★ 由 Xnip 截图  The property of the propert			
指令	意义	示例	注释
%a	当地工作日的缩写。	Sun, Mon,, Sat (en_US); So, Mo,, Sa (de_DE)	(1)
%A	本地化的星期中每日的完整名称。	Sunday, Monday,, Saturday (en_US); Sonntag, Montag,, Samstag (de_DE)	(1)
%w	以十进制数显示的工作日,其中0表示星期日, 6表示星期六。	0, 1,, 6	
%d	补零后,以十进制数显示的月份中的一天。	01, 02,, 31	(9)
%b	当地月份的缩写。	Jan, Feb,, Dec (en_US); Jan, Feb,, Dez (de_DE)	(1)
%B	本地化的月份全名。	January, February,, December (en_US); Januar, Februar,, Dezember (de_DE)	(1)
%m	补零后,以十进制数显示的月份。	01, 02,, 12	(9)
&À	补零后,以十进制数表示的,不带世纪的年 份。	00, 01,, 99	(9)
& Y	十进制数表示的带世纪的年份。	0001, 0002,, 2013, 2014,, 9998, 9999	(2)
%Н	以补零后的十进制数表示的小时(24 小时 制)。	00, 01,, 23	(9)
%I	以补零后的十进制数表示的小时(12 小时 制)。	01, 02,, 12	(9)
%p	本地化的 AM 或 PM 。	AM, PM (en_US); am, pm (de_DE)	(1), (3)
%M	补零后,以十进制数显示的分钟。	00, 01,, 59	(9)
%S	补零后,以十进制数显示的秒。	00, 01,, 59	(4), (9)
%f	以十进制数表示的毫秒,在左侧补零。	000000, 000001,, 999999	(5)
% Z	UTC 偏移量,格式为 ±HHMM[SS[.ffffff]] (如果是简单型对象则为空字符串)。	(空), +0000, -0400, +1030, +063415, -030712.345216	(6)
% Z	时区名称(如果对象为简单型则为空字符 串)。	(空), UTC, EST, CST	
%j	以补零后的十进制数表示的一年中的日序号。	001, 002,, 366	(9)
%U	以补零后的十进制数表示的一年中的周序号 (星期日作为每周的第一天)。 在新的一年中 第一个星期日之前的所有日子都被视为是在第 0周。	00, 01,, 53	(7), (9)
%W	以十进制数表示的一年中的周序号(星期一作 为每周的第一天)。 在新的一年中第一个第期 一之前的所有日子都被视为是在第 0 周。	00, 01,, 53	(7), (9)
%c	本地化的适当日期和时间表示。	Tue Aug 16 21:30:00 1988 (en_US); Di 16 Aug 21:30:00 1988 (de_DE)	(1)
%x	本地化的适当日期表示。	08/16/88 (None); 08/16/1988 (en_US); 16.08.1988 (de_DE)	(1)
%X	本地化的适当时间表示。	21:30:00 (en_US); 21:30:00 (de_DE)	(1)
88	字面的 '%' 字符。	%	
为了方便起见,还包括了C89标准不需要的其他一些指令。这些参数都对应于ISO 8601日期值。			
指令	意义	示例	注释
%G	带有世纪的 ISO 8601 年份,表示包含大部分 ISO 星期 (%v) 的年份。	0001, 0002,, 2013, 2014,, 9998, 9999	(8)
%u	以十进制数显示的 ISO 8601 星期中的日序号,其中 1 表示星期一。	1, 2,, 7	
%V	以十进制数显示的 ISO 8601 星期,以星期一作为每周的第一天。 第 01 周为包含 1 月 4日的星期。	01, 02,, 53	(8), (9)

datetime模块

格式代码

date(MINYEAR, 1, 1) date.min 类属性 date.max date(MAXYEAR, 12, 31) timedelta(days=1) date.resolution 实例属性 .\_\_getattribute\_\_('year') \_\_eq\_\_(...) 等于(x==y) x.\_\_eq\_\_(y) \_\_ge\_\_(...) 大于等于(x>=y) x.\_\_ge\_\_(y) \_\_gt\_\_(...) 大于(x>y) x.\_\_gt\_\_(y) \_\_le\_\_(...) 小于等于(x<=y) x.\_\_le\_\_(y) \_\_lt\_\_(...) 小于(x x.\_\_lt\_\_(y) \_\_ne\_\_(...) 不等于(x!=y) x.\_\_ne\_\_(y) 返回timedelta对象 date1 - date2 返回一个包含三个值的元组,三个值依次为: year年份, date(year, month, day) week number周数, weekday星期数(周一为1...周日为7) .isocalender() .fromisocalendar(year, week, day) 返回date对象 返回 YYYY-MM-DD 格式的日期字符串 .isoformat() ISO标准 .fromisoformat(date\_string) 返回一个对应于以 YYYY-MM-DD 格式给出的 date\_string 的 date 对象 返回日期所在的星期数(周一为1...周日为7) .isoweekday() 周一为0...周日为6 .weekday() .\_\_format\_\_('%Y-%m-%d') 等价于 .strftime('%Y-%m-%d') 字符串输出 .\_\_str\_\_() '2017-03-22' .ctime() 'Wed Mar 22 00:00:00 2017' 返回一个类型为time.struct\_time的数组,类似于time.localtime() time.struct\_time((d.year, d.month, d.day, 0, 0, 0, d.weekday(), yday, -1)) .timetuple() hours, minutes 和 seconds 值均为 0, 且 DST 旗标值为 -1 返回公元公历开始(1年1月1日)到现在的天数 .toordinal() 其它 .fromordinal() .fromtimestamp(timestamp) 根据时间戳返回日期 返回一个替换指定日期字段的新date对象,不影响原date对象 .replace(year=, month=, day=) .today() 返回当前日期 time(0, 0, 0, 0)time.min 类属性 time(23, 59, 59, 999999) time.max timedelta(microseconds = I) time.resolution .hour、.minute、.second、.microsecond、.tzinfo、.fold 实例属性 .\_\_getattribute\_\_('hour') 比较 time(hour, minute, second, microsecond, tzinfo, fold=0) .isoformat(timespec='auto') .fromisoformat(time\_string) .replace(hour=, minute=, second=, microsecond=, tzinfo=, \* fold=0) .\_\_str\_\_() 方法 .strftime(format) .\_\_format\_\_(format) .utcoffset() .dst() .tzname() .today() .now(tz=None) .utcnow() .fromtimestamp(timestamp, tz=None) .utcfromtimestamp(timestamp) 返回对应于时间戳的 UTC datetime .fromordinal(ordinal) .combine(date, time, tzinfo=self.tzinfo) date对象和time对象结合 .fromisoformat(date\_string) .fromisocalendar(year, week, day) 返回一个对应于 date\_string,根据 format 进行解析得到的 datetime 对象 datetime.strptime(date\_string, format) 返回具有同样 hour, minute, second, microsecond, fold 和 tzinfo 属性性的 time 对象 .timetz() 返回一个具有新的 tzinfo 属性 tz 的 datetime 对象 .astimezone(tz=None) YYYY-MM-DDTHH:MM:SS.ffffff, 如果 microsecond 不为 0 utcoffset()返回None YYYY-MM-DDTHH:MM:SS, 如果 microsecond 为 0 YYYY-MM-DDTHH:MM:SS.ffffff+HH:MM[:SS[.ffffff]], 如果 microsecond 不为 0 .isoformat(sep='T', timespec='auto') utcoffset()不返回None YYYY-MM-DDTHH:MM:SS+HH:MM[:SS[.ffffff]], 如果 microsecond 为 0 可选参数 sep (默认为 'T') 为单个分隔字符, 可设为sep='' 会被放在结果的日期和时间两部分之间 datetime(year, month, day, hour, minute, second, microsecond, tzinfo, fold=0) .replace(year=, month=, day=, hour=, minute=, second=, microsecond=, tzinfo=, \* fold=) .utcoffset() 实例方法 .tzname() .timetuple() .utctimetuple() .toordinal() .timestamp() .weekday() .isoweekday() .isocalendar() .ctime() .strftime(format) .\_\_format\_\_(format) 类属性 min、max、resolution fold取0、1 用于在重复的时间段中消除边界时间歧义。取值 0 (1) year, month, day, hour, minute, 表示两个时刻早于(晚于)所代表的同一边界时间。 实例属性 second, microsecond, tzinfo, fold 只有 days, seconds 和 microseconds 会存储 days 属性 seconds timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0) microseconds timedelta.total\_seconds() 返回时间间隔包含了多少秒 datetime\_CAPI