

Perl und Ajax

# Perl und Ajax

---

# Perl und Ajax



Dürfte den meisten hier wohl gut  
bekannt sein ;-)

# Perl und Ajax

---

Perl und **Ajax**



nicht das Putzmittel...

# Ajax

---

- Eine "Web 2.0"-Anwendung ohne Ajax?
  - Kaum noch vorstellbar
- Das ist !keine! Ajax-Einführung
- AJAX → Asynchronous JavaScript And XML
  - Asynchron ? Nicht unbedingt
  - JavaScript ? Ja, das schon
  - And XML ? Nicht unbedingt (JSON)
- Warum?
  - [http://localhost/cgi-bin/Ajax\\_Perl/index\\_cgi\\_ajax.cgi](http://localhost/cgi-bin/Ajax_Perl/index_cgi_ajax.cgi)

# Perl und Ajax

---

**Perl und Ajax**



darum soll es gehen...

# Perl und Ajax

---

- Wie so häufig bietet CPAN alles was man braucht!
- Drei Module möchte ich kurz vorstellen
  - CGI::Ajax
  - HTML::Prototype
  - OpenThought

# HTML::Prototype

---

- Die Doku ist ziemlich rudimentär
- Zu umfangreich für Anfänger, für Prototype-Nutzer wohl gut geeignet
- Integration mit Catalyst auf jeden Fall sehr gut
- Insgesamt das Modul mit dem ich am wenigsten zu tun habe.
- Am wenigsten JS bei diesen drei Modulen notwendig

# CGI::Ajax

---

- Das Ajax-Urgestein
- Sehr einfach, relativ gute Doku
- Einfaches Matching von Subroutinen zu JavaScript-Funktionen

```
my $ajax = CGI::Ajax->new(
    'show_details' => \&show_details,
    'delete2'      => \&delete_entry,
);
print $ajax->build_html( $cgi_obj, erzeuge_html() );
```

# CGI::Ajax

---

```
my $ajax = CGI::Ajax->new(  
    'show_details' => \&show_details,  
    'delete2'      => \&delete_entry,  
);  
print $ajax->build_html( $cgi_obj, erzeuge_html() );
```

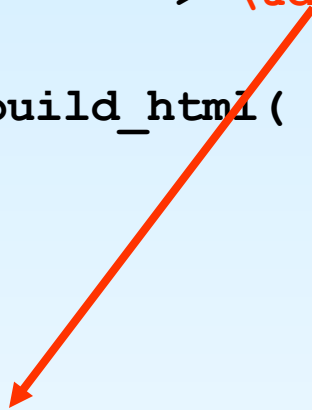


Die Namen der JavaScript-Funktionen...

# CGI::Ajax

---

```
my $ajax = CGI::Ajax->new(  
    'show_details' => \&show_details,  
    'delete2'      => \&delete_entry,  
);  
print $ajax->build_html( $cgi_obj, erzeuge_html() );
```



Die Referenzen auf die Funktionen im Perl-Skript...

# CGI::Ajax

---

```
my $ajax = CGI::Ajax->new(  
    'show_details' => \&show_details,  
    'delete2'      => \&delete_entry,  
);  
print $ajax->build_html( $cgi_obj, erzeuge_html() );
```



Schreibt jede Menge JavaScript in den <head> -Bereich ...  
Man muss keine JavaScript-Funktionen selbst schreiben, ...

# CGI::Ajax

---

..., aber

```
<a href="#"  
    onclick="delete2(['action__delete_task',  
                    'id__<%= ID %>'],  
                    ['task<%= ID %>']);">  
Löschen</a>
```

# CGI::Ajax

---

- Nicht alles klappt problemlos:
  - Integration in Frameworks nicht optimal (Es gibt kein „Skip-Headers“ o.ä.)
  - Fehlermeldungen nicht immer aussagekräftig
- Insgesamt aber gerade für den Einstieg ganz gut geeignet

# OpenThought

---

- Kein(!) Mapping von Perl-Funktionen auf JavaScript-Funktionen.
- Wenige zentrale JavaScript-Funktionen der OpenThought-Klasse
  - CallUrl
  - FetchHtml
- Ein zentrales Objekt, auch bei Frameworks


# OpenThought

---

```
my $obj = OpenThought->new;
```

```
$obj->param(  
    {  
        html_div => $html_code,  
    }  
);
```

```
print $obj->response;
```



Welche Elemente sollen  
welche Werte anzeigen?

# OpenThought

---

```
<a href="#"
  onclick="OpenThought.CallUrl(
    '<%= URL %>',
    'action=delete_task', 'id=<%= ID %>');
">Löschen</a> ||
<a href="#"
  onclick="OpenThought.CallUrl(
    '<%= URL %>',
    'action=show_detail', 'id=<%= ID %>');
">Aufgabe anzeigen</a>
<div id="result<%= ID%>"></div>
```

# OpenThought

---

- Sehr „leichtgewichtig“
- Ganz gute Doku
- Aufrufe ziemlich einfach
- Mein „Lieblings“ Ajax-Modul
- Sehr einfache Integration in Frameworks
- Bietet mehr als nur einfache Ajax-Funktionalität
  - Funktionen, die auch in „normalen“ JS-Funktionen verwendet werden können
- Sehr gutes Debugging
  - Siehe [http://localhost/cgi-bin/Ajax\\_Perl/index\\_openthought.cgi](http://localhost/cgi-bin/Ajax_Perl/index_openthought.cgi)

# Weiterführende Infos

---

- CPAN (<http://search.cpan.org/>)
- \$foo – Perl-Magazin Ausgabe 4

