

Übung zur Vorlesung  
**Multimedia im Netz**

Doris Hausen  
Ludwig-Maximilians-Universität München  
Wintersemester 2009/2010

# Prototype

- <http://prototypejs.org>
- „Prototype is a JavaScript Framework that aims to ease development of dynamic web applications.“



- „Installation“
  - prototype.js von der deren Webseite runterladen
  - prototype.js in HTML/PHP/etc.-Seiten einbinden

```
<script src="prototype.js" type="text/javascript"></script>
```

# XMLHttpRequest

- „Standard“ Request
  - `new Ajax.Request(url[, options])`
- Updater
  - `new Ajax.Updater(container, url[, options])`
- Periodical Updater
  - `new Ajax.PeriodicalUpdater(container, url[, options])`

# Ajax.Request

- „Normaler“ AJAX Request

- Beispiel

```
new Ajax.Request(url, {  
  method: 'get',  
  onSuccess: function(transport) {  
    //Funktionsinhalt  
  });  
}
```

- Callbacks (Details: <http://www.prototypejs.org/api/ajax/options>)

- onCreate
- onUninitialized
- onLoading
- onLoaded
- onInteractive
- onXYZ, onSuccess or onFailure
- onComplete

# Ajax.Updater

- AJAX Request & Verarbeitung der Rückgabe
- Beispiel
  - ```
new Ajax.Updater(  
    'result',  
    'script.php', {  
    parameters: { inhalt: $F('inhalt') }  
    });
```
  - `result`: Name der ID wo die Rückgabe sichtbar wird;  
Hier `<div id="result"></div>`
  - `script.php`: URL an die der Request geschickt wird
  - `inhalt: $F('inhalt')`: Nutzereingaben aus dem Formular  
Hier: `<input id="inhalt" type="text"/>`

# Ajax.PeriodicalUpdater

- Wie AJAX Updater der sich regelmäßig wieder aufruft
- Beispiel
  - `new Ajax.PeriodicalUpdater('result', 'script.php', { frequency: 2 });`
  - `result`: Name der ID wo die Rückgabe sichtbar wird;  
Hier `<div id="result"></div>`
  - `script.php`: URL an die der Request geschickt wird
  - `frequency`: Wie lang es dauert bis der Request wieder aufgerufen wird (in Sekunden)

# Observer

- Um Änderungen z.B. Nutzereingaben zu überwachen
- `Event.observe(element, eventName, handler)`
  - `element`: ID des zu überwachenden Events
  - `eventName`: JavaScript EventHandler ohne “on” (siehe auch: <http://de.selfhtml.org/javascript/sprache/eventhandler.htm>)  
z.B. `click`, `keyup`, `keypress`
  - `handler`: Funktion die aufgerufen wird wenn Änderung auftritt
- `new Form.Observer(element, frequency, handler)`
  - `frequency`: wie oft er auf Änderungen überprüft

# Weitere Infos zu Prototype

- API:  
<http://api.prototypejs.org>
- Tipps und Tutorials:  
<http://www.prototypejs.org/learn>