

aardio 范例: 入门

```
//入门
import console;

console.log( time.tick(), "系统运行时间 (毫秒数)" )

//time 库文档: https://www.aardio.com/zh-cn/doc/library-guide/builtin/time/time.html
var tm = time.now();

//指定格式化时间
tm.format = "%a %B %Y %m %d %H:%M:%S";
tm.locale = "enu"; //整定格式化语言, 无论指定什么语言, 文本都必须是aardio默认的UTF8编码
console.log("时间格式化为字符串", toString(tm));

//也可以在 toString 函数中指定格式化参数
var str = toString(tm, "%Y年%m月%d日 %H时%M分%S秒", "chs")
console.log("时间格式化为简体中文", str);

/*
可使用格式串参数指定的规则重新将文本解析为时间对象,
不指定时间格式串时默认值为 '%Y/%m/%d %H:%M:%S', 此格式串可兼容解析 ISO8601 格式时间
*/
console.log( time("2017-05-27T16:56:01Z") )

/*
格式化输入文本并解析为时间时, 使用尽可能宽松的规则:

忽略日期时间首尾部不匹配字符。
一个空白字符可以匹配任意多个空白字符。
一个标点可以匹配任意个连续的标点。
一个字母可以匹配任意个连续的字母, 一个非 ASCII 字符可以匹配任意个连续的非 ASCII 字符。
支持 ISO8601 格式省略分隔符的写法。

如果输入文本中的时间数值超出日期范围, 则返回 null 。
但如果出现当月不存在的日期且小于 31 号时会顺推为下月时间。

解析时如果输入文本提前结束, 但解析时间未完成则返回 null 。
最后一个格式化标记解析成功以后如果还有剩余的字符串,
首先跳过前面的空白字符, 从前面取其他连续的非空白字符存入时间对象的endstr属性内。
endstr可用于后续解析iso8601等格式的时区。
*/
var tm = time(str, "%Y年%m月%d日 %H时%M分%S秒", "chs")

tm.year += 2;
console.log(tm, "增加2年")

tm.addsecond(30)
console.log(tm, "增加30秒")

tm.addminute(180)
console.log(tm, "增加180分")

tm.addhour(2)
console.log(tm, "增加两小时")

tm.addday(365)
console.log(tm, "增加365天")

tm.addmonth(-24)
console.log(tm, "倒退24个月")

var tm2 = time.now()

console.log( tm2.diffmonth(tm) , "相差月份")
console.log( tm2.diffday(tm) , "相差天数")
console.log( tm2.diffhour(tm) , "相差小时数")
console.log( tm2.diffminute(tm) , "相差分钟数")
console.log( tm2.diffsecond(tm) , "相差秒数")
console.log( tonumber(tm) - tonumber(tm2) , "相差秒数, 作用同上")

console.log('\n关系运算符, 相等、不等')
console.log( "tm2==tm", tm2 == tm )
console.log( "time.now()== time.now()", time.now()== time.now() )

console.log('\n关系运算符, 大于、小于')
```

```
console.log( "tm2>tm", tm2 > tm )
console.log( "time.now() > time.now()", time.now() > time.now() )

console.log('\n关系运算符,大于等于、小于等于')
console.log( "tm2 <= tm", tm2 <= tm )
console.log( "time.now() <= time.now()", time.now() <= time.now() )

/*
time.gmt() 创建一个time对象,并且设置HTTP协议协定的GMT时间格式,并将该对象的格式化语言设为英文
*/
console.log( time.gmt() )
console.log( time.gmt( tm2.utc() ) ); //参数是其他time对象时,必须先转换为UTC时间

//HTTP时间RFC 1123格式,写错了空格这些自动修正
console.log(time.gmt("Sun,07Feb2016 081122 +7"))

//兼容HTTP时间RFC 850格式,星期写错了自动修正
console.log(time.gmt("Sunddddday, 07-Feb-16 08:11:22 +0700"))

//支持iso8601省略分隔符的格式
console.log(time.iso8601("20170822 123623 +0700"))

//支持iso8601省略分隔符的格式
console.log(time.iso8601("20170822 123623 +7"))

//兼容iso8601省略分隔符的写法
console.log(time("20170822 123623"))

//省略分隔符的写法 14位数字
console.log(time("20170822123623"))

//省略分隔符的写法 12位数字
console.log(time("170822123623"))

console.pause()
```

[Markdown 格式](#)