0_2_listen

March 17, 2023

1 Datenstrukturen

1.1 Listen

Listen stellen eine weitere Form einer eindimensionalen Datenstrukturen dar. Sie bestehen aus einem bis vielen Elementen verschiedener Datentyps und Strukturen.

1.1.1 Erzeugung von Listen

[ ]:
```
# Listen können aus einem Datentyp bestehen.
l1 <- list(1L, 2L, 3L)
l1
```

[ ]:
```
# Listen können aus unterschiedlichen Datentypen bestehen.
l2 <- list("Name", "Vorname", 13, TRUE)
l2
```

[ ]:
```
typeof(l2)
```

[ ]:
```
str(l2)
```

1.1.2 Adressieren der Elemente einer Liste

Die Indizes sind 1-basiert, d.h., das erste Element einer Liste hat die Nummer 1.

Die Zugriff erfolgt über den Index in eckigen Klammern.

[ ]:
```
# Zugriff auf das dritte Element.
l2[3]
```

1.1.3 Ändern von Listenelementen

[ ]:
```
# Änderung des dritten Elements.
l2[3] <- 25
l2
```

1.1.4 Ergänzen von Listenelementen

Die ergänzte Liste muss einer Variablen zugewiesen werden. append() selbst ändert die Liste nicht.

[ ]:
```
# Ein Element anhängen.
append(l2, "Gelb")
```

[ ]:
```
# Das Element ist noch nicht in der Liste.
l2
```

[ ]:
```
# Die neue Liste muss explizit gespeichert werden.
l2 <- append(l2, "Gelb")
l2
```

1.1.5 Entfernen von Listenelementen

[ ]:
```
# Entfernen des ersten Elementes.
l2 <- l2[-1]
l2
```

[ ]:
```
# Entfernen des letzen Elements.
# length() gibt die Länge einer Liste zurück.
l2 <- l2[-length(l2)]
l2
```

[ ]:
```
# Entfernen mehrerer Elemente.
l3 <- list(1, "A", "B", TRUE, 2, 10)
l3
```

[ ]:
```
# Entfernen des dritten und vierten Elements.
l3 <- l3[-c(3,4)]
l3
```

1.1.6 Schleifen und Listen

Es gibt eine For-Schleife, die jedes Listenelement anspricht.

[ ]:
```
# Einfache For-Schleife, Bedingung in runden Klammern, Anweisungen in␣
↪geschweiften Klammern.
l3 <- list(1, "A", "B", TRUE, 2, 10)
for (l in l3) {
    print(l)
}
```

1.1.7 Prüfen auf Existenz

Die Liste kann auf die Existenz eines Elements geprüft werden, ohne dass man über alle Elemente iterieren muss.

[ ]:
```
# Prüfung mit %in%-Operator.
l3 <- list(1, "A", "B", TRUE, 2, 10)
print("A" %in% l3)
```

```
print(TRUE %in% l3)
print(FALSE %in% l3)
print(2 %in% l3)
```

[ ]:
```
# Prüfung mehrerer Elemente.
print(c(1, 2) %in% l3)
```

1.1.8 Listen von Listen

[[]] gibt das Element selbst zurück

[] gibt eine Liste der gewählten Elemente zurück

[ ]:
```
# Drucken von Listen.
print(list(3, 4))
```

[ ]:
```
# Verschachtelte Listen.
l4 <- list(c(3, 1), c(4, 2), list("A", 3, "C"))
print(l4)
```

[ ]:
```
l4
```

[ ]:
```
# Zugriff.
l4[2]
```

[ ]:
```
# Zugriff.
l4[[2]][2]
```

[ ]:
```
# Zugriff.
l4[[3]][1]
```

[ ]:
```

```