How are we doing? Please help us improve Stack Overflow.   **Take our short survey**

What is the best way to initialize a JavaScript Date to midnight?

Asked 8 years, 11 months ago   Active 9 months ago   Viewed 171k times

▲

388

What is the simplest way to obtain an instance of new Date() but set the time at midnight?

`javascript`

▼

★

46

asked Oct 8 '10 at 20:24

Sixty4Bit
5,473 11 41 57

10 Answers

▲

771

▼

✓

The `setHours` method can take optional `minutes` , `seconds` and `ms` arguments, for example:

```
var d = new Date();
d.setHours(0,0,0,0);
```

That will set the time to `00:00:00.000` of *your current timezone*, if you want to work in UTC time, you can use the `setUTCHours` method.

edited Oct 28 '18 at 0:48   answered Oct 8 '10 at 20:31

Greg   **CMS**
7,070 3 23 31   **630k** 167 862 819

---

33   FYI, I'm running this method (d.setHours(0,0,0,0)) in a reduce function on 270K rows, and it's over 20 seconds faster than doing d.setHours(0);
      d.setMinutes(0); d.setSeconds(0); Great answer @CMS! – jbnunn May 9 '12 at 20:05 ✎

---

26   This needs to be done in two lines to keep d as a Date object. I recently fixed a bug that did it in one line: var d = new Date().setHours(0, 0, 0, 0)

works perfectly: `new Date(new Date().setHours(0,0,0,0))` . — SRack Nov 1 '17 at 14:58 ✎

Just wanted to clarify that the snippet from accepted answer gives the **nearest midnight in the past**:

190

```
var d = new Date();
d.setHours(0,0,0,0); // last midnight
```

If you want to get the **nearest midnight in future**, use the following code:

```
var d = new Date();
d.setHours(24,0,0,0); // next midnight
```

edited Jul 5 '18 at 18:28   answered Apr 3 '13 at 14:21

glepretre   Dan
7,777   5   37   55   **36k**   29   100   133

---

6   What happens if today is 25 hours long (clocks adjust forward for Daylight Saving)? – qntm Sep 6 '13 at 16:19

8   This is client side script, midnight is midnight, despite daylight savings.. Also noteworthy mention.. not every place in the world uses DST (daylight savings time) – chris Nov 21 '13 at 19:40

@qntm: daylight saving time change is always applied at 2am-4am, so that it will be only one midnight during that day :). But yes, 3am can happen twice during the day (facepalm) – Dan Nov 22 '13 at 9:39

@chris No, I mean the second snippet. "24 hours after midnight today" is not always the same thing as "midnight tomorrow". In these cases, does the second snippet work? – qntm Nov 23 '13 at 17:18

3   @qntm: Date.setHours does not blindly add 24 hours, it's more intelligent. Instead, it returns a timestamp of the moment, when the clock will show 24 next time. Taking into consideration time saving. Try to play in the console yourself. Interesting, setHours(25) returns the timestamp of the tomorrows 1am – Dan Nov 25 '13 at 13:54

A one-liner for object configs:

53   `new Date(new Date().setHours(0,0,0,0));`

```
dateFieldConfig = {
    name: "mydate",
    value: new Date(new Date().setHours(0, 0, 0, 0)),
}
```

edited Dec 4 '18 at 10:39   answered May 7 '15 at 12:07

Zon
7,285   4   51   62

3   Thanks! i was wondering how i can do it in a single line because i was setting a object property! Great answer. – Paulo Roberto Rosa Aug 31 '17 at
     11:52

     Thanks Zon, this helped me to write it single line. – HungrySoul Jul 8 at 12:13

Just going to add this here because I landed on this page looking for how to do this in moment.js and others may do too.

21   [Rationale: the word "moment" already appears elsewhere on this page so search engines direct here, and moment.js is widespread
         enough to warrant to being covered going on how often it is mentioned in other date-related SO questions]

So, in version 2.0.0 and above:

```
date.startOf('day');
```

For earlier versions:

```
date.sod();
```

Docs:

http://momentjs.com/docs/#/manipulating/start-of/

answered Jun 27 '15 at 22:03

> to get a date object back, use this `moment(DATE_OBJECT).startOf('day').toDate();` – xinthose Jan 29 '16 at 17:37

> 2   yay, lets add a 52kb javascript library to do something you can do in 2 lines. – Bruce Lim Mar 31 '16 at 2:56

> 16   And what if you're already using moment.js for one of the many good reasons you might...? How about reading the rationale I gave before adding a sarcastic comment? – andyhasit Mar 31 '16 at 9:20

You can probably use

8
```
new Date().setUTCHours(0,0,0,0)
```

if you need the value only once.

answered Jan 14 '16 at 12:36

Stan
91 1 2

> This does not obtain an instance of a Date, as the OP required. – cobberboy Apr 28 '16 at 10:59

Adding usefulness to @Dan's example, I had the need to find the next midday or midnight.

4
```
var d = new Date();
if(d.getHours() < 12) {
    d.setHours(12,0,0,0); // next midnight/midday is midday
} else {
    d.setHours(24,0,0,0); // next midnight/midday is midnight
}
```
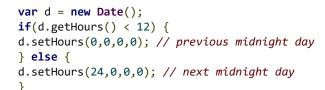
This allowed me to set a frequency cap for an event, only allowing it to happen once in the morning and once in the afternoon for any visitor to my site. The date captured was used to set the expiration of the cookie.

answered Dec 5 '13 at 17:21

If calculating with dates summertime will cause often 1 uur more or one hour less than midnight (CEST). This causes 1 day difference when dates return. So the dates have to round to the nearest midnight. So the code will be (ths to jamisOn):

```
var d = new Date();
if(d.getHours() < 12) {
d.setHours(0,0,0,0); // previous midnight day
} else {
d.setHours(24,0,0,0); // next midnight day
}
```

answered Mar 28 '16 at 11:07

user3887038
100 1 6

In case you already have d3.js as a dependency in your project, or don't mind bringing it in, d3-time (d3.js library is modular as of v4.0.0) has got Intervals.

They might prove useful when setting dates to "default" values, e.g. midnight, 0.00 seconds, the first of the month, etc.

```
var d = new Date(); // Wed Aug 02 2017 15:01:07 GMT+0200 (CEST)
d3.timeHour(d) // Wed Aug 02 2017 00:00:00 GMT+0200 (CEST)
d3.timeMonth(d) // Tue Aug 01 2017 00:00:00 GMT+0200 (CEST)
```

answered Aug 2 '17 at 13:06

Peemster
71 1 8

So I've noticed a trend in JavaScript related questions posted on SO: no matter the complexity of the solutions put forward using native JS, any answers addressing the issue with the help of an external library will inevitably at some point in time get downvoted. Realizing this, I took care to post my answer with the quick preface: *In case you already have example.js as a dependency*. And yet still the downvote, anonymous SO user? that hurts, man :'- (– Peemster Jan 10 '18 at 14:57

I fell that this answer should not be down voted especially since the one proposing usage of moment.js has 16 up votes (at the moment of writing this

I have made a couple prototypes to handle this for me.

2

```
// This is a safety check to make sure the prototype is not already defined.
Function.prototype.method = function (name, func) {
    if (!this.prototype[name]) {
        this.prototype[name] = func;
        return this;
    }
};

Date.method('endOfDay', function () {
    var date = new Date(this);
    date.setHours(23, 59, 59, 999);
    return date;
});

Date.method('startOfDay', function () {
    var date = new Date(this);
    date.setHours(0, 0, 0, 0);
    return date;
});
```

if you dont want the saftey check, then you can just use

```
Date.prototype.startOfDay = function(){
  /*Method body here*/
};
```

Example usage:

```
var date = new Date($.now()); // $.now() requires jQuery
console.log('startOfDay: ' + date.startOfDay());
console.log('endOfDay: ' + date.endOfDay());
```

edited Jun 5 '18 at 14:39   answered Jun 5 '18 at 14:29

Mike

559   3   7

-4

▼

```
                var a = new Array();
                                    var b = new Array();

                                    var c = new Array();
                                    var date = new Date().toJSON();
                                    a = date.split("T");
                                    a = a[0];
                                    b = a.split("-");
                                    var currentDate = new Date(b[0],b[1] , b[2] -29
,-19,30,0).toJSON();

                                    c = currentDate.split(".");
                                    var newDate = c[0];

console.log(newDate);
```

answered Nov 3 '16 at 6:59

Gopal Anand
27 1   10

2   Kudos for creativity :D – Bryant Jackson Jan 9 '18 at 14:27