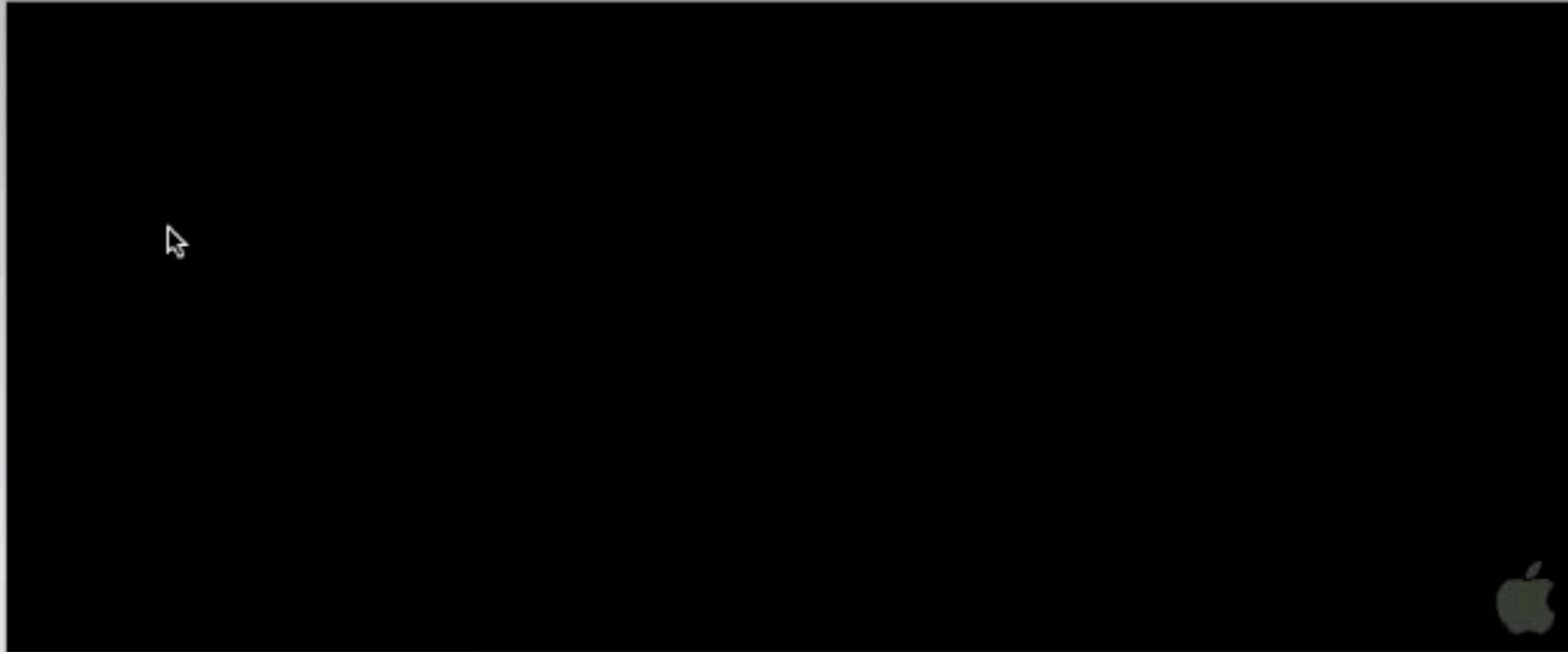


Video



Try it out: Use the control panel to the right to scale up or down or add a mask while the video is playing.

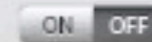
About this demo: The HTML5 video tag allows you to integrate video within your website's code. And Safari offers HTTP streaming, so playback quality dynamically adjusts to the available speed of wired or wireless networks — perfect for viewing on mobile devices such as iPad, iPhone, and iPod touch.

Tron Legacy © 2010 Disney Enterprises Inc. All rights reserved.

Scale



Mask



Perspective



[Visit the Safari Dev Center to learn more and download sample code](#)



Medientechnik

jQuery

Themen heute

- Wiederholung
- jQuery
- AJAX

Wiederholung

\$() - Die jQuery-Funktion

- `$("selektor").aufgabe();`
- selektor: Vor allem die aus CSS bekannten Selektoren
- `$("a")` - gilt für alle Links im DOM. Erzeugt intern ein Objekt
- Dieses Objekt kann mit mehreren Funktionen verbunden werden.
- `$("a").aufgabe();` erzeugt dann ein weiteres Objekt usw:
- `$("a").aufgabe().aufgabe2();`

Beispiele

```
$("#a.hideMe").click(function(){  
  
    $(this).hide("slow");  
  
    return false;  
  
});
```

this bezieht sich, analog zu Java, auf das aktuelle Objekt. In diesem Fall der angeklickte Link mit der Klasse hideMe.

Beispiele

```
$("#a").click(function(){
```

```
    $(this).addClass("test").html("hier wurde schon geklickt");
```

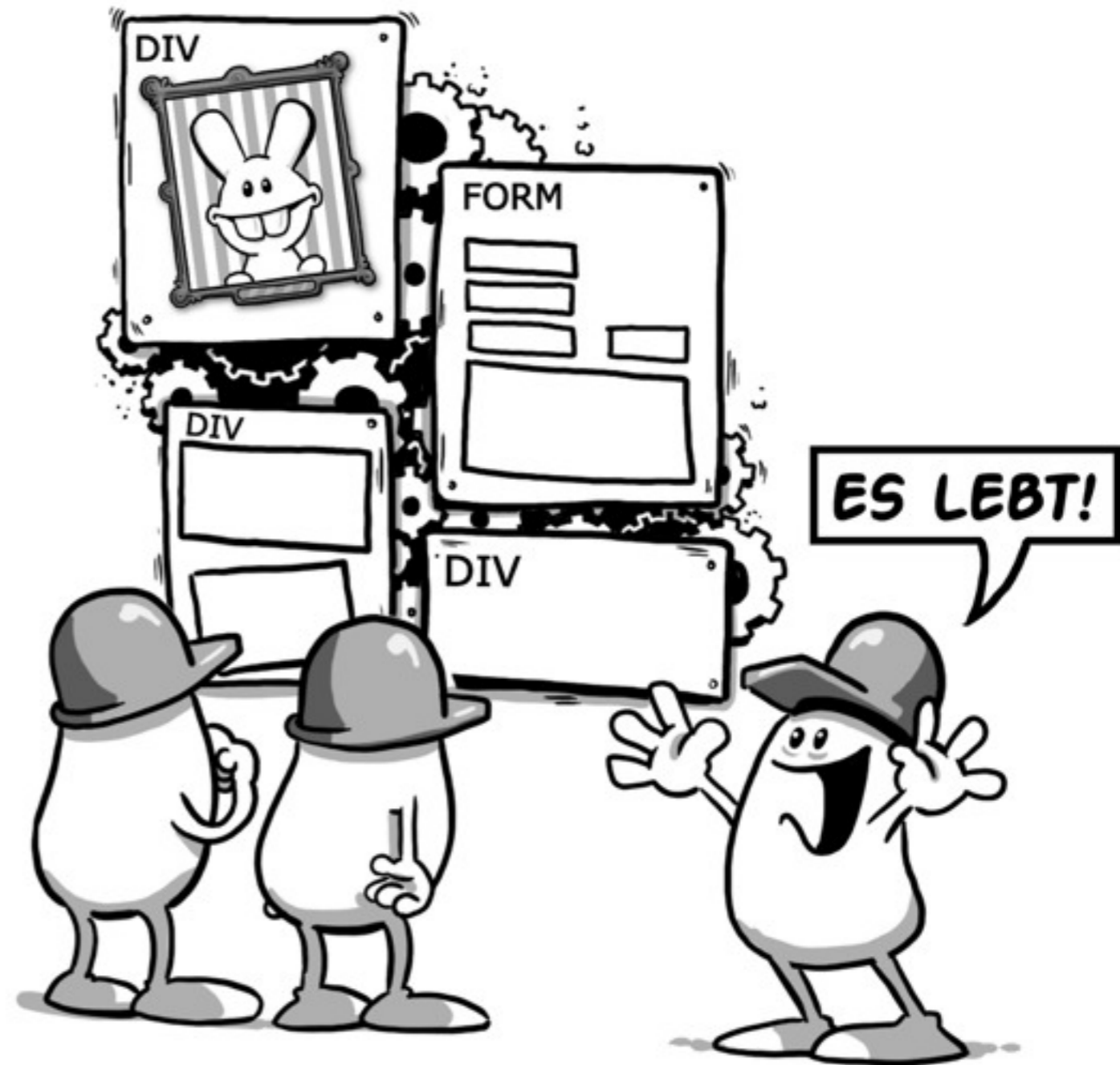
```
    return false;
```

```
});
```

Fragen?



jQuery



Document ready

- Wofür wird die document ready Methode benötigt?

```
$(document).ready(function(){
```

```
});
```

Document ready

- DOM muss erst vollständig geladen sein damit jQuery ausgeführt wird
- Hintergrund: Vorher kann man mit jQuery die Seite nicht verändern (das DOM manipulieren)

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

jQuery einbinden

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown(
    });
  });
</script>
```

ready-Event: Die gebundene Funktion wird ausgeführt, sobald das DOM fertig aufgebaut ist

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      // ...
    });
  });
</script>
```

Woran soll die Funktion gebunden werden? Hier an Elemente mit der Klasse „button“


```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#foot
    });
  });
</script>
```

Welches Event löst die gebundene Funktion aus?

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

Das Element das verändert wird,
wenn das click-Event eintritt

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

```
<script type="text/javascript">
<script type="text/javascript">
    $(document).ready(function() {
        $('.button').click(function() {
            $('#footer').slideDown('slow');
        });
    });
</script>
```

Was passiert mit dem Element? In diesem Fall wird slideDown mit dem Parameter ‚slow‘ ausgeführt.

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('.button').click(function(){
      $('#footer').slideDown('slow');
    });
  });
</script>
```

Events

- Möglichkeit bei einem Ereignis die Seite zu verändern.
- Verfügbare Events sind u.a.:
 - click, hover
 - keyup, keydown
 - submit
 - ...
- Aufbau normalerweise:

```
$("#selektor").event(function(){  
    // magic  
});
```


Events - delegate()

```
$("#body").delegate("p", "click", function(){  
    $(this).after("<p>Another paragraph!</p>");  
});
```

Entspricht...

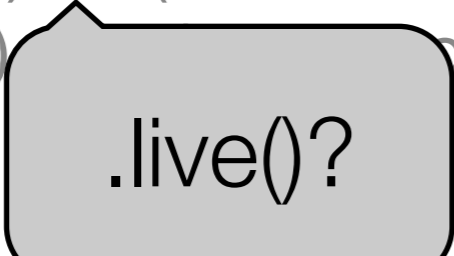
```
$("#body").each(function(){  
    $("p", this).live("click", function(){  
        $(this).after("<p>Another paragraph!</p>");  
    });  
});
```

Events - delegate()

```
$("#body").delegate("p", "click", function(){  
    $(this).after("<p>Another paragraph!</p>");  
});
```

Entspricht...

```
$("#body").each(function(){  
    $("p", this).live("click", function(){  
        $(this).after("<p>Another paragraph!</p>");  
    });  
});
```



.live()?

Selektoren

```
<span class="wir">Wir</span> <span class="werden">werden</span> <span  
class="weltmeister">Weltmeister</span>
```

```
$("#span.wir").mouseover(function(){  
    $(this).css("color","black");  
});
```

```
$("#span.werden").mouseover(function(){  
    $(this).css("color","red");  
});
```

```
$("#span.weltmeister").mouseover(function(){  
    $(this).css("color","gold");  
});
```

Danke an Miriam, Bianca und Indre

nth-child Selektor

```
<p><span>Wir</span> <span>werden</span> <span>Weltmeister</span></p>
```

```
$("#p:nth-child(1)").mouseover(function(){  
    $(this).css("color", "black");  
});
```

```
$("#p:nth-child(2)").mouseover(function(){  
    $(this).css("color", "red");  
});
```

```
$("#p:nth-child(3)").mouseover(function(){  
    $(this).css("color", "gold");  
});
```

nth-child Selektor

```
<p><span>Wir</span> <span>werden</span> <span>Weltmeister</span></p>
```

```
$("#p:nth-child(1)").mouseover(function(){  
    $(this).css("color","black");  
});
```

```
$("#p:nth-child(2)").mouseover(function(){  
    $(this).css("color","red");  
});
```

```
$("#p:nth-child(3)").mouseover(function(){  
    $(this).css("color","gold");  
});
```

jQuery zählt bei nth-child ab 1 und nicht ab 0!

.filter()

```
$('#a.click').click(function(){  
    alert('ah ich wurde geklickt');  
});
```

```
$('#a.hide').click(function(){  
    $(this).hide();  
    return false;  
});
```

```
$('#a.error').hover(function(){  
    $(this).css('background','red');  
});
```

```
$("#a")  
    .filter(".click")  
    .click(function(){  
        alert('ah ich wurde geklickt');  
    }).end()  
    .filter(".hide")  
    .click(function(){  
        $(this).hide();  
        return false;  
    }).end()  
    .filter(".error")  
    .hover(function(){  
        $(this)  
            .css('background','red');  
    })  
    .end();
```

filter()

- `$('#li').filter(':even').css('background-color', 'red');`
- `$('#li').filter(function(index) {
 return $('#strong', this).length == 1;
}).css('background-color', 'red');`

filter()

- `$('li').filter(':even').css('background-color', 'red');`
- `$('li').filter(function(index) {
 return $('strong', this).length == 1;
}).css('background-color', 'red');`

Gilt für die li-
Elemente an der
Position 1, 3, 5, ...

filter()

- `$('#li').filter(':even').css('background-color', 'red');`
- `$('#li').filter(function(index) {
 return $('#strong', this).length == 1;
}).css('background-color', 'red');`

filter()

- `$('#li').filter(':even').css('background-color', 'red');`
- `$('#li').filter(function(index) {
 return $('#strong', this).length == 1;
}).css('background-color', 'red');`

Fortgeschritten: Filter-Funktion liefert das erste li-Element das genau ein strong-Element beinhaltet.

filter()

- `$('#li').filter(':even').css('background-color', 'red');`
- `$('#li').filter(function(index) {
 return $('#strong', this).length == 1;
}).css('background-color', 'red');`

nth-child Selektor + filter()

```
<p><span>Wir</span> <span>werden</span> <span>Weltmeister</span></p>
```

```
$("#p")  
  .filter(":nth-child(1)")  
    .mouseover(function(){  
      $(this).css("color","black");  
    })  
  .end()  
  .filter(":nth-child(2)")  
    .mouseover(function(){  
      $(this).css("color","red");  
    })  
  .end()  
  .filter(":nth-child(3)")  
    .mouseover(function(){  
      $(this).css("color","gold");  
    })  
  .end();
```

.live()

- Problem: Elemente, die von jQuery neu ins DOM eingefügt werden lassen sich nicht ohne weiteres mit jQuery manipulieren.
- Lösung: .live() bindet eine Funktion an alle Elemente auch wenn sie erst später eingefügt werden.

Beispiel

HTML:

```
<body>  
  <div class="clickme">Click here</div>  
</body>
```

jQuery:

```
$('#body').append('<div class="clickme">Another target</div>');
```

```
$('.clickme').click(function() {  
  // magic  
});
```

Besser

HTML:

```
<body>  
  <div class="clickme">Click here</div>  
</body>
```

jQuery:

```
$('#body').append('<div class="clickme">Another target</div>');
```

```
$('.clickme').live('click', function() {  
  // magic (working)  
});
```

Fragen?



Animationen

- Einfache Animationen erstellt man u.a. mit
 - fadeIn()
 - fadeOut()
 - hide()
 - show()
 - slideDown()
 - slideUp()

Eigene Animationen: .animate()

- .animate(properties, [duration], [easing], [callback])
- properties: CSS
- duration: Dauer in ms
- easing: Übergang: linear oder swing (Standard)
- callback: Methode, die an diese Animation gebunden wird

.animate() - Beispiel

```
$("#go").click(function(){  
  $(".block").animate({  
    width: "70%",  
    opacity: 0.4,  
    marginLeft: "0.6in",  
    fontSize: "3em",  
    borderWidth: "10px"  
  }, 1500 );  
});
```

```
$("#right").click(function(){  
  $(".block").animate({"left": "+=50px"}, "slow");  
});
```

Fragen?



AJAX

AJAX

- AJAX = Asynchronus JavaScript and XML
- asynchrone Datenübertragung zwischen Browser und Server
- HTTP-Anfragen während die Seite angezeigt wird ohne diese neu laden zu müssen.
- Nachteile:
 - kein „Zurück“-Button
 - ohne Framework nur umständlich umsetzbar (Kompatibilitätsprobleme)
 - Polling: „Sind wir schon da?“ - „Nein“ - „Sind wir schon da?“...
 - Keine Lesezeichen
 - ...

.ajax()

- `jQuery.ajax(settings);`
- `settings` u.a.:
 - `async`
 - `complete()`
 - `error`
 - `type`
- `url`
- `...`

.ajax() - Beispiel

```
$.ajax({  
  url: 'ajax/test.html',  
  success: function(data) {  
    $('result').html(data);  
    alert('Load was performed.');  }  
});
```


JSON

- JavaScript Object Notation
- <http://www.json.org/>
- Grundidee wie XML allerdings wesentlich schlanker (keine .xsd)
- Im Web oft beliebter als XML

Aufbau

- Objekt: beginnt mit { und endet mit }
- Eigenschaften bestehen aus Schlüssel und Wert getrennt von einem Doppelpunkt: "Kreditkarte" : "Xema".
- Ein Wert kann Objekt, Zeichenkette oder Array sein
- Eigenschaften werden mit Komma getrennt
- Arrays beginnen mit [und enden mit]
- Boolesche Werte haben die Bezeichnung true oder false

Beispiel

```
{
  "Kreditkarte" : "Xema",
  "Nummer"      : "1234-5678-9012-3456",
  "Inhaber"     : {
    "Name"      : "Reich",
    "Vorname"   : "Rainer",
    "Geschlecht" : "männlich",
    "Vorlieben" : [ "Reiten", "Schwimmen", "Lesen" ],
    "Alter"     : null
  },
  "Deckung"     : 2e+6,
  "Währung"     : "EURO"
}
```

jQuery & JSON

- `jQuery.getJSON(url, [data], [callback(data, textStatus)])`
- `test.json`:

```
{  
  "foo": "The quick brown fox jumps over the lazy dog.",  
  "bar": "ABCDEFGFG",  
  "baz": [52, 97]  
}
```
- `jQuery`:

```
$.getJSON('ajax/test.json', function(data) {  
  $(' .result').html('<p>' + data.foo + '</p>' +  
    + '<p>' + data.baz[1] + '</p>');  
});
```

jQuery & JSON

- `jQuery.getJSON(url, [data], [callback(data, textStatus)])`

- `test.json:`

```
{  
  "foo": "The quick brown fox jumps over the lazy dog.",  
  "bar": "ABCDEFGFG",  
  "baz": [52, 97]  
}
```

- `jQuery:`

```
$.getJSON('ajax/test.json', function(data) {  
  $('result').html('<p>' + data.foo + '</p>'  
    + '<p>' + data.baz[1] + '</p>');  
});
```

```
<p>  
  The quick brown...  
</p>  
<p>  
  97  
</p>
```

Fragen?



jQuery + Twitter

Twitter einbinden

```
$(document).ready(function(){
  var tweeturl = "http://twitter.com/status/user_timeline/mtech2010.json?
count=5&callback=?";
  $.getJSON(tweeturl, function(data){
    $.each(data, function(i, item) {
      var txt = item.text
        .replace(/(https?:\V\[-a-z0-9._~:\V?#@!$&\'()*+,\;=%]+)/ig, '<a href="$1">
$1</a>')
        .replace(/@+([\_A-Za-z0-9-]+)/ig, ' <a href="http://twitter.com/$1">@$1</
a>')
        .replace(/#+([\_A-Za-z0-9-]+)/ig, ' <a href="http://search.twitter.com/
search?q=$1">#$1</a>');
      $('<p></p>').addClass(i%2 ? 'even' : 'odd').html(txt).prependTo('#tweet');
    });
  });
});
```


Fragen?



Vielen Dank für Eure Aufmerksamkeit.