

Einleitung	IX
1 Hinein ins Vergnügen	1
Warum funktionale Programmierung?	1
JavaScript	2
Funktionale Sprachelemente und Konzepte	10
Unterscheidungsmerkmale von Programmiersprachen	11
Aufbruch in die Welt der funktionalen Programmierung und der Gewürze	13
Rezeptor: Über hundert Curry-Gerichte, funktional und automatisch erstellt	14
2 Abstraktionen	21
Modellierung: Theorie und Praxis	22
Abstraktion von einem Wert: Variablen	23
Abstraktion von einem Codeblock: Funktionen	24
Abstraktion von einer Funktion: Funktionen höherer Ordnung	25
3 Ein Topf mit Curry	31
Präfix, Infix, Stelligkeit	33
Teilweise Funktionsanwendung	40
Automatische Curryfizierung	40
Implizite Argumente, Komposition und Point-Free-Style	42
4 Gemüse, Map, Reduce und Filter	45
Auberginen und die Funktionen höherer Ordnung	45
Weitere Funktionen höherer Ordnung: Map, Reduce, Filter und Kolleginnen	46
Parallele Berechnung auf Arrays mit Map	48

Kombinieren von Lösungen mit Reduce	52
Auswählen von Lösungen mit Filter.	59
Zusammenhang zwischen Reduce und Map	60
Verallgemeinerung der Funktionen höherer Ordnung auf Arrays	61
5 Rekursion	65
Vom Problem zum Programm: Spezifikation, Ein- und Ausgabe	65
Rekursive Funktionen	66
Strukturelle Rekursion auf Listen.	77
Strukturelle Rekursion auf Bäumen	83
6 Event-basierte Programmierung und Continuations	89
Continuations.	93
7 Vom Lambda-Kalkül und Lammcurry	101
Ausdrücke im Lambda-Kalkül	103
Reduktion.	106
Substitution	108
Wohlgeformte Ausdrücke	111
Gleichheit von Ausdrücken – Äquivalenz	111
Auswertungsstrategien und Normalformen	112
Repräsentation von Daten im Lambda-Kalkül.	114
8 Typen	129
Typisierung und Typüberprüfung	129
JavaScript ist dynamisch typisiert.	129
JavaScript ist schwach typisiert	130
Vorsicht beim Umgang mit Typen in JavaScript	131
Primitive Datentypen	131
Arrays	133
Undefined und Null	134
Primitive Datentypen versus Objekte.	135
Polymorphismus.	137
JavaScript: Polymorphismus durch Prototypen	139
Quellen der Verwirrung durch das Typsystem	141
Stärkere Typen für JavaScript.	143
Fazit zu JavaScripts Typsystem	145

9	Kochen als Monade	149
	Ein programmierbares Semikolon	150
	Monaden als Behälter: Dinge, Kochtöpfe und Monaden.	153
	Das Monaden-Muster	154
	Monaden links, Monaden rechts, Monaden überall	155
	Zusammenhang zwischen Continuation-Passing-Style und Monaden	163
	Warum das Ganze?	165
	Weblinks und weiterführendes Material	167
10	Nachtisch und Resteessen	169
	Eigenheiten der Sprache JavaScript	170
	Konzepte der funktionalen Programmierung.	172
	Funktionale Bibliotheken.	173
	Sprachen, die JavaScript verbessern wollen	174
	Die Welt der funktionalen Sprachen abseits von JavaScript	176
	Anhang: Link- und Literaturliste	179
	Index	187