

Übungsblatt 6

Ziele

- ✓ Inhalte dynamisch nachladen mit Hilfe von AJAX

Aufgaben

Formular

Lieblingstier:

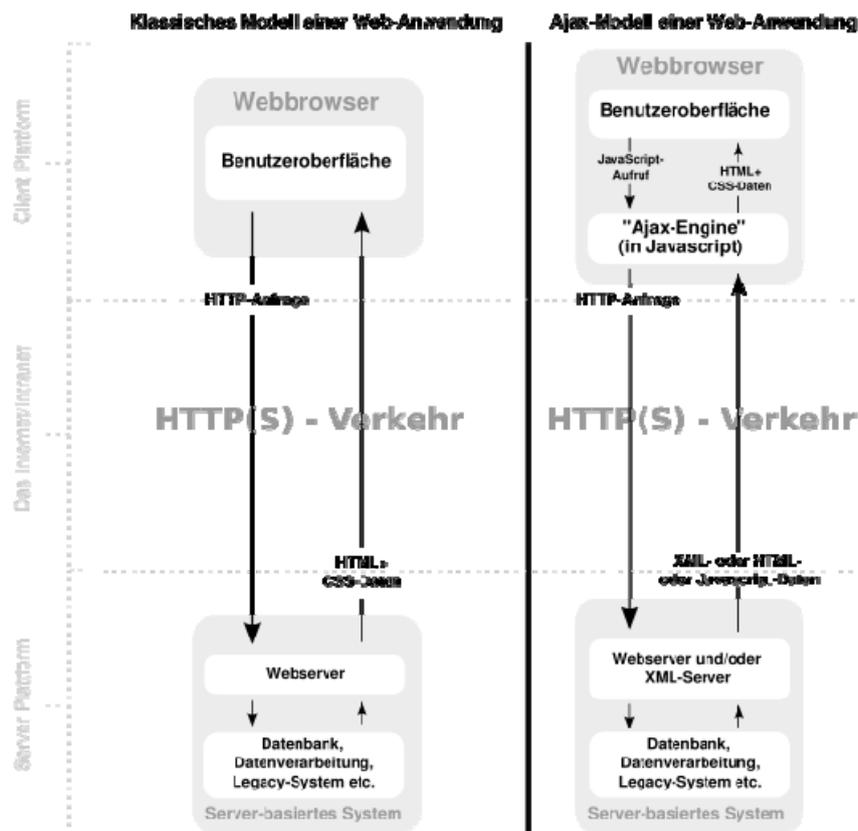


(by Flickr-User: fofurasfelinas)

Aufgabe 1: Ein- und Ausblenden von DOM-Elementen

Manche Inhalte von Webseiten sollen dem Nutzer nur bei Bedarf präsentiert werden aber ohne die gesamte Seite neu zu laden. Das hilft die Ladezeiten zu verkürzen und macht die Benutzung der Webseite dynamischer. Eine Möglichkeit dies zu erreichen ist dabei mehr Inhalte als notwendig herunterzuladen und diese nur bei Bedarf einfach darzustellen.

- Erstellen Sie eine Internetseite die ein Formular mit einem `Dropdown`-Feld mit den Werten „Hund“, „Katze“ und „Maus“ enthält.
- Laden Sie jeweils ein Hunde-, Katzen- und Mäusebild aus dem Internet und fügen Sie es auf der Internetseite ein. Denken Sie dabei ein Bildrechte und verwenden Sie z.B. *Creative Commons* Bilder. Jedes Bild sollte in einen eigenen `div`-Container mit eigener `id`-gepackt werden.
- Machen Sie sich mit dem `style`-Attribut „`display`“ vertraut. Die beiden wichtigsten Werte bei diesem Attribut sind „`block`“ und „`hidden`“. Doch was bewirken die anderen Attribute?
- Schreiben Sie nun eine `JavaScript`-Funktion die alle Bilder beim Öffnen der Seite versteckt.
- Mit einer zweiten `JavaScript`-Funktion zeigen sie nun stets das Bild an welches der Benutzer per `Dropdown` ausgewählt hat.
- Nenn Sie zwei Szenarien in denen eine solche Strategie des Herunterladens von deutlich mehr Inhalten nicht mehr funktioniert.



Quelle: Wikimedia Commons

Aufgabe 2: AJAX

Bisher wurden alle darzustellenden Inhalte immer beim Laden der Seite schon mitgeliefert oder auf der Seite des Benutzers erzeugt. Mit Hilfe von AJAX wird es möglich neue Daten zu laden ohne die Internetseite komplett neu aufzubauen. AJAX steht dabei für „Asynchronous JavaScript and XML“ und ist eigentlich bereits Teil des heutigen JavaScripts.

- Informieren Sie sich über die Benutzung des XMLHttpRequest-Objekts. Kopieren Sie die Datei aus Aufgabe 1 und entfernen Sie die Inhalte der einzelnen div-Container der Bilder. Speichern Sie diese Inhalte in externen Dateien. Sie benötigen allerdings einen neuen div-Container, in dem später dann die geladenen AJAX-Inhalte dargestellt werden.
- Schreiben Sie die Seite so um, dass die Inhalte der Text-Dateien basierend auf der Auswahl aus dem Dropdown per AJAX nachgeladen werden. Das XMLHttpRequest-Objekt selbst können Sie bereits beim Laden des JavaScripts erzeugen. Der eigentliche Ladeaufruf bestehend aus open und send darf dann erst aufgelöst werden sobald sich die Selektion des Dropdown-Feldes ändert.
- Welche Auswirkung hat der letzte Parameter des open-Aufrufs? Wann sollte man dort true und wann false einsetzen?
- Das XMLHttpRequest-Objekt teilt seinen Status über einen den readyState mit. Was bedeuten die verschiedenen readyState-Werte? Wie wird das JavaScript über einen Statuswechsel informiert?
- Beim richtigen readyState sollte nun der geladene Inhalt im AJAX-Container dargestellt werden.

- f) Erstellen Sie sich einen eigenen `AJAX` status indicator auf `www.ajaxload.info` und verwenden Sie diesen während die externe Datei geladen wird. Beim Laden von der Festplatte werden Sie feststellen, dass die Anfragen von `AJAX` so schnell erfolgen, dass der Statusindikator gar nicht sichtbar wird. Verzögern Sie deshalb die Ausgabe des Inhalts um zwei Sekunden. Benutzen Sie hierzu `setTimeout` und vergessen Sie nicht einen gesetzten Timer wieder zu entfernen.

Aufgabe 3: AJAX mit Prototype

`AJAX`-Anfragen mit der `XmlHttpRequest`-Methode zu erstellen ist nicht immer optimal und einfach. Das `Prototype`-Framework ist ein `Open-Source-JavaScript` Projekt das es sich zur Aufgabe gemacht hat einfacheren Zugriff auf `AJAX`-Anfragen und andere Dinge zu ermöglichen.

- a) Laden sie sich die aktuelle `prototype.js` herunter binden Sie sie ein und lösen Sie die Aufgabe erneut mit Hilfe von `Prototype`.