

**Aufgabe 1**

Schreiben Sie eine Funktion, die bei gegebenem Radius eines Kreises Fläche und Umfang wie folgt ausgibt:

```
R> kreis(1)
Die Fläche ist: 3.141593
Der Umfang ist: 6.283185
```

**Aufgabe 2**

Schreiben Sie eine analoge Funktion für Rechtecke.

**Aufgabe 3**

Schreiben Sie eine Funktion, die einmal würfelt (`sample()`), die gewürfelte Zahl ausgibt, und dann dem Benutzer gratuliert, falls ein Sechser gewürfelt wurde, oder anderenfalls den Benutzer auffordert, noch einmal zu würfeln.

**Aufgabe 4**

Schreiben Sie eine Funktion, die überprüft, ob eine Person für eine Achterbahn im Prater groß genug ist. Im Argument `groesse` der Funktion steht die Körpergröße, im Argument `mindest` steht die Mindestkörpergröße. Die Funktion soll nun prüfen, ob die Mindestkörpergröße überhaupt erreicht ist oder nicht. Falls die Mindestkörpergröße nicht erreicht ist, soll das Programm nur ausgeben: „dafür sind Sie zu klein“. Anderenfalls soll das Programm prüfen, ob die Mindestkörpergröße um mehr als 5cm überschritten wurde, und in diesem Fall ausgeben: „passt“, anderenfalls „das war knapp“.

**Aufgabe 5**

Schreiben Sie eine Funktion, die abhängig von einer Schulnote ausgibt, ob die Note positiv oder negativ ist. Die Note kann dabei entweder numerisch oder als Zeichenkette („sehr gut“, „gut“, ...) übergeben werden. Stellen Sie sicher, daß Groß/Kleinschreibung keine Rolle spielt (z.B. `tolower()`). Bei falsch geschriebenen Noten geben Sie eine Fehlermeldung aus.

**Aufgabe 6**

Der junge Gauß hat eine kleine Formel entwickelt, mit der man zu einem Datum bestehend aus Tag, Monat und Jahr den zugehörigen Wochentag ermitteln kann. Schreiben Sie eine Funktion, die zu einem vom Benutzer eingegebenen Datum den zugehörigen Wochentag ermittelt. Damit die untenstehende Formel richtige Ergebnisse liefert, muß das Datum zwischen dem 01.01.1582 und dem 31.12.3000 liegen.

Die Berechnung erfolgt mit der folgenden von Gauß stammenden sogenannten Kalenderformel. Das zu behandelnde Datum sei in den Variablen `tag`, `monat`, `jahr` abgelegt. Damit werden zwei Hilfsgrößen `h` und `k` wie folgt berechnet:

Bedingung	Hilfsvariable h	Hilfsvariable k
<code>monat &lt;= 2</code>	<code>h = monat + 12</code>	<code>k = jahr - 1</code>
<code>monat &gt; 2</code>	<code>h = monat</code>	<code>k = jahr</code>

Der Wochentag `wt` ergibt sich nun durch:

```
wt = tag + 2*h + (3*h + 3) div 5 + k + k div 4 - k div 100 + k div 400 + 1
wt = wt mod 7
```

Dadurch erhält die Variable `wt` einen Wert zwischen 0 und 6. Dabei bedeutet 0 = Sonntag, 1 = Montag, usw.

`div` ist der Operator für Ganzzahldivision, so gilt etwa  $15 \text{ div } 7 = 2$ , da 7 in 15 zweimal enthalten ist. Der Rest wird dabei nicht berücksichtigt, analog  $17 \text{ div } 3 = 5$ . In R berechnet der Operator `%%` die Ganzzahldivision.

`mod` ist der Restoperator, so gilt etwa  $15 \text{ mod } 7 = 1$ , da 15 geteilt 7 den Rest 1 ergibt. Der ganze Anteil wird dabei nicht berücksichtigt, analog  $17 \text{ mod } 3 = 2$ . In R wird der Restoperator realisiert durch `%%`.

### Aufgabe 7

Schreiben Sie ein Progrämmchen, das eine Liste der Zweierpotenzen von 0 bis 10 erstellt:

```
2 hoch 0 ist 1
2 hoch 1 ist 2
2 hoch 2 ist 4
2 hoch 3 ist 8
2 hoch 4 ist 16
2 hoch 5 ist 32
2 hoch 6 ist 64
2 hoch 7 ist 128
2 hoch 8 ist 256
2 hoch 9 ist 512
2 hoch 10 ist 1024
```

Verwenden Sie eine Schleife, nicht 10 Kommandos. Erlauben Sie, daß Basis (oben 2) und maximale Potenz (oben 10) vom Benutzer angegeben werden können.

### Aufgabe 8

Ein typischer Anwendungsfall für eine `while`-Schleife: Würfeln Sie so lange, bis Sie einen Sechser gewürfelt haben! Geben Sie jeden Wurf in einer Zeile aus. Modifizieren Sie die Funktion, daß `n` Würfe in jeweils einer Zeile ausgegeben werden.

### Aufgabe 9

Schreiben Sie eine Funktion, das eine Basensequenz (der Basen Thymin, Adenin, Cytosin, Guanin) wie `catggctgcagc` in 3 Basen lange Stücke (Tripletts, Codons) zerlegt und in einem Vektor abspeichert:

```
Basensequenz: catggctgcagc
Tripletts: cat, ggc, tgc, agc
```

Erweitern Sie die Funktion dahingehend, daß ein Vektor von Basensequenzen übergeben werden kann und eine Liste von Vektoren mit Codons retourniert wird (Das Zerlegen soll in einer Schleife mit Hilfe von `substr()` erfolgen).

### Aufgabe 10

Die **Fibonacci-Folge** ist eine unendliche Folge von Zahlen (den Fibonacci-Zahlen), bei der sich die jeweils folgende Zahl durch Addition der beiden vorherigen Zahlen

$$f_n = f_{n-1} + f_{n-2}$$

mit den Anfangswerten  $f_0 = 0$  und  $f_1 = 1$  ergibt. Also, 0, 1, 1, 2, 3, 5, 8, 13, ... Schreiben Sie eine `while`-Schleife, welche in einem Vektor `fib` die ersten 30 Fibonacci-Zahlen abspeichert. Schreiben Sie damit eine Funktion mit Argumenten `von` und `bis`, die den Vektor der entsprechenden Fibonacci-Zahlen liefert (also z.B.  $f_{10}, \dots, f_{20}$ ).