoformat())
```

Класс **time** поддерживает следующие методы:

strftime(**"Строка формата"**) — возвращает отформатированную строку. В строке формата можно указывать комбинации специальных символов.

```
import datetime

t = datetime.time(23, 12, 38, 375000)
print(t.strftime("%H:%M:%S"))
```

Класс **datetime**

Класс **datetime**

Класс **datetime** из модуля **datetime** позволяет выполнять операции над комбинацией даты и времени. Конструктор класса имеет следующий формат:

datetime(Год, Месяц, День, hour, minute, second, microsecond, tzinfo, fold)

Первые три параметра являются обязательными. Остальные значения можно указывать через запятую в порядке следования параметров или присвоить значения по названиям параметров.

Класс **datetime**

Год— в виде числа, расположенного в диапазоне между значениями, хранящимися в константах MINYEAR (1) и MAXYEAR (9999);
Месяц — число от 1 до 12 включительно;
День — число от 1 до количества дней в месяце;
hour — часы (число от 0 до 23);
minute — минуты (число от 0 до 59);
second — секунды (число от 0 до 59);
microsecond — микросекунды (число от 0 до 999999);
tzinfo — зона (экземпляр класса tzinfo или значение None);
fold — порядковый номер отметки времени. Значение 0 (используется по умолчанию) обозначает первую отметку, значение 1 — вторую. Введено в Python 3.6 для тех случаев, когда в данной временной зоне практикуется перевод часов с зимнего на летнее время и обратно, в результате чего часы могут дважды в сутки показывать одинаковое время.

```
import datetime

dt = datetime.datetime(2017, 11, 21)
print(dt)
dt = datetime.datetime(2017, 11, 21, hour=17, minute=47)
print(dt)
dt = datetime.datetime(2017, 11, 21, 17, 47, 43)
print(repr(dt))
print(str(dt))
```

```
2017-11-21 00:00:00
2017-11-21 17:47:00
datetime.datetime(2017, 11, 21, 17, 47, 43)
2017-11-21 17:47:43
```

Для создания экземпляра класса datetime также можно воспользоваться следующими методами:

today() — возвращает текущие дату и время:

```
import datetime

dt = datetime.datetime.today()
print(dt)
```

now(Зона) — возвращает текущие дату и время. Если параметр не задан, то метод аналогичен методу today():

```
import datetime

dt = datetime.datetime.now()
print(dt)
```

fromtimestamp(Количество секунд, [Зона]) — возвращает дату, соответствующую количеству секунд, прошедших с начала эпохи:

```
import datetime

dt = datetime.datetime.fromtimestamp(1500000000)
print(dt)
```

fromordinal(Количество дней с 1-го года) — возвращает дату, соответствующую количеству дней, прошедших с 1-го года. В качестве параметра указывается число от 1 до datetime.datetime.max.toordinal():

```
import datetime

dt = datetime.datetime.fromordinal(737059)
print(dt)
```

strptime(<Строка с датой>, <Строка формата>) — разбирает строку, указанную в первом параметре, в соответствии со строкой формата, создает на основе полученных из разобранной строки данных экземпляр класса datetime и возвращает его. Если строка не соответствует формату, возбуждается исключение ValueError. Метод учитывает текущую локаль:

```
dt = datetime.datetime.strptime("21.11.2019", "%d.%m.%Y")
print(dt)
```

Результат можно получать используя следующие атрибуты:

year — год (число в диапазоне от MINYEAR до MAXYEAR);

month — месяц (число от 1 до 12);

day — день (число от 1 до количества дней в месяце).

hour — часы (число от 0 до 23);

minute — минуты (число от 0 до 59);

second — секунды (число от 0 до 59);

microsecond — микросекунды (число от 0 до 999999);

tzinfo — зона (экземпляр класса tzinfo или значение None);

fold — порядковый номер отметки времени. Значение 0 (используется по умолчанию) обозначает первую отметку, значение 1 — вторую. Введено в Python 3.6

```
dt = datetime.datetime.now()
print('День:', dt.day, 'Часы:', dt.hour, 'Минуты:', dt.minute)
```

Над экземплярами класса datetime можно производить операции:

datetime2 = datetime1 + timedelta — прибавляет к дате указанный период;
datetime2 = datetime1 — timedelta — вычитает из даты указанный период;
timedelta = datetime1 — datetime2 — возвращает разницу между датами.
Можно также сравнивать две даты с помощью операторов сравнения.

```
d1 = datetime.datetime(2021, 9, 21, 17, 54, 8)
d2 = datetime.datetime(2021, 9, 1, 12, 31, 4)
t = datetime.timedelta(days=10, minutes=10)
print('d1 + t =', d1 + t)
print('d1 - t =', d1 - t)
print('d1 - d2 =', d1 - d2)
print('d1 > d1 :', d1 > d2, 'd1 != d1 :', d1 != d2)
```

Класс datetime поддерживает следующие методы:

date() — возвращает экземпляр класса date, хранящий дату

time() — возвращает экземпляр класса time, хранящий время

timetz() — возвращает экземпляр класса time, хранящий время

timestamp() — возвращает вещественное число, представляющее количество секунд, прошедшее с начала эпохи (с 1 января 1970 г.)

replace(year, month, day, hour, minute, second, microsecond,tzinfo) — возвращает дату с обновленными значениями. Значения можно указывать через запятую в порядке следования параметров или присвоить значения по названиям параметра

timetuple() — возвращает объект struct_time с датой и временем

weekday() — возвращает порядковый номер дня в неделе (0 — для понедельника, 6 — для воскресенья):

strftime(Строка формата) — возвращает отформатированную строку. В строке формата можно указывать комбинации специальных символов