

Arrays und die Doppelpunktnotation

Ziel: Ziel der nächsten Übungen ist die Verwendung grundlegender Befehle zum Erstellen und Verändern von Arrays und das Üben der Doppelpunkt Notation für den Zugriff auf Arrays.

Voraussetzung: Die Eigenschaften von Arrays und von Funktionen, die Arrays erzeugen bzw. verändern werden im Kapitel [Kapitel 3](#) erklärt.

Basiswissen über dieses Kapitel des Skriptums ist unbedingte Voraussetzung für die Übung. Probieren Sie daher u.a. die Befehle [zeros](#), [ones](#), [eye](#), [diag](#) aus, und versuchen Sie mit der [Doppelpunkt](#) Notation auf Teile der Matrizen zuzugreifen. Eine weitere ausführliche online Dokumentation zu Matrizen gibt es hier: [Arrays and Matrices](#).

Übungen zur Manipulation von Matrizen:

1. Erstellen einer einfachen Matrix: `array-const.m`
2. Erstellen einer Matrix mit Schachbrett-muster: `array-chess.m`
3. Erstellen einer grossen aus Diagonalmatrizen bestehenden Matrix: `array-diag.m`
4. Wie eine Diagonalmatrix, aber mit 1 statt 0 bei nichtdiagonal-elementen: `array-diagplus.m`
5. Mit Zufallszahlen gefüllte symmetrische Matrix: `array-randsym.m`
6. Erzeugung und Überprüfung eines [magischen Quadrats](#): `array-magic.m`
7. Matrix mit zeilenweise durchnumerierten Elementen: `array-numberst.m`
8. Sternförmige Matrix: `array-star.m`
9. Variation der sternförmigen Matrix: `array-rstar.m`
10. Matrix mit NaN an den Rändern: `array-setnan.m`

Hinweis:

In den folgenden Übungen sollen nicht übergebene ([nargin](#)), oder leere [] Parameter sollen gleich am Anfang auf die entsprechenden Defaultwerte gesetzt werden. Das kann etwa so aussehen:

```
if nargin < 1 || isempty(m), m=3; end
if nargin < 2 || isempty(n), n=4; end
if nargin < 3 || isempty(v), v=3; end
```

In [MLTutor Hinweise](#) werden diese Programmzeilen näher besprochen.