

XLogoOnline – Befehlsübersicht

In diesem Dokument werden alle Logo-Befehle aufgelistet, die in XLogoOnline verfügbar sind. Beachten Sie, dass nicht alle davon für den Unterricht an Primarschulen geeignet und notwendig sind.

Grundbefehle

FD <i>zahl</i>	Bewege die Schildkröte eine Anzahl Schritte vorwärts.
BK <i>zahl</i>	Bewege die Schildkröte eine Anzahl Schritte rückwärts.
RT <i>winkel</i>	Drehe die Schildkröte an der Stelle nach rechts um einen gegebenen Winkel.
LT <i>winkel</i>	Drehe die Schildkröte an der Stelle nach links um einen gegebenen Winkel.
CS	Lösche die Zeichnung und setze die Schildkröte zurück ins Zentrum.

```
rt 90 fd 100
cs
fd 100 rt 90 bk 200 lt 360/4 fd 125.5
```

Arbeiten mit Farben

SETPCfarbe	Die Stiftfarbe wird auf die gegebene Farbe festgelegt.
SETSCfarbe	Die Hintergrundfarbe wird auf die gegebene Farbe festgelegt.

Benutzt werden die folgenden Farbbezeichnungen, Zahlen oder RGB-Farbwerthe {R G B}:

Name	Zahl	{Rot Grün Blau}	
black	0	{0 0 0}	
red	1	{255 0 0}	
green	2	{0 255 0}	
yellow	3	{255 255 0}	
blue	4	{0 0 255}	
magenta	5	{255 0 255}	
cyan	6	{0 255 255}	
white	7	{255 255 255}	
darkgray	8	{128 128 128}	
lightgray	9	{192 192 192}	
darkred	10	{128 0 0}	
darkgreen	11	{0 128 0}	
darkblue	12	{0 0 128}	
orange	13	{255 200 0}	
pink	14	{255 175 175}	
purple	15	{128 0 255}	
brown	16	{153 102 0}	

```
setpc red fd 200 setscc {128 0 255}
```

Kommentare

#text Zeilen, die mit einem Hashtag beginnen, werden als Kommentar verstanden und nicht ausgeführt. Kommentare sind nur im Editor erlaubt. Mehrzeiliger Kommentar: Ctrl+Shift+7

```
fd 100 rt 120
#fd 100 rt 120 → wird nicht ausgeführt
fd 100 rt 120
```

Stift-Manipulation

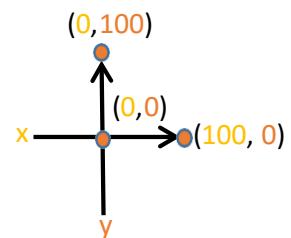
SETPW <i>zahl</i>	Setze die Stiftbreite auf den gegebenen Wert. Der Standardwert ist 1. Achtung: intern ist der Wert 2, um Aliasing-Effekte zu vermeiden.
PU	Hebe den Stift hoch. Die Schildkröte fährt, ohne zu zeichnen.
PD	Setze den Stift ab. Ab sofort werden wieder Linien gezeichnet.
PE	Radiergummi-Modus: Gezeichnete Linien werden ausgeradiert (übermale mit Hintergrundfarbe).
PPT	Beende Radiergummi-Modus
WASH	Lösche alle bisher gezeichneten Linien. Hintergrundfarbe und Position der Schildkröte bleiben erhalten.

```
fd 100 rt 90 fd 100 setsc green
wash
setpw 5 fd 100 pu fd 100 pd
```

Schildkröten-Manipulation

HT	Verstecke die Schildkröte. Sie kann zeichnen, ist aber unsichtbar.
ST	Zeige die Schildkröte, falls versteckt.
SETX <i>x</i>	Setze die x-Koordinate der Schildkröte. Ursprung ist jeweils (0, 0) im Zentrum.
SETY <i>y</i>	Setze die y-Koordinate der Schildkröte.
SETXY <i>x y</i>	Setze x- und y-Koordinate der Schildkröte
HOME	Setze die Schildkröte zurück auf (0, 0), ohne dabei die gezeichneten Linien zu löschen.
SETHEADING <i>z</i>	Setze die Ausrichtung <i>z</i> der Schildkröte manuell fest.

Ausrichtung	Wert
oben	0
rechts	90
unten	180
links	270



```
ht fd 100 st setx 200 fd 100 home
fd 100 setheading 90 fd 100
```

Arithmetische Funktionen

<i>RANDOM max</i>	Generiere eine zufällige natürliche Zahl im Bereich zwischen 0 und max. Bsp: RANDOM 1 generiert 0 oder 1
<i>MOD a b</i>	Berechne den Rest der Division von a geteilt durch b. Bsp: MOD 21 5 ist 1, denn: $21 = 5 \cdot 4 + 1$
<i>POWER a b</i>	Potenziere b mit a: Berechne b^a .
<i>SQRT a</i>	Ziehe die Wurzel von a: Berechne \sqrt{a} .
<i>LOG a</i>	Logarithmus von a zur Basis 10: $\log_{10}(a)$
<i>ABS a</i>	Berechne den absoluten Wert von a. Falls a ein negatives Vorzeichen hat, berechne $a * (-1)$, um den positiven Wert von a zu erhalten.
<i>SIN a</i>	Berechne $\sin(a)$
<i>ARCSIN a</i>	Berechne $\sin^{-1}(a)$
<i>COS a</i>	Berechne $\cos(a)$
<i>ARCCOS a</i>	Berechne $\cos^{-1}(a)$
<i>TAN a</i>	Berechne $\tan(a)$
<i>ARCTAN a</i>	Berechne $\tan^{-1}(a)$

```
print mod 21 5
fd 200 rt 90 fd 200 rt 135 fd sqrt
2*200*200
```

Mathematische Konstanten

<i>PI</i>	Eine auf Maschinengenauigkeit exakte Darstellung der Zahl pi (3.1415)
<i>E</i>	Eine auf Maschinengenauigkeit exakte Darstellung der Eulerzahl e (2.7182)

Interaktion mit der History

<i>PRINT zahl</i>	Schreibe eine Zahl in die History
<i>PRINT [text]</i>	Schreibe Text in die History
<i>CT</i>	Leere die History.

```
print [pi hat den Wert:] print pi
```

Neue Befehle definieren

Neue Befehle definieren wir im Editor. Danach stehen sie uns wie eingebaute Befehle zur Verfügung. Mit Parametern schreiben wir Befehle, die unterschiedliche Bilder zeichnen können. Achtung Namenswahl: Bereits definierte Befehle können nicht „überschrieben“ werden.

Ohne Parameter	Mit Parametern
<i>TO NAME befehle END</i>	<i>TO NAME :PARAMETER befehle END</i>
<i>TO QUADSEITE100 fd 100 rt 90 END</i>	<i>TO QUADRAT :GR repeat 4[fd :GR rt 90] END</i>

Kontrollstrukturen

Ohne die typischen Kontrollstrukturen Schleifen und bedingte Ausführung, werden Programme stets Befehl-für-Befehl von oben nach unten ausgeführt. Schleifen und bedingte Ausführung ändern dies.

Schleifen

Mit Schleifen können Befehle mehrmals wiederholt werden. Logo kennt zwei Schleifen-Arten:

- *REPEAT zahl [befehle]*
Wiederhole die Befehle in den eckigen Klammern so oft es die gegebene Zahl verlangt.
`repeat 4 [fd 100 rt 90]`
- *WHILE [abbruchbedingung][befehle]*
Wiederhole die Befehle in den eckigen Klammern, solange die Bedingungen erfüllt sind.
Achtung: Programme deren Bedingung immer erfüllt ist, stoppen niemals.
`make "a 4
while[:a>0][fd 50 rt 90 make "a :a-1]`

Bedingte Ausführung

Mit dem Keyword IF können wir selektiv Teile unseres Codes ausführen oder nicht. Es gibt zwei Arten dieser bedingten Ausführung:

- *IF (bedingung)[befehle]*
WENN-DANN: Führe die Befehle in eckigen Klammern nur dann aus, falls eine gegebene Bedingung erfüllt ist.
`make "a 1 if(:a=0)[fd 100]`
- *IF (bedingung)[befehle][befehle]*
WENN-DANN-SONST: Falls die Bedingung erfüllt ist, führe den ersten Code-Block aus, sonst den zweiten.
`make "a 1 if(:a=0)[fd 100][bk 100]`

Ausführungssteuernde Befehle

STOP Brich die Ausführung an dieser Stelle ab. Ausserhalb einer Schleife wird das Programm vollständig abgebrochen.

Bug Bei STOP in innerer Schleife einer verschachtelten Schleife bricht nur die innere Schleife ab.

WAIT zahl Pausiere die Ausführung. Wait 100 pausiert die Ausführung für 1 Sekunde.

MAKE "a b Definiere eine Variable mit dem Namen a und dem Wert b. Diese Variable kann durch :a verwendet werden.
Variablen sind ausschliesslich in ihrem Definitionsbereich verfügbar.

```
make "x 50 repeat 50 [fd :x make "x :x-1 print :x]
ht repeat 30 [fd 99 wait 20 wash]
```