

Kapitel 8

Datentypen - Klassen

Dieses Kapitel ist erst im Entstehen.

Information findet man derzeit unter der MATLAB-Hilfe für [Datentypen](#).

In MATLAB gibt es wie praktisch in jeder Programmiersprache verschiedene Datentypen bzw. Klassen von Daten. Es gibt mehrere numerische Datentypen [8.1](#), einen logischen Datentyp [8.2](#), einen Datentyp für Zeichen bzw. Zeichenketten [8.3](#), zwei Klassen für Behälter [8.4](#) die wiederum alle anderen Typen beinhalten können, und zwei Klassen für Funktionen [8.5](#).

8.1 Numerische Datentypen

Die numerischen Datentypen unterscheiden sich durch den Speicherplatz, den sie verbrauchen, durch die größte bzw. die kleinste darstellbare Zahl und durch die Möglichkeit negative Zahlen darzustellen oder nicht.

KLASSE	B	KLEINSTE ZAHL	GRÖSSTE ZAHL
double	8	$\pm 2.2251e-308$	$\pm 1.7977e+308$
single	4	$\pm 1.1755e-38$	$\pm 3.4028e+38$
int64	8	-9223372036854775808	9223372036854775807
uint64	8	0	9223372036854775807
int32	4	-2147483648	2147483647
uint32	4	0	2147483647
int16	2	-32768	32767
uint16	2	0	32767
int8	1	-128	127
uint8	1	0	127

Zu dieser Tabelle sind einige Bemerkungen notwendig.

Speicherplatz: Der Eintrag `B` steht für den verbrauchten Speicherplatz für eine Zahl in BYTES, wobei ein BYTE 8 BITS entspricht. Ein BIT ist die kleinste Speicherplatzeinheit, die die beiden Werte 0 oder 1 speichern kann. Um den Speicherplatz eines Feldes zu berechnen, muss man diese Zahl mit der [Anzahl](#) der Elemente im Feld multiplizieren. Handelt es sich ausserdem um eine [Komplexe-Zahl](#) mit [Real-](#) und [Imagiär-](#)Teil muss man den Wert nochmals mit zwei multiplizieren. Der Befehl `whos` gibt Auskunft über den Speicherbedarf der Variablen.

Darstellung: Die Darstellung $2.2251e-308$ bei sogenannten Fließkommazahlen steht für den Wert 2.225110^{-308} , das heisst die Zahl hinter dem `e` stellt den sogenannten Exponenten dar.

Anzahl der Stellen:

Unsignierte Ganze Zahlen:

Kleinste darstellbare Zahl:

Grösste darstellbare Zahl:

Genauigkeit:

Besondere Zahlen:

8.1.1 Fließkomma Datentypen

8.1.1.1 Der Datentyp `double`

Der hauptsächlich in MATLAB verwendete Datentyp ist der Typ `double`. Verwendet man keine Angabe eines Datentyps wird automatisch `double` gewählt.

INPUT	OUTPUT	ERLÄUTERUNG
<code>x = [1:5]</code>	<code>[1 2 3 4 5]</code>	
<code>class(x)</code>	<code>'double'</code>	Klasse
<code>isa(x, 'double')</code>	1	Klassenüberprüfung
<code>isnumeric(x)</code>	1	numerisch - wahr
<code>isfloat(x)</code>	1	Fließkomma - wahr
<code>isinteger(x)</code>	0	ganzzahlig - falsch

Obwohl es also den Anschein hat, als ob die Variable `x` ganzzahlig ist, gehört sie zum Datentyp `double`. Will man sicherstellen, dass die Zahlen nur ganzzahlig, das heisst mit geringerem Speicherbedarf verwendet bzw. gespeichert werden, muss man einen ganzzahligen Datentyp [8.1.2](#) verwenden. Man sollte aber bedenken, dass praktisch alle mathematischen Operationen den Datentyp `double` benötigen.

Darüber hinaus liefern auch Befehle zum automatischen Erzeugen von Feldern (`zeros`, `ones`, ...) ohne Angabe einer Klasse den Datentyp `double`.

8.1.1.2 **Der Datentyp** `single`

8.1.2 **Ganzzahlige Datentypen**

8.2 **Der logische Datentyp** `logical`

8.3 **Der Zeichen-Datentyp** `char`

8.4 **Klassen für Behälter**

8.4.1 **Der Zellen-Klasse** `cell`

8.4.2 **Der Struktur-Klasse** `struct`

8.5 **Klassen für Funktionen**

8.5.1 **Die Klasse** `inline`

8.5.2 **Die Klasse** `functionhandle`