

Date Functions and Formatting

First example is this:

Create a logic so that the script should always choose a date that is 3 days ahead of the today's date and make sure that 3 days ahead date is not landing on any weekends i.e., Sat/Sun.

```
function getThreeDaysAhead()
{
var myDate = new Date();
myDate.setDate(myDate.getDate() + 3); //Three days ahead of today's date
var day = myDate.getDay(); //getDay() returns 0-6, 0 for sunday.. , 6 for saturday
log(day);
if (day == 6) {
myDate.setDate(myDate.getDate() + 2); //If saturday , do plus 2 days
} else if (day == 0) {
myDate.setDate(myDate.getDate() + 1); // If Sunday , do plus 1 day
}

return formatDate(myDate);
}
```

```
function getThreeDaysAhead()
{
    var myDate = new Date();
    myDate.setDate(myDate.getDate() + 3); //Three days ahead of today's date
    var day = myDate.getDay(); //getDay() returns 0-6, 0 for sunday.. , 6 for saturday
    log(day);
    if (day == 6) {
        myDate.setDate(myDate.getDate() + 2); //If saturday , do plus 2 days
    } else if (day == 0) {
        myDate.setDate(myDate.getDate() + 1); // If Sunday , do plus 1 day
    }

    return formatDate(myDate);
}
```

Date Formatting functions:

```

function formatDate(date) {
  var d = new Date(date),
  month = '' + (d.getMonth() + 1),
  day = '' + d.getDate(),
  year = d.getFullYear();

  if (month.length < 2)
    month = '0' + month;
  if (day.length < 2)
    day = '0' + day;

  return [month, day, year].join('/');
}

```

Below are some examples using toLocaleDateString();

```

var date2 = new Date(date1).toLocaleDateString("en-US");
log(date2); //will return in US time format
var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };
var date3 = new Date(date1).toLocaleDateString("en-US", options);
log(date3); ////will return in US time format with options as specified above
var date4 = new Date(date1).toLocaleDateString("en-GB", options);
log(date4); //will return in UK time format

```

```

//This is to format the date in mm/dd/yyyy format , ordinary method of formatting
function formatDate(date) {
  var d = new Date(date),
    month = '' + (d.getMonth() + 1),
    day = '' + d.getDate(),
    year = d.getFullYear();

  if (month.length < 2)
    month = '0' + month;
  if (day.length < 2)
    day = '0' + day;

  return [month, day, year].join('/');
}

var date1 = getThreeDaysAhead();
log(date1);
//Below are examples using locales in javascript that will format according to specific region/default locale settings.
var date2 = new Date(date1).toLocaleDateString("en-US");
log(date2); //will return in US time format
var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };
var date3 = new Date(date1).toLocaleDateString("en-US", options);
log(date3); ////will return in US time format with options as specified above
var date4 = new Date(date1).toLocaleDateString("en-GB", options);
log(date4); //will return in UK time format

```

[date.js](#)

some examples from browser console using toLocaleDateString functions:

```

> new Date().toLocaleDateString("en-US");
< '11/17/2021'
> var options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };new Date().toLocaleDateString("en-US", options)
< 'Wednesday, November 17, 2021'
> new Date().toLocaleDateString("en-GB");
< '17/11/2021'
> new Date().toLocaleDateString("ja-JP");
< '2021/11/17'
> new Date().toLocaleDateString("zh-CN");
< '2021/11/17'

```

These functions can be directly used in ds file – customJS as action and function name can be called simply like the normal javascript calling.

File

Options

Edit

Record

Play

Pause

Stop

Step Over

No	Action	Accessor	Value	Description
1	Include	"{ds}/date.js"		
2	Custom JS	getThreeDaysAhead()		
3	Store	date	getThreeDaysAhead()	

Last updated: July 03, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #4

Created 2026-07-31 15:39:55 UTC by Admin

Updated 2026-10-14 14:02:48 UTC by AI_Mananger