

# NaN- Setzen

Schreiben Sie eine MATLAB-Funktion `array_setnan.m`, die folgende Aufgaben erfüllt:

1. Übergabeparameter sind  $m$
2. Erzeugen Sie die folgende Matrix der Größe  $m \times n$  (Defaultwert  $5 \times 6$ ),

$$A = \begin{bmatrix} \text{nan} & 1 & 1 & 1 & 1 & \text{nan} \\ 1 & \text{nan} & \text{nan} & \text{nan} & \text{nan} & 1 \\ 1 & \text{nan} & \text{nan} & \text{nan} & \text{nan} & 1 \\ 1 & \text{nan} & \text{nan} & \text{nan} & \text{nan} & 1 \\ \text{nan} & 1 & 1 & 1 & 1 & \text{nan} \end{bmatrix} .$$

3. Beginnen Sie mit einer Matrix mit lauter Einsen und setzen Sie dann mit Hilfe der **Doppelpunkt-**Notation in zwei Befehlszeilen einmal die Eckpunkte und dann den Mittelbereich auf nan. Dabei ist immer das Keyword `end` zu verwenden.

Hinweis:

```
A(1, [1, end]) = nan
```

so kann man zwei Zellen des Arrays gleichzeitig auf nan setzen. Ganz ähnlich lassen sich gleichzeitig alle Eck-Zellen auf nan setzen. Probieren Sie am besten ein paar Kombinationen der Array-Indizierung: siehe [Scriptum Kap 3.6 Hinweis:](#)

Der Wert nan symbolisiert in MATLAB "Not a Number". Mit nan kann man immer weiterrechnen, wobei die Rechenregel ganz einfach ist. Jede Operation mit nan gibt wieder nan.

Gesucht: Funktion `array_setnan.m`

```
[A] = array_setnan(m,n)
    m      : Zeilenanzahl
    n      : Spaltenanzahl
    A      : Rückgabematrix
```

---

Anschauungsbeispiel:

```
>> array_setnan(4,7)
ans =
    NaN     1     1     1     1     1     NaN
     1    NaN    NaN    NaN    NaN    NaN     1
     1    NaN    NaN    NaN    NaN    NaN     1
    NaN     1     1     1     1     1     NaN
```