http://pythonworld.ru/moduli/modul-datetime.html#more-730

Модуль datetime предоставляет классы для обработки времени и даты разными способами. Поддерживается и стандартный способ представления времени, однако больший упор сделан на простоту манипулирования датой, временем и их частями.

**Классы, предоставляемые модулем datetime:**

Класс **datetime.date**(year, month, day) - стандартная дата. Атрибуты: year, month, day. Неизменяемый объект.

Класс **datetime.time**(hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - стандартное время, не зависит от даты. Атрибуты: hour, minute, second, microsecond, tzinfo.

Класс **datetime.timedelta** - разница между двумя моментами времени, с точностью до микросекунд.

Класс **datetime.tzinfo** - абстрактный базовый класс для информации о временной зоне (например, для учета часового пояса и / или летнего времени).

Класс **datetime.datetime**(year, month, day, hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - комбинация даты и времени.

Обязательные аргументы:

* datetime.MINYEAR (1) <= year <= datetime.MAXYEAR (9999)
* 1 <= month <= 12
* 1 <= day <= количество дней в данном месяце и году

Необязательные:

* 0 <= minute < 60
* 0 <= second < 60
* 0 <= microsecond < 1000000

Методы класса datetime:

Наиболее интересные директивы используемые для форматирования времени.  
Расположены не в алфавитном порядке, а в логическом)

* %S — секунды. От 0 до 61
* %M — минуты. От 00 до 59
* %H — час. От 00 до 23
* %I — час. От 1 до 12
* %p -После перед полуднем или после (AM или PM)
* %d — день. От 1 до 31
* %j — день как номер года. От 001 до 366
* %m — месяц. От 01 до 12
* %y — год в виде 2-х последних чисел. От 00 до 99
* %Y — год в виде полного числа

**datetime.today()** - текущая дата, время равно 0.

**datetime.fromtimestamp(timestamp)** - дата из стандартного представления времени.

**datetime.fromordinal(ordinal)** - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.

**datetime.now**(tz=None) - объект datetime из текущей даты и времени.

**datetime.combine**(date, time) - объект datetime из комбинации объектов date и time.

**datetime.strptime**(date\_string, format) - преобразует строку в datetime (так же, как и функция strptime из [модуля time](http://pythonworld.ru/moduli/modul-time.html)).

**datetime.strftime**(format) - см. функцию strftime из модуля time.

**datetime.date**() - объект даты (с отсечением времени).

**datetime.time**() - объект времени (с отсечением даты).

**datetime.replace**([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]]]) - возвращает новый объект datetime с изменёнными атрибутами.

**datetime.timetuple**() - возвращает struct\_time из datetime.

**datetime.toordinal**() - количество дней, прошедших с 01.01.1970.

**datetime.timestamp**() - возвращает время в секундах с начала эпохи.

**datetime.weekday**() - день недели в виде числа, понедельник - 0, воскресенье - 6.

**datetime.isoweekday**() - день недели в виде числа, понедельник - 1, воскресенье - 7.

**datetime.isocalendar**() - кортеж (ISO year, ISO week number, ISO weekday).

**datetime.isoformat**(sep='T') - красивая строка вида "YYYY-MM-DDTHH:MM:SS.mmmmmm" или, если microsecond == 0, "YYYY-MM-DDTHH:MM:SS"

**datetime.ctime**() - см. ctime() из модуля time.

Пример работы с классом datetime:

>>> from datetime import datetime, date, time

>>> *# Using datetime.combine()*

>>> d = date(2005, 7, 14)

>>> t = time(12, 30)

>>> datetime.combine(d, t)

datetime.datetime(2005, 7, 14, 12, 30)

>>> *# Using datetime.now() or datetime.utcnow()*

>>> datetime.now()

datetime.datetime(2007, 12, 6, 16, 29, 43, 79043) *# GMT +1*

>>> datetime.utcnow()

datetime.datetime(2007, 12, 6, 15, 29, 43, 79060)

>>> *# Using datetime.strptime()*

>>> dt = datetime.strptime("21/11/06 16:30", "%d/%m/%y %H:%M")

>>> dt

datetime.datetime(2006, 11, 21, 16, 30)

>>> *# Using datetime.timetuple() to get tuple of all attributes*

>>> tt = dt.timetuple()

>>> for it in tt:

... print(it)

...

2006 *# year*

11 *# month*

21 *# day*

16 *# hour*

30 *# minute*

0 *# second*

1 *# weekday (0 = Monday)*

325 *# number of days since 1st January*

-1 *# dst - method tzinfo.dst() returned None*

>>> *# Date in ISO format*

>>> ic = dt.isocalendar()

>>> for it in ic:

... print(it)

...

2006 *# ISO year*

47 *# ISO week*

2 *# ISO weekday*

>>> *# Formatting datetime*

>>> dt.strftime("%A, %d. %B %Y %I:%M%p")

'Tuesday, 21. November 2006 04:30PM'