

# Javascript mit jQuery



- Freie Javascript Bibliothek seit 2006
- Wiederverwendbarkeit von Standard Funktionen
- Erweitertes Event System
- Hilfsfunktionen wie z.B. *each*
- Nutzen von Effekten wie *fade* oder *slide*
- Ajax-Funktionalitäten
- Unzählige freie Plugins



- Möglichst nur eine Bibliothek nutzen
- Einbindung der jQuery.js – Bietet alle genannten Funktionalitäten
- Zugriff auf Objekte mit \$-Funktion
- jQuery.noConflict(); gibt das \$ für anderen Bibliotheken frei
- Lokale Einbindung:  
`<script src="pfad-zur-jquery/jquery.js"></script>`
- Gehostete Einbindung:  
`<script src="http://code.jquery.com/jquery-latest.js"></script>`  
`<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>`



- Beispiel 1:  

```
<script>  
    /* Hier der jQuery-Code */  
</script>
```
- Beispiel 2:  

```
$(document).ready(function(){  
    /* Hier der jQuery-Code */  
});
```
- Beispiel 3:  

```
$(document).ajaxComplete(function(){  
    /* Hier der jQuery-Code */  
});
```
- Einbinden im `<head>` - Bereich einer HTML Datei oder extern.



# Ausgaben von Testdaten

- Nutzung von alert:

```
<script>
    $(document).ready(function(){
        alert($('a').attr('title'));
    });
</script>
```

- Nutzung der Javascript Konsole des Browsers:

```
<script>
    $(document).ready(function(){
        console.log($('a').attr('title'));
    });
</script>
```

(DEMO)



- Einen Link verändern:

```
<script>
$(document).ready(function(){
    $('a').attr({
        title: 'jQuery-Kurs',
        href: 'http://www.html-seminar.de/jquery.htm'
    });
});
```

- Inhalte eines Tags verändern:

```
$('h1').html('mit jQuery Inhalte ändern');
```



# Selektieren von spezifischen Inhalten

- Elemente über ID oder Klasse selektieren:

```
<script>
    $(document).ready(function(){
        alert($('#myElementID').html());

        alert($('.myElementClass').html());
    });
</script>
```

- IDs sind eindeutig, Klassen können mehrfach in einer DOM-Struktur vorkommen.
- Elemente können mit mit zusätzlichen Selektoren ausgewählt werden:

```
:first, :last, :nth-child(3) -> alert($('.myElementClass:first').html());
```



- Elemente über CSS Funktion verändern:

```
<script>
  $(document).ready(function(){

    $('p').css('color', '#FF0000');

    $('p').addClass('error');

    $('p').removeClass('error');

  });
</script>
```

CSS:

```
p.error { color: #FF0000; }
```



- Elemente auf Klick selektieren und reagieren:

```
<script>
    $(document).ready(function(){

        $('p.absatz').click(function(){
            $('p.absatz').css('text-decoration', 'underline');
        });

    });
</script>
```

CSS:

```
p.error { color: #FF0000; }
```



- Mittels show/hide Funktionen können Elemente ein- und ausgeblendet werden  
(DEMO)
- Animieren der Funktion über *slow*, *fast* oder *in Millisekunden*
- Toggle Funktion mit nur einem Link nutzen  
(DEMO)
- Rückmeldung über Abschluss der Funktion mittels Callback  
(DEMO)



# DOM – ELEMENTE verschieben

jQuery

- Funktionen können verkettet werden:

```
$(".fa_participant_4711").detach().prependTo('#participant_form');
```

- *prepend / append vs. prependTo / appendTo*



# Eigene Funktionen schreiben

jQuery

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

```
    $(' .btn').('click', function(){  
        scrollTop();
```

```
    });  
});
```

```
function scrollTop() {
```

```
    $("html, body").animate({ scrollTop: 0 }, "slow");  
}
```



# API Dokumentation nutzen

jQuery

<https://api.jquery.com/>