

7 Index

Convention over Configuration.....	
CoC.....	27, 29, 33, 36, 38, 44, 68, 78, 89, 102, 125, 200, 203, 229, 249, 261, 264
Hinweis.....	
Ajax-Werte werden nicht „dauerhaft“ im Browser vermerkt und gehen beim Blättern (noch) verloren.....	146
Caching-Strategien auf die spätere Produktiv-Umgebung ausrichten.....	96, 100, 111, 112, 118, 121, 124, 130
InstantRails.....	25
„@content_for_layout“ ist überholt, ersetzt durch das einfache „yield“.....	52, 102, 116, 145
Linux-Apache-MySQL-PHP/Perl/Python.....	
LAMP.....	30
model-view-control.....	
MVC.....	22, 32, 118
RubyGems.....	21
SOA.....	10
Tipp.....	
Ajax nur gezielt einsetzen, da Blättern und Lesezeichen nicht funktionieren.....	101
Bei Datenmanipulationen auf den korrekten Zustand der Transaktionssteuerung achten.....	141
Bei der Initialisierung des Application-Controllers noch keine Browser-Kommunikation durchführen.....	104
Bei Javascript die Fehlerkonsole im Auge behalten!.....	211, 213
Beim Blättern die Abstimmung von Ajax-Werten im Browser und im Programm kontrollieren!.....	214
Beim Speichern komplexer Instanzen den Umfang der Speicherung beachten.....	176
Cachen im Zusammenhang mit dem Browser ist mit Vorsicht zu genießen.....	149, 168, 214, 218
CGI untauglich für „in-memory“-Speicherung.....	185
Die von Rails erzeugten Tags können über die Funktion „(Seiten-)Quelltext“ des Browsers angesehen werden.....	105
Diverse Debugger je nach Anforderung nutzen.....	93
Firefox: Javascript-Reset und HTML-Reset können unterschiedliche Ergebnisse liefern.....	216
Für Anfänger: Was einfache Templates von Partials unterscheidet.....	104
Javascript kann gelegentlich den Kontrollfluss verlieren.....	213
Javascript kann über die Funktion „Javascript/Fehler-Konsole“ des Browsers angesehen werden (im Firefox unter „Extras“)......	125
Javascript lässt sich meist recht gut über „alert“ verfolgen.....	151, 227
Konventionen auch beachten!.....	79
Listen-Präsentationen als Partial ermöglichen Mehrfachverwendung.....	141
MySQL: Namen mit Bindestrichen in linksgerichtete Anführungszeichen setzen.....	174
MySQL: Namen mit Bindestrichen vermeiden.....	174
options_from_collection_for_select.....	85
Ordner "javascript".....	77
rake löscht Testumgebung.....	78
Reservierte Worte generell meiden.....	95
Rubyforge und Sourceforge nach Rails-Beispielen überprüfen.....	90, 97
scaffold-Generator zur Erzeugung von Testfällen und Anschauungsbeispielen nutzen.....	90
Templates debuggen.....	146
Tooltips per CSS.....	105, 109
Versionen von Rails im Auge behalten.....	91
Zwei kritische Gesellen: Scrollbalken und Drag & Drop.....	176, 227, 256
“irb“ nicht vergessen! neue Befehle oder Ab-fragen einfach ausprobieren.....	91
„Log“ nicht vergessen! Bei Problemen das Rails-Protokoll prüfen.....	175
4ff-Methode.....	
Akzentuierend.....	72
aufgabenbezogener Typ.....	71
Aufzeichnend.....	72
Auswertend.....	73
Deskriptiv.....	72

Deskriptive.....	74, 172
Dokumentarisch.....	72
Dokumentative.....	74, 172
Eigengewicht.....	69
Exit-Distanz.....	71
Exitgewicht.....	71
feldbezogene Typ.....	71
feldbezogener Typ.....	71
Klassifizierend.....	72
Logische Felder.....	74, 169
Logische Felder sind zentral für die Anwendung.....	171
Portal-Distanz.....	70
Portalgewicht.....	71
Profilgewicht.....	69
Stempelnd.....	72
Tentakelzahl.....	69
Typgewicht.....	73
Umleitend.....	73