turn('page', 4);

API DOCUMENTATION

turn.js 4th release

www.turnjs.com

©2012 Emmanuel Garcia – All rights reserved

The **turn.js** API was conveniently built as a jQuery plugin, it provides access to a set of features and allows you to define the user interaction.

The API includes properties, methods and events. Most setter and getter functions have the same name.

Essentials

- jQuery 1.7 or later

Browser Support

- Safari for iOS (iPad, iPhone, iPod)
- Safari 5
- Chrome 11
- Chrome for Android
- Firefox 9
- IE 9
- IE 8 Using turn.html4.js

4th Release - Commercial Version

+ Added option **autoCenter**
+ Added option **zoom**
+ Added property **animating**
+ Added property **zoom**
+ Added method **center**
+ Added method **destroy**
+ Added method **is**
+ Added method **zoom**
+ Added event **missing**
+ Added event **zooming**
+ Added class **.even**
+ Added class **.fixed**
+ Added class **.hard**
+ Added class **.odd**
+ Added class **.own-size**
+ Added class **.sheet**
+ Added class **.sheet**
+ Added the **ignore** attribute
+ New **turn.html4.js**
+ New **scissors.js**
+ Changed the class **.turn-page** to **.page**
+ Improved the animation frame generator with requestAnimationFrame
+ Improved the animation speed for hard pages with CSS3 transitions
+ Redesigned the event sequence to listen to only three events
+ Fixed issue #79
+ Fixed issue #91
+ Fixed issue about the event order turning + turned
+ Fixed issue about appending pages in wrong locations

Preparing the HTML

Turn.js uses a DOM element as a container for all the pages. This DOM element is typically the flipbook. There are three ways to add pages to your flipbook:

1. **Adding pages directly in the container.** The HTML contains all the content of your flipbook. For example:

```
<div id="flipbook">
   <div>Page 1</div>
   <div>Page 2</div>
   <div>Page 3</div>
   <div>Page 4</div>
</div>
```

2. **Adding pages dynamically through the API.** The HTML contains only the container and all the pages will be dynamically added after getting the data through an Ajax Request. For example:

```
<div id="flipbook">
   <!– Load the content here-->
</div>
```

3. **A combination of both.** Sometimes it's a better idea to have some pages in the HTML and another loaded dynamically. For example:

```
<div id="flipbook">
   <div>Front1</div>
   <div>Front 2</div>
   <!– Load the content here-->
   <div>Back 2</div>
   <div>Back 1</div>
</div>
```

Performance

Turn.js can work with very long flipbooks. That is, there's no limit regarding the amount of pages that it can contain. This is because turn.js only keeps in DOM the last 6 pages no matter how long the book is. There will still be references to the content that was previously loaded, but it's possible to reduce the cache in order to release memory.

IE8/7 Support

The turn.js library includes turn.html4.js, which is optimized for IE 8 as well as any other browsers with no support of CSS3 features such as transformation. You can use CSS tricks to include turn.html4.js for older versions of IE, but it's recommended to use the Modernizr library and a resource loader like YepNope.js both available on modernizr.com. For instance the following JavaScript code:

```
yepnope({
     test : Modernizr.csstransforms,
     yep: ['lib/turn.js', 'css/flipbook-html5.css']
     nope : ['lib/turn.html4.js', 'css/flipbook-html4.css']
});
```

CSS Classes

Turn.js uses **classes** to define the way a page or the flipbook as a whole should look. The classes also allow you to add CSS rules or use them as a selector for that page.

List of classes

1. even

This class describes even pages when display is set to double. For example: Page 2, Page 4, and so on. Notice that even pages will always be in the right of the flipbook.

2. fixed

Indicates that a page should not be removed from the DOM even when the page is out of range.

3. hard

Sets a hard transition effect for a page.

By setting all the pages as hard you will create the same effect as in Flipboard.

For example:

```
<div id="flipbook">
     <div class="hard">Page 1</div>
     <div class="hard">Page 2</div>
</div>
```

4. odd

This class describes odd pages when display is set to double. For example: Page 1, Page 3, and so on. Notice that odd pages will always be in the left of the flipbook.

5. own-size

Customizes the size of a page. The size rules can be added directly to the page selector or using the style parameter. For example:

```
<div id="flipbook">
     <div>Page 1 with default size</div>
     <div class="own-size" style="width:100;
          height:100px;">
          Page 2 with own size
     </div>
</div>
```

6. page

This class describes every page. It provides a unique subclass for all the pages no matter its number.
This subclass allows you to set the size of all the pages:

```
.flipbook{
     width:800px;
     height:600px;
}

.flipbook .page{
     width:400px;
     height:600px;
}
```

7. p[0-9]+

This class describes a particular page. For instance, .p1 refers to the first page, .p2 to the second page and so on.
You can change the number of a page no matter its order in the HTML:

```
<div id="flipbook">
     <div class="p100">Page 100</div>
```

```
    <div class="p1">Page 1</div>
</div>
```

You can also use it as a jQuery selector:

```
$('#flipbook .p100').doSomething();
```

8. shadow

This class describes the visible area of the flipbook. That is, because it surrounds the flipbook, it's suitable for a shadow around the flipbook. For example:

```
.flipbook .shadow{
    box-shadow: 0 4px 10px #666;
}
```

9. sheet

Makes the page looks like a sheet of paper. This is the default style for all the pages.

Setting the size of the flipbook

There're three ways to set the size of a flipbook.

1) Using CSS, for example:

```
.flipbook{
    width:800px;
    height:600px;
}

.flipbook .page{
    width:400px;
    height:600px;
}
```

Notice that the width of the page is half the size of the flipbook.

2) **Using options, for example:**

```
$('#flipbook').turn({width:800, height:600});
```

3) **Using the size method, for example:**

```
$('#flipbook').turn('size', 800, 600);
```

Display

The display defines how many pages are visible in the flipbook. While using turn.js on an iPad or iPhone, there would be some problems to turn pages if the orientation of the device is portrait. For that reason, turn.js introduces a new view called single. There are two views: double, which shows two pages and single, which shows only one page.

Views

A view is a set of pages that are visible on the screen; in general that moment depends on the current page. For example, when the display of a flipbook of 10 pages is set to *double*, the pages would be grouped like this:

1
2-3
4-5
6-7
8-9
10

This flipbook has 6 views. The general relation is: *totalPages/2 +1* Therefore, if the current page is 5, the view in double display would be: *[4,5]*

Using display single, the view will always have only one page. So, there will be the same number of pages and views.

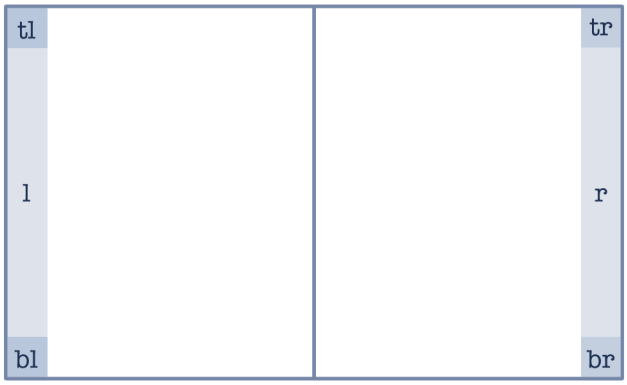
Ignoring elements in the flipbook

Turn.js reserves an HTML attribute called *ignore* which can be added to some elements in the flipbook to not select them as pages. For example:

```
<div id="flipbook">
     <div ignore="1"> Something else </div>
     <div>Page 1</div>
     <div>Page 2</div>
     <div ignore="1"> Something else </div>
     <div>Page 3</div>
</div>
```

Corners

The corners identify every interactive region on the flipbook. For example:



Description:

tl: Top Left.
tr: Top right.
r: Right (hard pages only)
br: Bottom right.
bl: Bottom left.
l: Left (hard pages only)

Constructor

The constructor is the function that will turn the container into a flipbook. Because turn.js uses a single instance, it's not necessary to have an external reference to it.

Example:

```
$('#flipbook').turn([options]);
```

Notice that the constructor only has one argument called options and it's optional.

Options

The options define characteristics of the flipbook. All the keys are optional.

Option	Type	Default value	Description
acceleration	Boolean	true	Sets the hardware acceleration mode, for touch devices this value must be true.

autoCenter	Boolean	false	Centers the flipbook depending on how many pages are visible.
display	String	single	Sets the display mode. Values: single, double
duration	Number	600	Sets the duration of the transition in milliseconds
gradients	Number	true	Shows gradients and shadows during the transition.
height	Number	Height of the selector	Sets the height of the selector
inclination	Number	0	Sets the inclination of the page during the transition
page	Number	1	Sets the first page
pages	Number	The number of pages in the DOM	Sets the number of pages
when	Object	{}	Sets event listeners
width	Number	Width of the selector	Sets the width of the page

Adding options to the constructor:

```
$('#flipbook').turn({
      display: 'double',
      inclination: 50,
      when: {
            turned: function(event, page, pageObj) {
```

```
                alert('the current page is ' + page);
            }
        }
});
```

Properties

Syntax:

```
$('#flipbook').turn('propertyName');
```

List of properties

- **animating**

  Returns true when animating a page.

  ```
  $('#flipbook').turn('animation');
  ```

- **display**

  Gets the current display. It can be *single* or *double*.

  ```
  $('#flipbook').turn('display');
  ```

- **page**

  Gets the current page.

  ```
  $('#flipbook').turn('page');
  ```

- **pages**

  Gets the number of pages within the flipbook

  ```
  $('#flipbook').turn('pages');
  ```

- **size**

  Gets the size of the flipbook. It would be an object with two keys, width and height.

  ```
  $('#flipbook').turn('size');
  ```

- **view**

  Gets the current view.

  ```
  $('#flipbook').turn('view');
  ```

- **zoom**

  Gets the current zoom. The default value is 1.

  ```
  $('#flipbook').turn('zoom');
  ```

Methods

Syntax:

```
$('#flipbook').turn('method name'[, argument1, argument2]);
```

When a method doesn't return a value, it can be connected to another methods, for example:

```
$('#flipbook').turn('method1').turn('method2');
```

List of methods

- **addPage**

  Adds a page to the flipbook.

Parameter	Type	Description
element	jQuery element	DOM element for the page.
pageNumber	Number	Page Number This parameter is optional and the default value is: `$('#flipbook').turn('pages')+1`

  For example:

```
element = $('<div />', {class: 'p10'});
$('#flipbook').turn('addPage', element);
```

The above code is equivalent to:

```
element = $('<div />');
$('#flipbook').turn('addPage', element, 10);
```

- **display**

  Set the display.

Parameter	Type	Description
displayMode	String	It can be *single* or *double*. Single means one page per view, meanwhile double two pages per view.

```
$('#flipbook').turn('display', 'single');
```

- **disable**

  Disables and enables the effect. If it's disabled, users won't be able to change the current page.

Parameter	Type	Description
disable	Boolean	True to disable the effect or false to enable.

```
$('#flipbook').turn('display', 'single');
```

- **destroy**

  Destroys the flipbook. That is, it removes all the pages from the DOM and memory.

For example:

```
$('#flipbook').turn('destroy');
$('#flipbook').turn('page', 1);
```

The last line will throw an error. You can also remove the container, for example:

```
$('#flipbook').turn('destroy').remove();
```

- **hasPage**

  Returns true if a page is in memory.

Parameter	Type	Description
pageNumber	Number	Page number.

  For example:

```
if ($('#flipbook').turn('hasPage', 1)) {
     alert('Page 1 is already in the flipbook');
}
```

- **next**

  Turns the view to the next one. For example:

```
$('#flipbook').turn('next');
```

- **is**

  Detects if a selector has an instance of turn.js. For example:

```
if (!$('#flipbook').turn('is')) {
     // Create a new flipbook
     $('#flipbook').turn();
}
```

- **page**

  Turns the page.

Parameter	Type	Description
page	Number	Page number.

  For example, the following example will turn the page to 10.

```
$('#flipbook').turn('page', 10);
```

- **pages**

  Sets the number of pages that the flipbook has. If the number of pages is less than the current one, it will remove the pages out of range.

Parameter	Type	Description
pages	Number	Number of pages.

  For example:

```
$('#flipbook').turn('hasPage', 10); // It's true.

$('#flipbook').turn('pages', 5); // Sets 5 pages

$('#flipbook').turn('hasPage', 10); // Returns false
```

- **peel**

  Shows a peeling corner.

Parameter	Type	Description
corner	String	Corner type. The corners can be: tl, tr, bl, br, r, l.

  For example:

  ```
  // To show the br corner

  $('#flipbook').turn('peel', 'br');

  // To hide all the corners

  $('#flipbook').turn('peel', false);
  ```

- **previous**

  Turns the view to the previous one. For example:

  ```
  $('#flipbook').turn('previous');
  ```

- **range**

  It returns an array of two values where the first element refers to a page from which next pages should be contained in DOM. The second element refers to the last page of the range. That is, the current range always has the following relationship: *range[0] <= current page <= range[1]*

Parameter	Type	Description
pageNumber	Number	A page number within a range. This parameter is optional and the default value is `$('#flipbook').turn('page');`

For example, in order to add new pages dynamically, it's necessary to use the range method:

```
var range = $('#flipbook').turn('range', 10);

for (var page = range[0]; page<=range[1]; page++){
     if (!$('#flipbook').turn('hasPage', page)) {
          $('#flipbook').turn('addPage',
               $('<div />'), page);
     }
}
```

The last example will add the pages that are closest to the page 10. Assuming that display is double, those pages would be [8, 9, 10, 11, 12, 13].

- **removePage**

  Removes a page from the DOM and all its references.

Parameter	Type	Description
pageNumber	Number	Number of the page to remove.

For example:

```
$('#flipbook').turn('removePage', 10);
```

- **resize**

  Recalculate the position of all the pages.

  ```
  $('#flipbook').turn('resize');
  ```

- **size**

  Sets the size of the flipbook.

Parameter	Type	Description
width	Number	New width for the flipbook.
height	Number	New height for the flipbook.

  For example:

  ```
  $('#flipbook').turn('size', 1000, 600);
  ```

- **stop**

  Stop the current animation. For example, it's possible to turn to a page without having animation.

  ```
  $('#flipbook').turn('page', 10).turn('stop');
  ```

- **zoom**

  Increases or reduces the size of the flipbook.

Parameter	Type	Description
factor	Number	Factor of multiplication. For example, 2 would increase in twice the size of the flipbook; meanwhile 0.5 would reduce the size to half of its current size.
duration	Number	Duration in milliseconds of the scaling animation. The default value is 500.

For example, to zoom out without animation:

```
$('#flipbook').turn('zoom', 0.5, 0);
```

Events

The events allow you to define behaviors to specific moments. It's possible to define events in two different ways:

1. **Using the *when* key of options**

   While adding event listeners, it will require to add the listeners before the constructor creates the flipbook. For example:

```
$('#flipbook').turn({
      when: {
            turning: function(event, page, pageObject) {

            }
      }
});
```

2. Using *bind*

jQuery provides a *bind* function in order to add listeners to elements. You can use *bind* to add as many listener as you need for an event. For example:

```
$('#flipbook').bind('turning',
     function(event, page, obj){
          alert('Page '+ page);
});
```

Using the event object

The first argument that all the listener functions share is the event object, which allows you to manipulate the propagation and default action of the event.

Some events are followed by an action that can be, for instance, to turn the page. Therefore, it's possible to prevent that action by using `event.preventDefault();` within the event function. It's not necessary to return *false* to prevent the default action.

List of events

- **end**

  This event is triggered after ending the motion of a page.

Parameter	Type	Description
event	Event	Event object.

page	Number	The page number
pageObject	Object	The page data

- **first**

  This event is triggered when the current page is 1.

Parameter	Type	Description
event	Event	Event object.

- **last**

  This event is triggered when the current page is `$('#flipbook').turn('pages')`. That is, the last page.

Parameter	Type	Description
event	Event	Event object.

- **missing**

  This event is triggered when some pages are required in the current range.

Parameter	Type	Description
event	Event	Event object.
pages	Array	Pages that must be added.

You can use this event to add pages through *addPage*. For example:

```
$('#flipbook').bind('missing', function(event, pages){
     for (var i = 0; i < pages.length; i++) {
          $(this).turn('addPage',
               $('<div />'), pages[i]);
     }
});
```

- **start**

This event is triggered before starting the motion of a page.

Parameter	Type	Description
event	Event	The event object. The default action is to start the animation. Preventing the default action, there wouldn't be interaction with any corner.
pageObject	Object	The page object.
corner	String	Corner Type. The corners can be: tl, tr, bl, br.

For instance, if you want to allow only corners at the bottom of the page, you can use the *start* event:

```
$('#flipbook').bind('start',
     function(event, pageObject, corner){
          if (corner=='tl' || corner=='tr') {
               event.preventDefault();
          }
});
```

You can also use the *start* event to change the next page of the current page:

```
$('#flipbook').bind('start',
     function(event, pageObject, corner){
          if (pageObject.page==1) {
               // pageObject.next of the 1st page is 2,
               // but let's change it:
               pageObject.next = 4;
          }
});
```

- **turning**

  This event is triggered before the flipbook turns the page.

Parameter	Type	Description
event	Event	The event object. The default action is to allow the flipbook to turn to a page.
page	Number	The new page number
view	Array	The new view

- **turned**

  This event is triggered after the flipbook turned the page.

Parameter	Type	Description
event	Event	The event object.
page	Number	The new page number
view	Array	The new view

- **zooming**

  This event is triggered when the zoom factor is changed.

Parameter	Type	Description
event	Event	The event object. The default action is to zoom.
newFactor	Number	The new zoom factor
current	Number	The current zoom factor