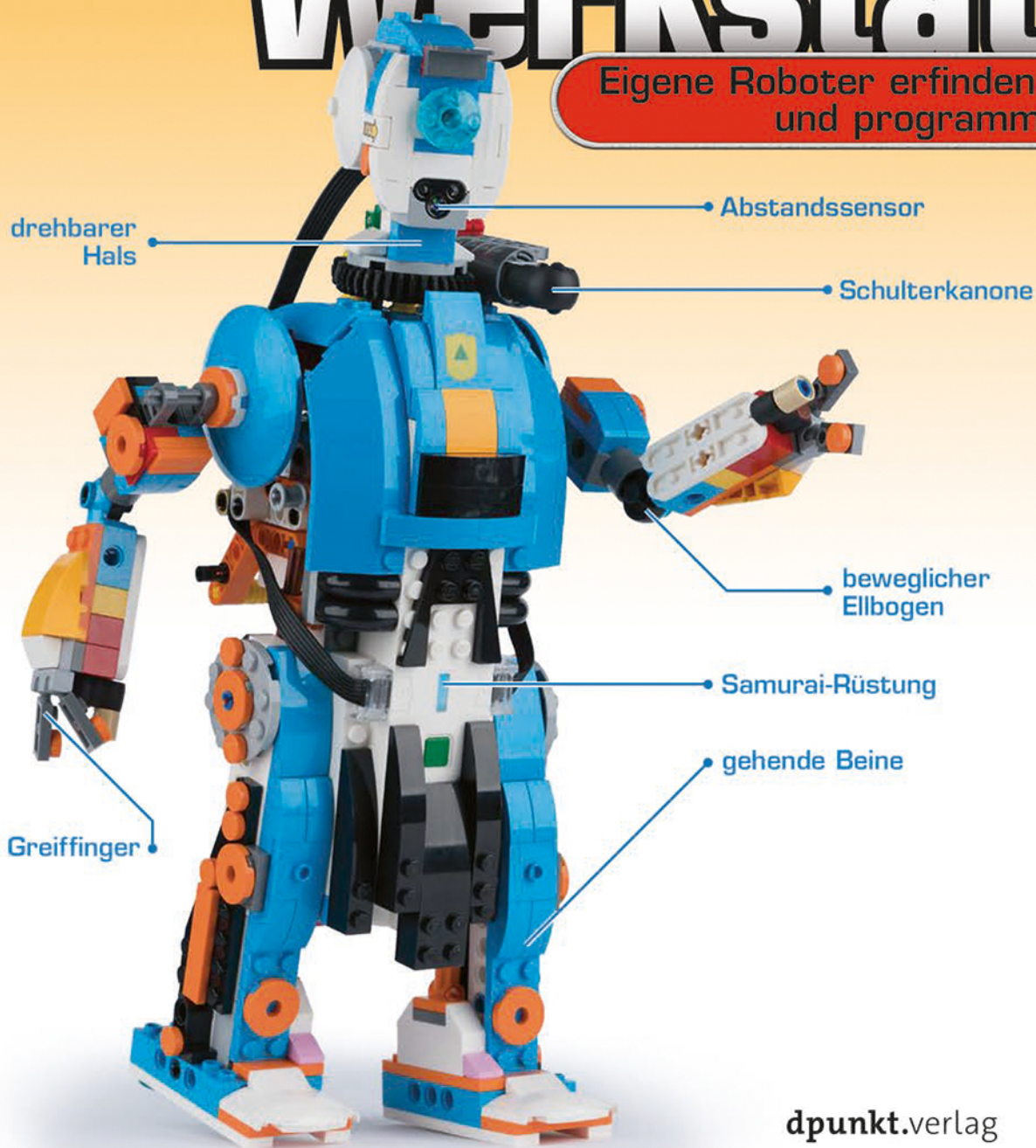


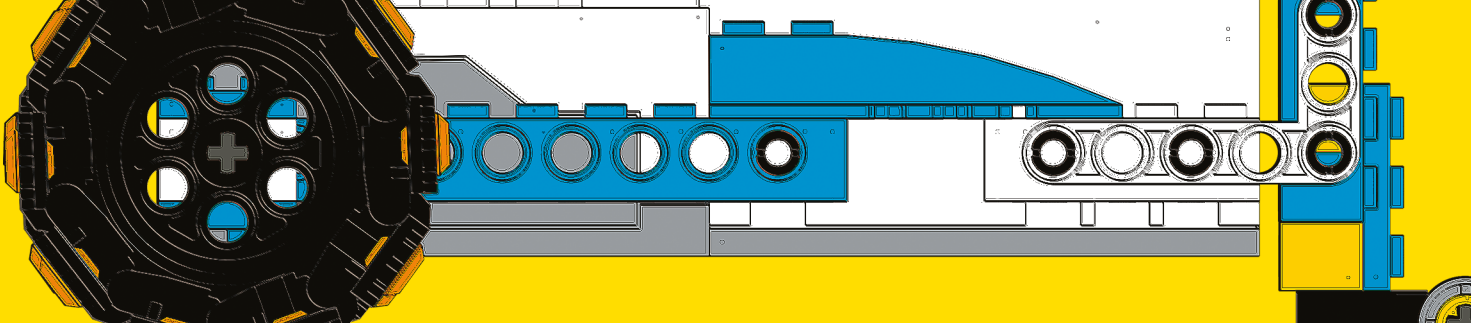
Die LEGO®-Boost- Werkstatt

Daniele Benedettelli

Eigene Roboter erfinden
und programmieren



dpunkt.verlag



2

Gestatten: MARIO!

In diesem Kapitel baust du einen Roboter auf Rädern und schreibst dein erstes Programm! Dazu verwendest du einen Abschnitt der LEGO-Boost-App, der als *Kreative Leinwand* bezeichnet wird und es dir erlaubt, eigene Programme von Grund auf zu erstellen.

Der einfache Roboter, den du in Abb. 2-1 siehst, trägt den Namen MARIO, was für *Multi-Accessory Robot Is Obligatory* steht, also etwa »Roboter mit viel Zubehör ist unverzichtbar«. Das heißt also, dass wir diesen Roboter mit viel Zubehör ausstatten können, sodass er eine Menge toller Sachen machen kann! In den späteren Kapiteln erweitern wir ihn dann auch tatsächlich um viele neue Zusatzteile.

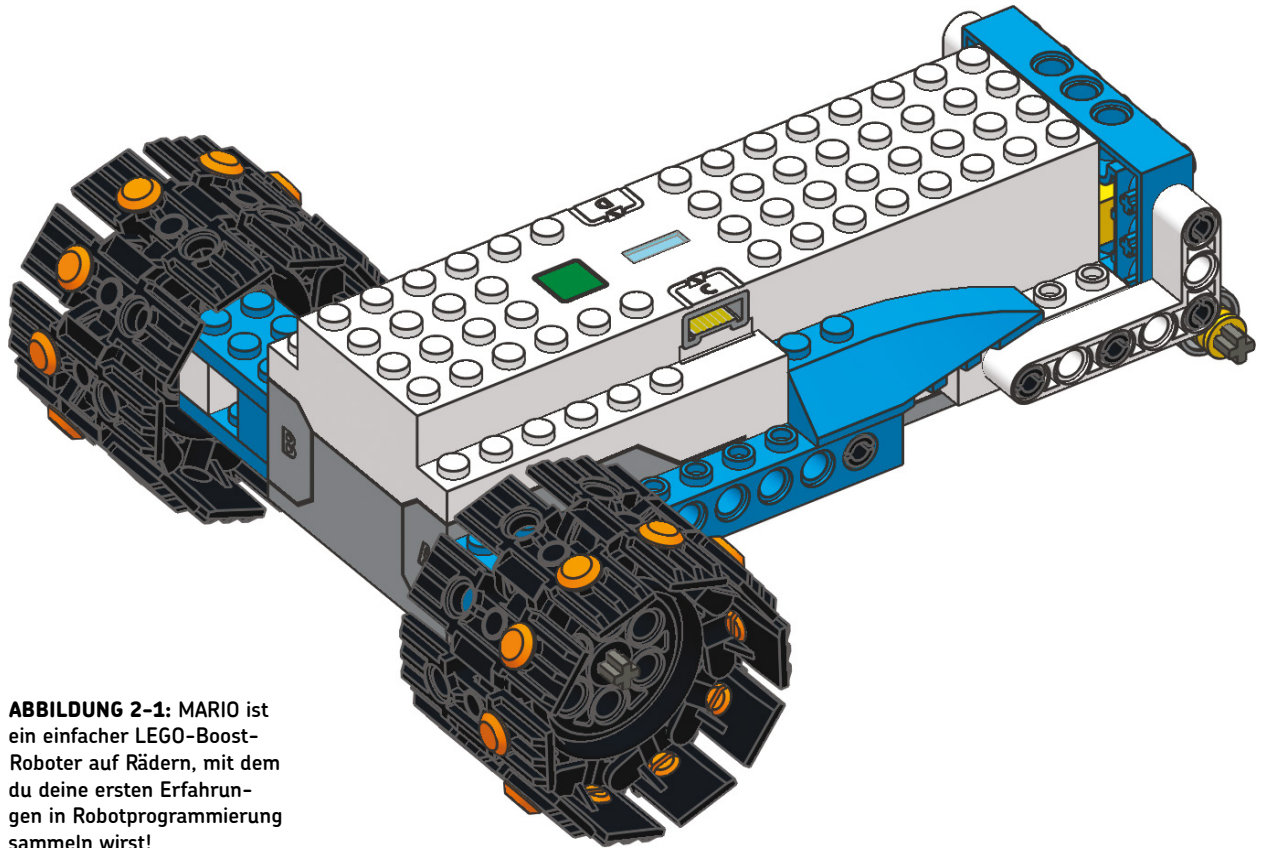
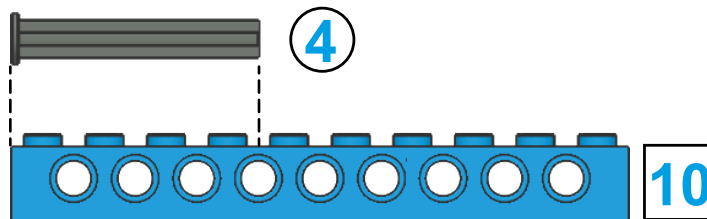


ABBILDUNG 2-1: MARIO ist ein einfacher LEGO-Boost-Roboter auf Rädern, mit dem du deine ersten Erfahrungen in Robotprogrammierung sammeln wirst!

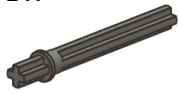
MARIO bauen

Die folgenden Anleitungen beschreiben den Bau Schritt für Schritt, wobei auch jeweils angegeben ist, welche Teile du brauchst. Bei einigen Teilen, etwa Achsen, gebe ich die Länge in LEGO-Einheiten an, damit klar ist, welche Größe du verwenden musst. Um die Achsen zu messen, legst du sie neben einen Technic-Stein und zählst die Noppen, wie Abb. 2-2 zeigt.

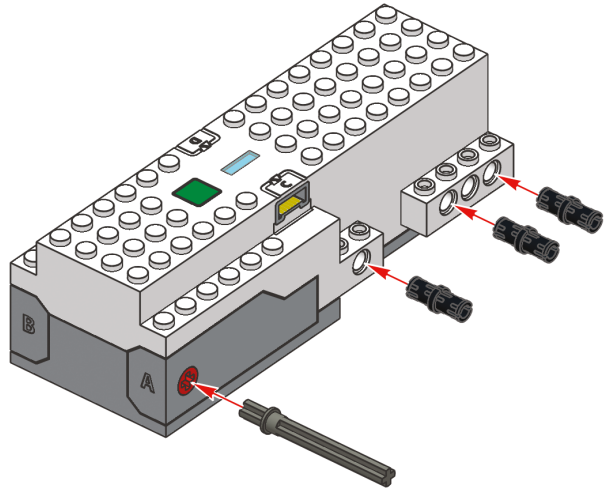
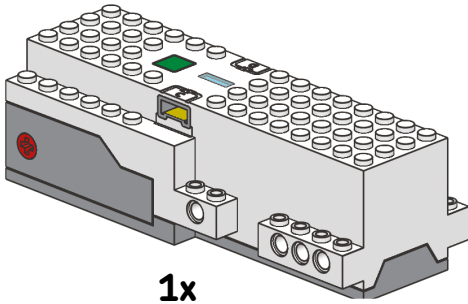
ABBILDUNG 2-1: Die Länge von Steinen wird durch die Anzahl der Noppen angegeben. Eine LEGO-Achse hat keine Noppen, aber du kannst sie neben einen Stein legen und dessen Noppen zählen, um die Länge der Achse zu bestimmen.



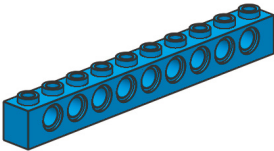
1



1x



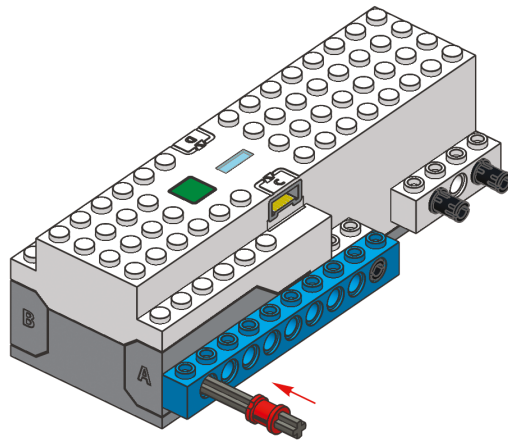
2



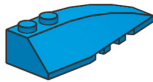
1x



1x



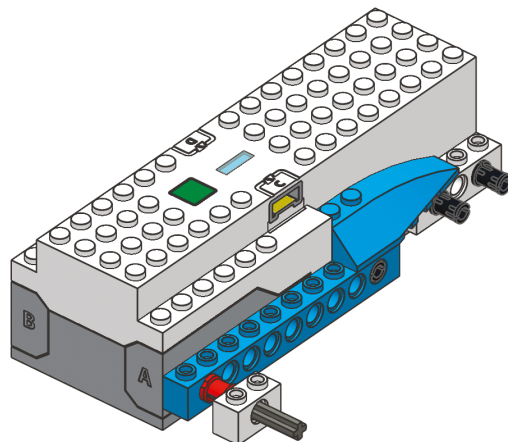
3



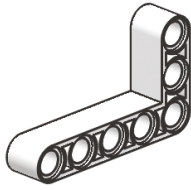
1x



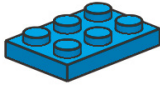
1x



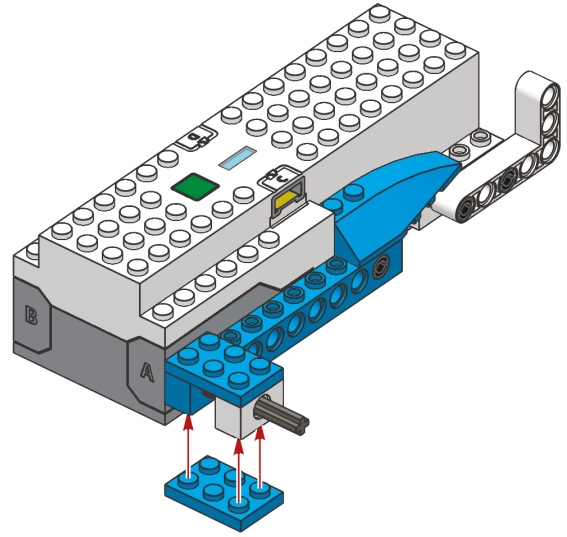
4



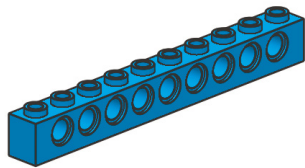
1x



2x



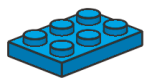
5



1x



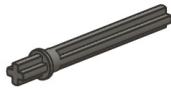
1x



2x



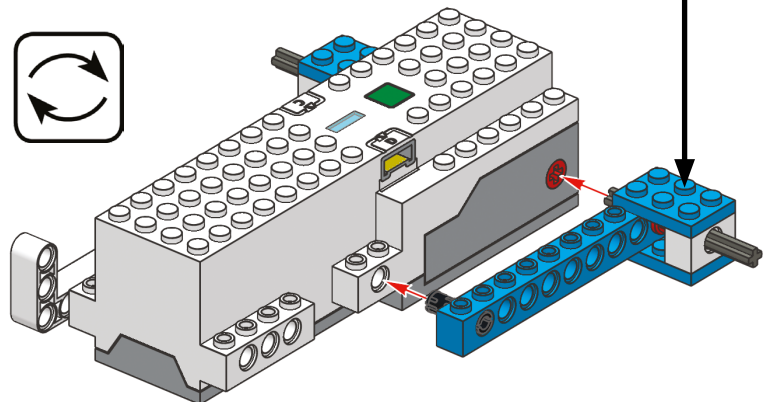
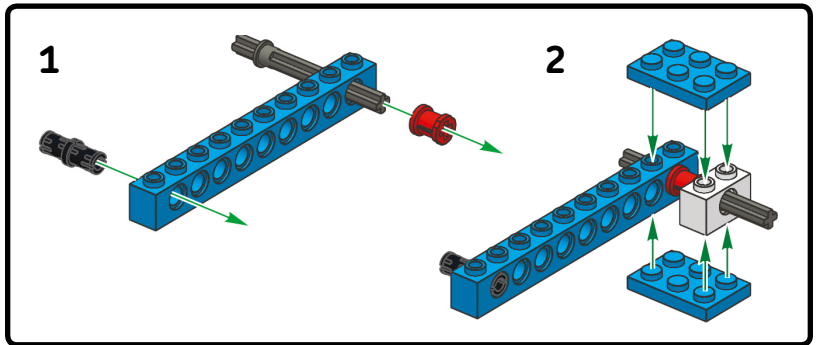
1x



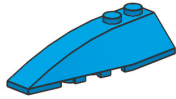
1x



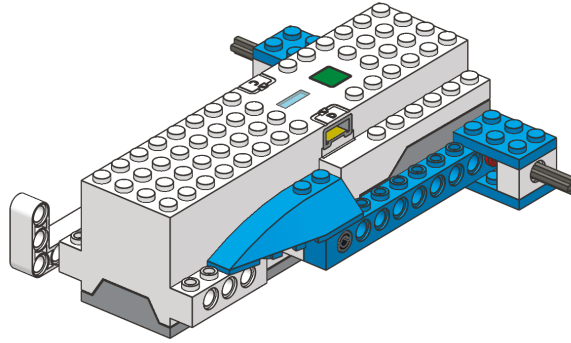
1x



6



1x



7



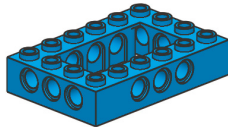
2x



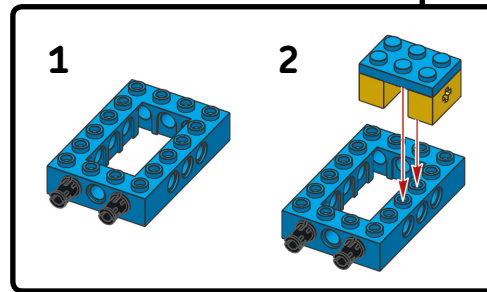
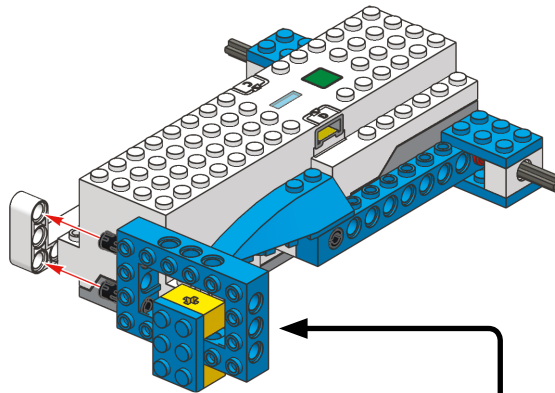
2x



1x



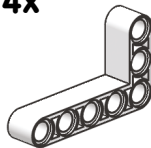
1x



8



4x



1x

