Multimedia im Netz
Online Multimedia
Winter semester 2015/16

Tutorial 08 – Minor Subject
Today’s Agenda

• Evaluation Results
• Quick Test
• Breakout Session: Assignment 07
• Advanced jQuery
  – Handy methods
  – Effects
• Breakout
  – Spotify search extensions
  – More functions: Images, links, search for tracks, etc.
• Round-up
Evaluation

Results & Discussion
Participants / Respondents

• N = 48
  – Major subject: 42
  – Minor subject: 6
Main reasons for not participating or not submitting the assignments

- No time: 20
- Too hard: 8
- Other Obligations: 8
- Too much: 6
- Too fast: 4
Amount of Work

• „Aufgaben brauchen oft viel Zeit, mindestens 6-8 Stunden, wenn alles bearbeitet werden soll“
  – Approx. 30 working hours per ECTS credit
  – Quick calculation: 6 ECTS $\rightarrow$ 6 * 30h = 180 working hours per semester
    - 14*3 = 42 hours for lecture
    - 13*3 = 39 hours for tutorials
  \[\rightarrow 99 \text{ hours for assignments} \]
  \[\rightarrow 99h / 13 \text{ assignments} = 7.6 \text{ working hours per assignment}.\]
See a summary of all responses:
https://goo.gl/qn3HwO
# Code-Along: Assignment 07

![Spotify Search](image)

<table>
<thead>
<tr>
<th>Artists</th>
<th>Searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eagles</td>
<td>70</td>
</tr>
<tr>
<td>Eagles Of Death Metal</td>
<td>65</td>
</tr>
<tr>
<td>Department Of Eagles</td>
<td>40</td>
</tr>
<tr>
<td>Screaming Eagles</td>
<td>14</td>
</tr>
<tr>
<td>Eagles &amp; butterflies</td>
<td>24</td>
</tr>
<tr>
<td>Eagles For Hands</td>
<td>21</td>
</tr>
<tr>
<td>Super Eagles</td>
<td>12</td>
</tr>
<tr>
<td>The Eagles</td>
<td>8</td>
</tr>
<tr>
<td>Golden Eagles Mardi Gras Indians</td>
<td>15</td>
</tr>
<tr>
<td>Synphonic Eagles</td>
<td>1</td>
</tr>
<tr>
<td>PCP Eagles</td>
<td>5</td>
</tr>
<tr>
<td>Eagles Gift</td>
<td>2</td>
</tr>
<tr>
<td>Giant Eagles</td>
<td>4</td>
</tr>
</tbody>
</table>
Advanced jQuery
Method Chaining

- Basically, any jQuery method returns another jQuery object, that you can now work with.
- (Possible) Advantages: readability, re-use of selection results
- Examples:

```javascript
$('#myDiv').removeClass('off').addClass('on');
```

```javascript
$('#myDiv')
  .css('color', '#cccccc')
  .removeClass('container')
  .attr('id', 'someNewID')
  .append('<span>Text</span>');
```
Efficient DOM-Traversal

• Traverse the DOM tree with CSS selectors and jQuery methods
• Useful to select and edit elements efficiently.

• Examples:
  – $("#myDiv").next("div")
    gets the first div-element that follows the element with ID myDiv

  – $("ul").find("li.item")
    finds all li-elements having the class item within all ul-elements

  – More Examples: http://api.jquery.com/category/Traversing/
$.each()

• „for-loop shortcut“

```
$.each(collection, function(index, item){
    // ...
});

// is equivalent to:

for(var i=0; i<collection.size(); i++){
    var item = collection.get(i);
}
```
\$(document).ready()

- \$(document).ready(onLoadCallback) ensures that the DOM has been loaded entirely and that you can access any element inside it.
- Once the DOM is loaded, the callback is fired

\$(document).ready(function(){
    //JavaScript-Code
});
Effects

• jQuery has a few short-hand methods to animate elements.
• Most notably:
  – .fadeIn(…)  
    run a fade-in animation (appear)
  – .fadeOut(…)  
    run a fade-out animation, the target will have display:none afterwards
  – .fadeToggle(…)  
    Either fades in or fades out the element.
  – .animate(…)  
    Generic animations of CSS properties
• Read more about the effects here: 
  https://api.jquery.com/category/effects/
Effects: Examples

```html
<script>
    var animatedDiv = $('.#animated');
    var fadedDiv = $('.#faded');
    var targets = fadedDiv.find('.target');

    function() {
        targets.fadeToggle();
    });

    $(#size).click(function() {
        var randomSize = (Math.random() * 100) + '%';
        animatedDiv.animate({
            width: randomSize,
            height: randomSize
        });
    });
</script>
```
Animation Callbacks

• You can also pass a function that will be called when the animation has finished.
• This is useful to create a sequence of animations.
• Example for a perpetual animation:

```html
<div id="inOut">Bling bling</div>
<script>
function appear(){
    $('#inOut').fadeOut(disappear);
}
function disappear(){
    $('#inOut').fadeIn(appear);
}

$(document).ready(disappear);
</script>
```
Break-Out: Extending the Spotify Search

https://www.youtube.com/watch?v=rJaZQ7tjLp0
Round-up

1. When and why do we often need `$(document).ready(...)`?
2. How do we find out if an animation is complete?
3. Name one advantage of method chaining.
4. Write down one thing that you’ve learned today and explain it to your neighbor.
Thanks!

What are your questions?