

XLogoOnline – Befehlsübersicht

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

Grundbefehle

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

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

Arbeiten mit Farben

- SETPC farbe** Die Stiftfarbe wird auf die gegebene Farbe festgelegt.
- SETSC farbe** Die Hintergrundfarbe wird auf die gegebene Farbe festgelegt.

Benutzt werden die folgenden Farbbezeichnungen, Zahlen oder Farbwerte [R G B] der additiven Farbmischung:

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 setsc [128 0 255]
```

Kommentare

- # text** Zeilen, welche mit einem Hashtag (#) beginnen, werden als Kommentar verstanden und nicht ausgeführt (alternativ: „Ctrl+Shift+7“). Nur im Editor (links) sind Kommentare erlaubt.

```
fd 100 rt 120
# fd 100 rt 120 --> wird nicht ausgeführt
fd 100 rt 120
```

Stift-Manipulation

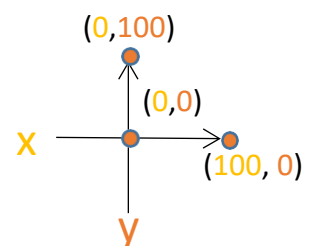
- SETPW zahl** Setze die Stiftbreite auf den gegebenen Wert. Standard ist 1 (Achtung: intern effektiv 2, da sonst Aliasing-Effekte auftreten).
- PU** Hebe den Stift hoch, die Schildkröte fährt, ohne zu zeichnen.
- PD** Setze den Stift ab - ab sofort werden wieder Linien gezeichnet.
- PE** Radiergummi-Modus: Gezeichnete Linien werden ausgeradiert (übermale mit der Hintergrundfarbe).
- PPT** Beende Radiergummi-Modus.
- WASH** Lösche alle bisher gezeichneten Linien. Die Hintergrundfarbe bleibt erhalten, ebensodie Position der Schildkröte.

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

Schildkröten-Manipulation

- HT** Die Schildkröte wird versteckt. Sie steht an derselben Stelle und kann zeichnen, allerdings sieht man sie dabei nicht.
- ST** Zeige die Schildkröte, falls versteckt.
- SETX x** Setze die x-Koordinate der Schildkröte. Ursprung ist jeweils (0, 0) im Zentrum
- SETY y** Setze die y-Koordinate der Schildkröte
- SETXY x y** Setze x- und y-Koordinate der Schildkröte
- HOME** Setze die Schildkröte zurück auf (0, 0), ohne dabei die gezeichneten Linien zu löschen.
- SETHEADING z** Setze die Ausrichtung z 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

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

```
fd 200 rt 90 fd 200 rt 135 fd sqrt 2*40000
```

Mathematische Konstanten

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

Interaktion mit der History

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

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

Neue Befehle definieren

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

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

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, wie die gegebene Zahl es verlangt.
`repeat 4 [fd 100 rt 90]`
- `WHILE [abbruchbedingung] [befehle]`
Wiederhole die gegebenen Befehle, solange die gegebene Bedingungen erfüllt sind. Achtung: Programmieren Bedingung immer erfüllt ist, werden in Folge niemals stoppen.
`while [:a>0] [fd 100 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 (wenn-dann und wenn-dann-sonst):

- `IF (bedingung)[befehle]`
WENN-DANN: Führe die Befehle in 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` Breche die Ausführung an dieser Stelle ab. Ausserhalb einer Schleife wird das Programm vollständig abgebrochen. **Bug** In verschachtelten Schleifen bricht nur die Schleife ab.
- `WAIT zahl` Pausiere die Ausführung. Hier gilt: 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 anschliessend verwendet werden durch :a. Variablen sind ausschliesslich verfügbar in ihren Definitionsbereichen (dem jeweiligen Befehl).

```
make "x 100 repeat 50 [fd :x make "x :x+1]  
repeat 100[fd 99 bk 99 wait 10 wash fd 10]
```