

第 9 节 GEE 的参数类型 (Date, Date Range)

时间是 GEE 栅格数据的重要参数。通过本节的学习，我们将掌握如何在 GEE 中构建、读取和调整时间参数。

9.1 Date

下边介绍时间的创建，代码及执行效果如下：

```
var date_1 = ee.Date('1970-01-01')
var date_2 = ee.Date(24*60*60*1000)

print(date_1,date_2)
```

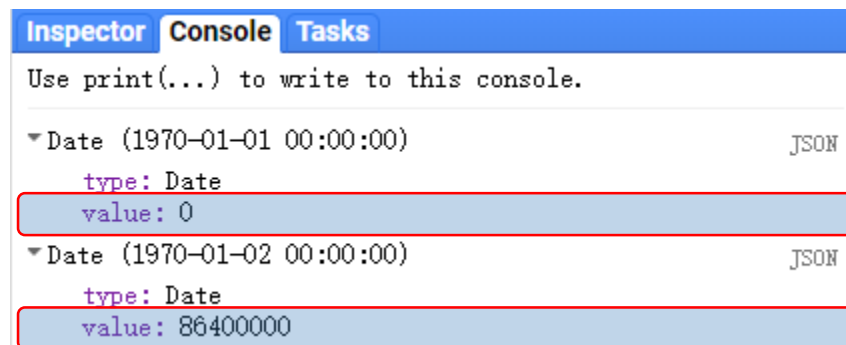


图 9.1 时间的创建

可以看出，GEE 中的时间是以 1970 年 1 月 1 日 0 时为起点，数值是距离目标时间的毫秒数。

下边介绍时间的格式创建，代码及执行效果如下：

```
var Date_From_YMD = ee.Date.fromYMD( 1992, 05, 10 );
print( Date_From_YMD );
```

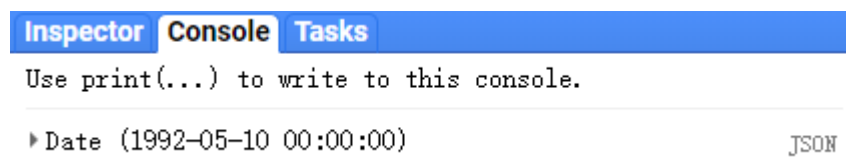


图 9.2 格式创建

下边介绍时间的 Parse 创建，代码及执行效果如下：

```
var TheDATE = ee.Date.parse( 'yyyy-MM-dd-HH-mm-ss',
                             '2019-03-27-16-26-12' );
print( TheDATE );
```

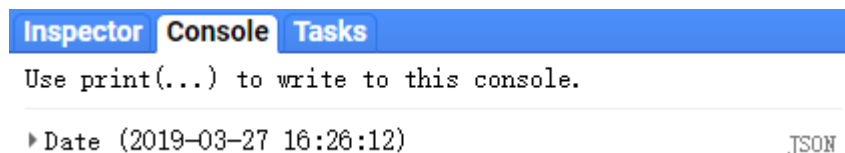


图 9.3 时间的 Parse 创建

下边介绍时间的单位增加命令，代码及执行效果如下：

```
var Date_0 = ee.Date( '1970-01-01' );
var Date_1 = Date_0.advance( 1, 'year' );
var Date_2 = Date_0.advance( 1, 'month' );
var Date_3 = Date_0.advance( 1, 'week' );
var Date_4 = Date_0.advance( 1, 'day' );
var Date_5 = Date_0.advance( 1, 'hour' );
var Date_6 = Date_0.advance( 1, 'minute' );
var Date_7 = Date_0.advance( 1, 'second' );

print( Date_1, Date_2, Date_3, Date_4, Date_5, Date_6, Date_7 );
```

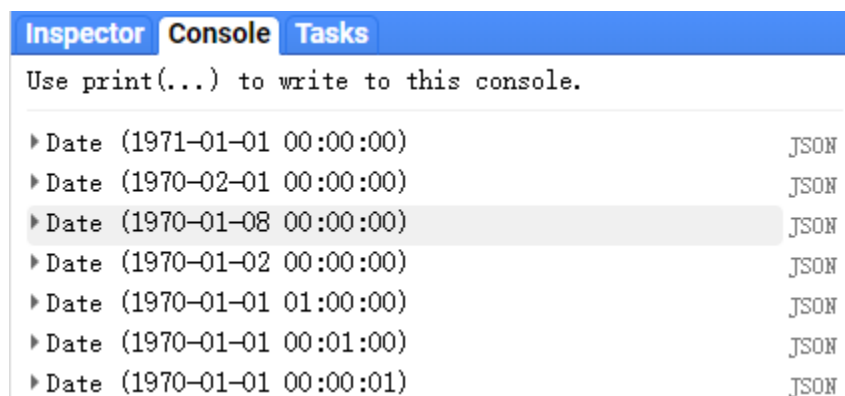


图 9.4 时间的单位增加

下边介绍时间的更新，代码及执行效果如下：

```
var Date_1 = ee.Date( '1992-05-10' );
var Date_2 = Date_1.update( 2019, 06, 9 );
print( Date_1, Date_2 );
```

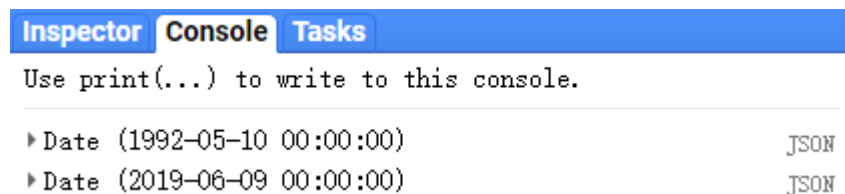
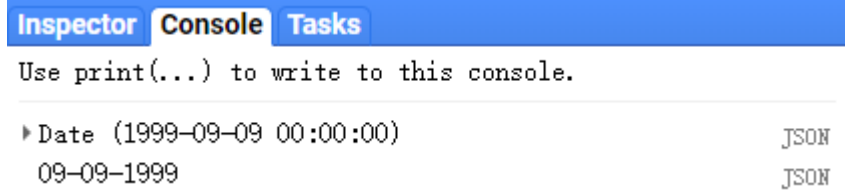


图 9.5 时间的更新

下边介绍时间的格式更改，代码及执行效果如下：

```
var Date_0 = ee.Date.fromYMD( 1999, 9, 9 );  
var Date_FMT = Date_0.format('MM-dd-yyyy');  
print( Date_0, Date_FMT );
```



Inspector Console Tasks

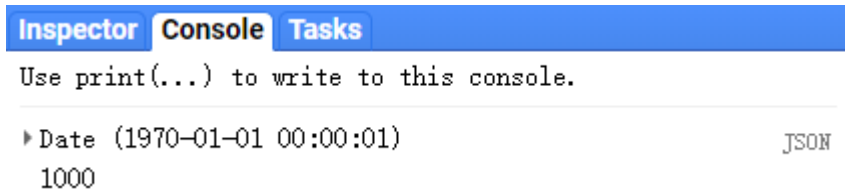
Use print(...) to write to this console.

▶ Date (1999-09-09 00:00:00)	JSON
09-09-1999	JSON

图 9.6 时间的格式更改

下边介绍时间的求毫秒值，代码及执行效果如下：

```
var Date_0 = ee.Date.parse( 'yyyy-MM-dd-HH-mm-ss', '1970-01-01-00-00-01' );  
var Time_Millis = Date_0.millis();  
print( Date_0, Time_Millis );
```



Inspector Console Tasks

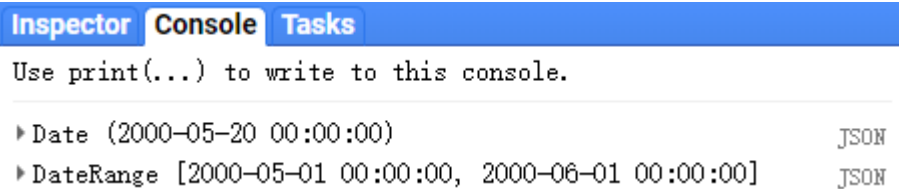
Use print(...) to write to this console.

▶ Date (1970-01-01 00:00:01)	JSON
1000	

图 9.7 时间的求毫秒值

下边介绍时间的范围提取，代码及执行效果如下：

```
var Date_0 = ee.Date( '2000-05-20' );  
var Date_Range = Date_0.getRange( 'month' );  
print( Date_0, Date_Range );
```



Inspector Console Tasks

Use print(...) to write to this console.

▶ Date (2000-05-20 00:00:00)	JSON
▶ DateRange [2000-05-01 00:00:00, 2000-06-01 00:00:00]	JSON

图 9.8 时间的范围提取

下边介绍时间单位值的提取，代码及执行效果如下：

```
var Date_0 = ee.Date( '2000-05-20' );  
var Date_Get = Date_0.get( 'month' );  
print( Date_0, Date_Get );
```

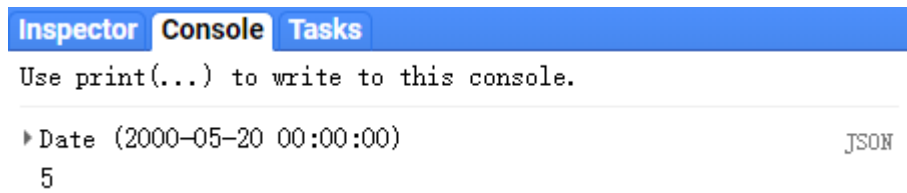


图 9.9 时间单位值的提取

下边介绍时间单位的比例求取，代码及执行效果如下：

```
var Date_0      = ee.Date( '2000-05-20' );  
var Date_Fraction = Date_0.getFraction( 'year' );  
print( Date_0, Date_Fraction, (4*30+20)/365);
```

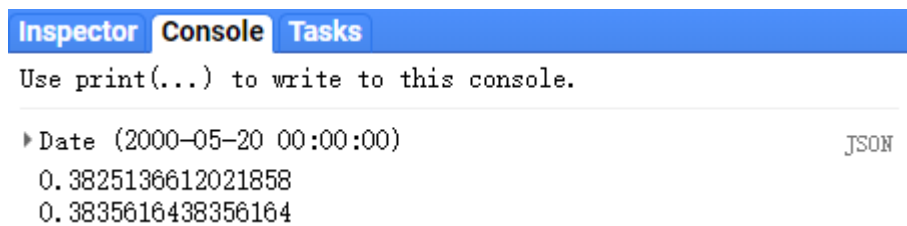


图 9.10 时间单位的比例求取

下边介绍逝去时间的求取，代码及执行效果如下：

```
var Date_0      = ee.Date( '2000-05-20' );  
var Date_Rela_1 = Date_0.getRelative( 'day','month' );  
var Date_Rela_2 = Date_0.getRelative( 'day','year' );  
var Date_Rela_3 = Date_0.getRelative( 'month','year' );  
var Date_Rela_4 = Date_0.getRelative( 'hour','year' );  
  
print( Date_Rela_1, Date_Rela_2, Date_Rela_3, Date_Rela_4);
```

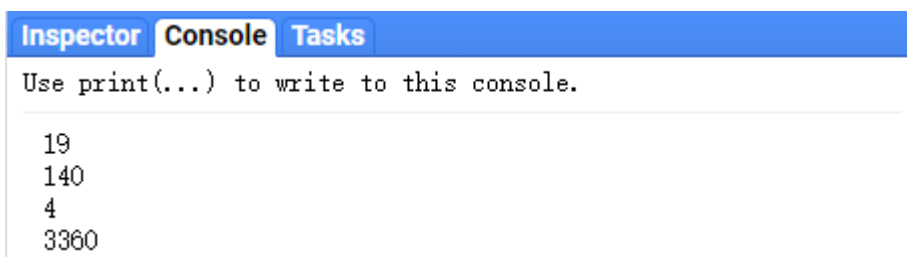


图 9.11 逝去时间的求取

下边介绍时间单位的比值求取，代码及执行效果如下：

```
var Ration = ee.Date.unitRatio( 'day', 'year' );  
print( Ration, 1/365 );
```

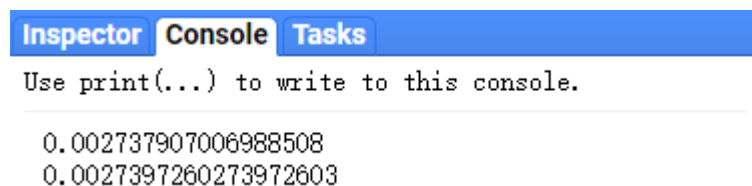


图 9.12 时间单位的比值求取

下边介绍时间差值的求取，代码及执行效果如下：

```

var Date_1 = ee.Date( '2019-01-01' );
var Date_2 = ee.Date( '2000-01-01' );
var Difference_Day = Date_1.difference( Date_2, 'day' );
var Difference_Year = Date_1.difference( Date_2, 'year' );
print( Difference_Day, Difference_Year );

```

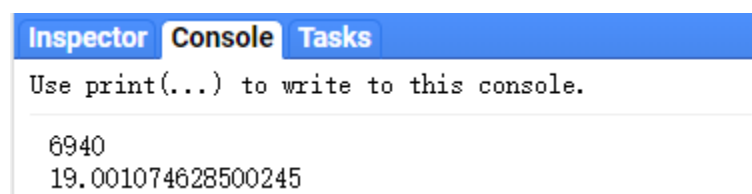


图 9.13 时间差值的求取

下边是本节介绍的常用时间函数，尝试回忆其语法和功能。

ee.Date()	ee.Algorithms.Date()	ee.Date.fromYMD()	ee.Date.parse()
Date.advance()	Date.update()		
Date.format()	Date.milis()	Date.getRange()	
Date.get()	Date.getFraction()	Date.unitRatio()	Date.difference()

9.2 Date Range

时间段也是时间参数的重要组成部分，通常可以利用时间段对遥感图像进行筛选。

下边介绍时间段的创建，代码及执行效果如下：

```

var Range_1 = ee.DateRange('2000-01-01')
var Range_2 = ee.DateRange('2000-01-01','2019-03-28')
print(Range_1,Range_2)

```

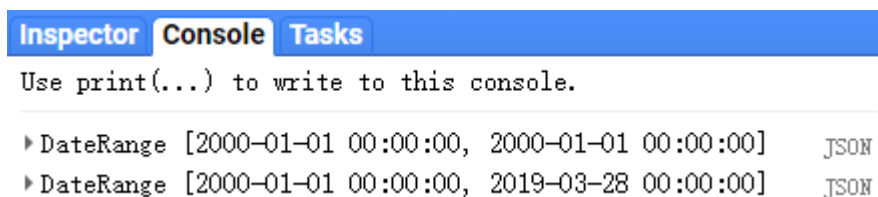


图 9.14 时间段的创建

下边介绍无限时间段的创建，代码及执行效果如下：

```
var Range_1 = ee.DateRange('2000-01-01','2019-01-01')
var Rang_Unbound = ee.DateRange.unbounded()
var test = Rang_Unbound.contains(Range_1)

print(Range_1,Rang_Unbound,test);
```

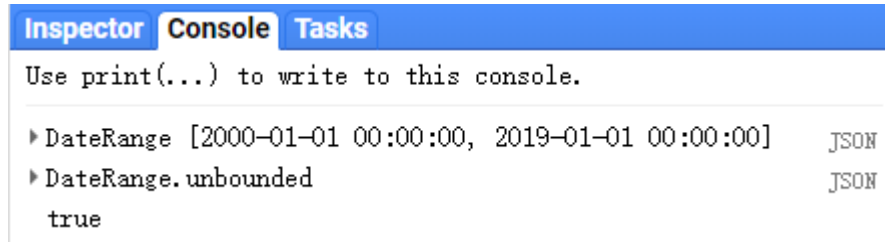


图 9.15 无限时间段的创建

下边介绍时间段的合并，代码及执行效果如下：

```
var Range_1 = ee.DateRange('1979-01-01','2008-01-01')
var Range_2 = ee.DateRange('2000-01-01','2019-03-28')
var Open_Change = Range_1.union(Range_2)
print(Range_1,Range_2,Open_Change)
```

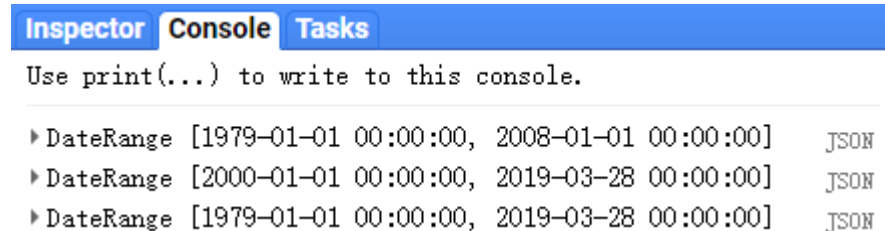


图 9.16 时间段的合并

下边介绍时间段的相交，代码及执行效果如下：

```
var Range_1 = ee.DateRange('1979-01-01','2008-01-01')
var Range_2 = ee.DateRange('2000-01-01','2019-03-28')
var Olympic = Range_1.intersection(Range_2)
print(Range_1,Range_2,Olympic)
```

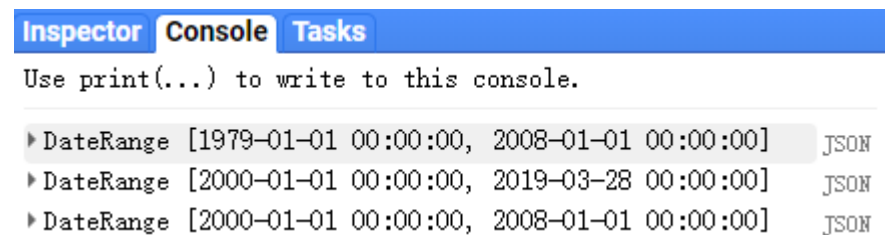


图 9.17 时间段的相交

下边介绍时间段的首位时间求取，代码及执行效果如下：

```
var Range_1 = ee.DateRange('2000-01-01','2019-03-28')
var Range_Start = Range_1.start()
var Range_End = Range_1.end()
print(Range_1,Range_Start,Range_End)
```

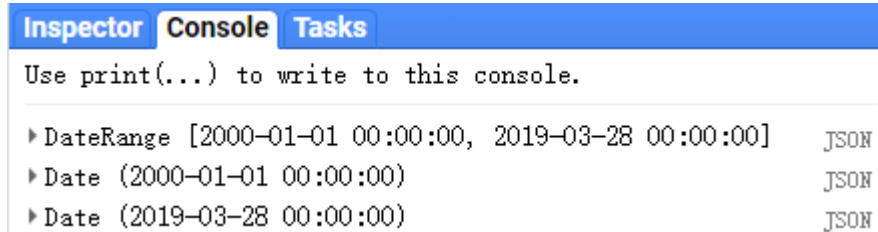


图 9.18 时间段的首位时间求取

下边介绍时间段的内容检测，代码及执行效果如下：

```
var Range_1 = ee.DateRange('1979-01-01','2008-01-01')
var Range_2 = ee.DateRange('2000-01-01','2019-03-28')

var Right_Wrong_1 = Range_1.intersects(Range_2)
var Right_Wrong_2 = Range_1.contains(Range_2)

print(Right_Wrong_1,Right_Wrong_2)
```

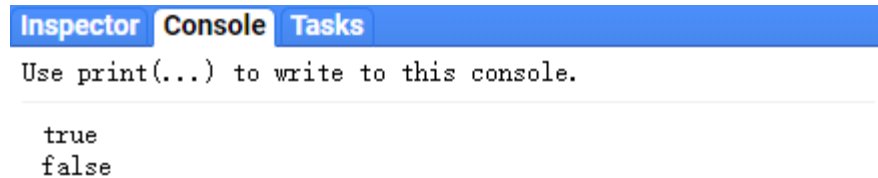


图 9.19 时间段的内容检测

下边介绍时间段的内容判断，代码及执行效果如下：

```
var Range_Empty = ee.DateRange('2019-01-01','2000-01-01');
var Right_Wrong_1 = Range_Empty.isEmpty()
var Right_Wrong_2 = Range_Empty.isUnbounded()

print(Right_Wrong_1,Right_Wrong_2);
```

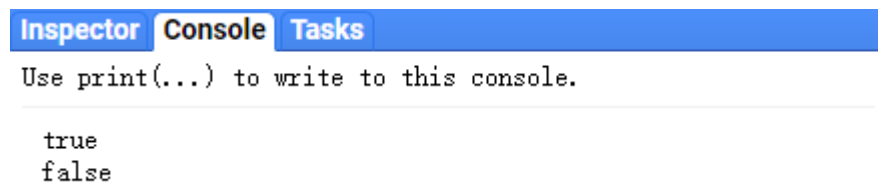


图 9.19 时间段的内容判断

下边是本节介绍的时间段的常见命令，尝试回忆其语法和功能。

<code>ee.DateRange()</code>	<code>ee.DateRange.unbounded()</code>	<code>DateRange.Union()</code>	<code>DateRange.intersection()</code>
<code>DateRange.start()</code>	<code>DateRange.end()</code>	<code>.intersencts()</code>	<code>.contains()</code>
		<code>.inEmpty()</code>	<code>.isUnbounded()</code>



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

本材料由王金柱（西南大学&迪肯大学）创作。如有需要请与我联系。

This doc contributed by Jinzhu Wang of Southwest University & Deakin University.

Email: wangjinzhu1a@gmail.com