



Praktikum zu LV 7328 - Ruby:

Übung 01

Informationsquellen
Nutzung von ruby, irb, Editoren
Strings: Ein *here document*



Organisatorisches



- Arbeitsverzeichnis:
`~/kurse/ruby/01/`
- Dateinamen:
`01-answers.txt` # Vorlage abholen, ausfüllen, abgeben
`01-heredoc.rb` # neu erstellen & abgeben
- Werkzeuge:
`ruby` # Der Interpreter
`irb` # Interactive Ruby "shell"
`emacs` # mit Ruby-Mode. NICHT X-Emacs
`scite` # Ein portabler Editor, auch
mit Ruby-Mode
- Vorlagen:
`/local0/werntges/kurse/ruby/01/01-answers.txt`



Die Aufgabe



- A: Arbeiten mit der Ruby-Referenz
 - Die Leistungsfähigkeit von Ruby ist nicht zuletzt in der Vielfalt eingebauter Klassen und deren zahlreichen Methoden begründet.
 - Suchen nach einer passenden Methode für eine gegebene Aufgabe ist daher üblich und notwendig.
 - Das "*pickaxe*"-Buch enthält das notwendige Referenzmaterial. Wir üben nun den Umgang damit:
- Vorbereitungen:
 - Besuchen Sie im Web die Seiten des "*pickaxe*"-Buches (Link unter "Begleitmaterial und Links" der Ruby-LV)
 - Wählen Sie das Kapitel "Built-in Classes and Methods".
 - Wählen Sie die Klasse "String" aus.



Die Aufgabe



- A: Suchaufgaben
 - 1: Suchen Sie nach Methoden, die einen gegebenen String
 - a) rechtsbündig (mit vorgegebener Breite)
 - b) linksbündig (dito)
 - c) zentriert (dito)ausgeben.
 - 2: Welche beiden Methoden liefern die Länge des Strings?
 - 3: Wie wandelt man alle Buchstaben eines Strings in
 - a) lauter Grossbuchstaben
 - b) lauter Kleinbuchstaben ?
 - 4: Finden Sie drei Wege, wie man testen kann, ob String s leer ist.



Die Aufgabe



- B: Tests mit Interactive Ruby (irb)
 - 1: Öffnen Sie eine Shell, wechseln Sie in Ihr Arbeitsverzeichnis zu dieser Aufgabe (s.o.), rufen Sie "irb" auf. Geben Sie ein:

```
a = "Norwegen"
b = "Dänemark"
c = "Dies ist ein Satz."
```
 - 2: Testen Sie die drei Methoden aus Teil A1, a-c mit String a und den "Feldbreiten" 25 und 26.
Wie verhält sich der Zentrierer, wenn es keine exakte Mitte gibt?
 - 3: Wenden Sie A2 und A3 a,b auf Strings a - c an.
Wie verhält sich Ruby im Fall A3(a) beim Umlaut?
 - 4: Testen Sie auch Ihre drei Antworten zu A4 aus.
 - 5: (*) Was ergibt `c.succ` ? Warum??



Die Aufgabe



- C: Ruby auf der Kommandozeile (Option -e)
 - 1: Verlassen Sie "irb" mittels `exit`, kehren Sie auf die Kommandozeile zurück. Ermitteln Sie nun jeweils mit einem Aufruf von `ruby`:
 - 2: die aktuelle Zahl der Sekunden der "Epoche" (seit 1.1.1970 GMT)
 - 3: die Mikrosekunden der Sekunde des Aufrufs
Arbeitet Ihre Systemuhr überhaupt mit Mikrosekunden-Auflösung?
 - 4: (*) die Mikrosekunden-Differenz zweier möglichst dicht auf einander folgender Zeitpunkte. Was ist Ihr kleinster μ s-Abstand?

Hinweise:

- Verwenden Sie die Ruby-Option -e
- Sie benötigen Methoden der Klasse `Time`.
- Geben Sie die Ergebnisse einfach mit `puts` aus.



Die Aufgabe



- D: Eine ausführbare Ruby-Datei
 - Legen Sie eine leere Datei `01-heredoc.rb` an. Geben Sie der Datei Ausführungsrechte (Hinweis: `chmod +x ...`).
 - Test: Schreiben Sie das "Hello World"-Beispiel aus der Vorlesung in diese Datei. Ist die Datei direkt ausführbar? Stimmt das Ergebnis? (Nein: Fehlersuche, Ja: Nun kann's losgehen...)
 - "Autoanswer-Szenario"
 - Entfernen Sie die "hello world"-Ausgabe wieder.
 - Fordern Sie zur Eingabe eines Namens auf (`print`).
 - Lesen Sie die Eingabe von der Kommandozeile (`gets`).
 - Erfassen Sie eine kleine Abwesenheitsnachricht als "here document", siehe beiliegendes Output-Beispiel.
 - (a) der eingegebene Name, (*b) das aktuelle Datum sollen zur Laufzeit ergänzt werden. Geben Sie das Ergebnis einfach mittels `puts` aus.
 - Mit einer Abfrage nach email-Adresse und einer Mailing-Methode wäre wir schon nahe einer brauchbaren kleinen Anwendung...



Die Aufgabe



- D: Beispielausgabe

```
$ ./01-heredoc.rb
```

```
Enter name: David
```

```
Wiesbaden, 2003-10-08
```

```
Dear David,
```

```
thanks for your message! I am currently unavailable  
and cut off from email, but will be back in the office  
by next Monday. I'll answer your message a.s.a.p.
```

```
Best regards,
```

```
(Ihr Name hier)
```

```
$
```