




# COMPUTERCRAFT 1.73

[Show more](#)

## BUILT-IN PROGRAMS

**label** *(breaking without a label loses all data and fuel)*  
**monitor** *side program arguments...*  
**bg / fg** *program arguments... (advanced / multishell)*  
**repeat** *(wireless repeater)*  
**gps host** *[x y z]*  
**gps locate**  
**refuel** *[amount]*  
**craft**  
**go** *[direction]*  
**equip / unequip** *side*  
**dig** *[distance]*  
**excavate** *[square size]*

## SHORTCUTS

 Shutdown computer  
 Reboot computer  
 Terminate running program  
Boot sequence: disk/startup, startup

## TURTLE API

forward / back / up / down()  
turnLeft / turnRight()  
select(slot) / getSelectedSlot()  
craft([quantity])  
getItemCount / getItemSpace()  
getItemDetail()  
*Returns: {name, count, damage}*  
equipLeft / equipRight()  
attack / attackUp / attackDown()  
dig / digUp / digDown() *(or till dirt)*  
place / placeUp / placeDown([signText]) *(or use bucket)*  
detect / detectUp / detectDown()  
inspect / inspectUp / inspectDown()  
*Returns: success, {name, metadata}, error*  
compare / compareUp / compareDown()  
compareTo()  
drop / dropUp / dropDown()  
suck / suckUp / suckDown([amount])  
refuel([quantity])  
getFuelLevel() / getFuelLimit()  
transferTo(slot, [quantity])

## DEFAULTS

Fuel limit (moves): 20,000 normal / 100,000 advanced  
Modem distance, wired (blocks): 256  
Modem distance, wireless (blocks): 64+ (16+ in thunderstorm)  
Resolution (chars): 51x19 chars computer / 7x5 monitor

#### DISK API

isPresent(*side*) / hasData(*side*) / hasAudio(*side*)  
setLabel(*side*, *label*) / getLabel(*side*) / getID(*side*)  
playAudio(*side*) / stopAudio(*side*)  
eject(*side*)

#### ETHERPAD.ZA3K.COM BOOTSTRAP

pastebin get LLia5Cd7 etherpad  
etherpad get etherpad etherpad  
etherpad get pull pull  
pull push

#### FILE HANDLES

close()  
readLine() / readAll() ("r")  
write / writeLine(*data*) ("w" or "a")  
flush() ("w" or "a")  
read() / write(*byte*) ("rb" or "wb")

#### FS API

open(*path*, *mode*)

#### GPS API

locate(*timeout*)

#### HELP API

topics()  
lookup(*topic*)

#### HTTP API

get(*url*, [*headers*]) (sync)  
post(*url*, *postData*, [*headers*]) (sync)  
request(*url*, [*postData*, [*headers*]])  
*Returns: nil, http\_success or http\_failure event sent later*

#### MULTISHELL API

getCurrent()  
setTitle(*tabID*, *title*) / getTitle(*tabID*)

#### OS API

computerID() / computerLabel() /  
setComputerLabel(*label*)  
clock() (*Monotone since boot*)  
time() / day()  
sleep(*seconds*)  
setAlarm(*time*) / cancelAlarm(*alarm*)  
setTimer(*seconds*) / cancelTimer(*timer*)  
pullEvent([*target-event*])  
queueEvent(*event*, *params...*)

#### NATIVE OS EVENTS

**key:** keycode  
**timer / alarm:** id  
**redstone** *(Any redstone input changed)*  
**disk / disk\_detach:** side  
**modem\_message:** side, frequency, replyFrequency, message, distanceTravelled  
**peripheral / peripheral\_detach:** side  
**mouse\_click:** button, x, y  
**mouse\_drag:** button, x, y  
**monitor\_touch:** side, x, y *(Right click on an advanced monitor)*  
**turtle\_inventory** *(Inventory changes)*

#### PERIPHERAL API

`getType(side)`  
`getMethods(side)`  
`wrap(side)`  
*Returns: peripheral table*  
*(A wired modem can connect peripherals indirectly)*

#### PRINTER PERIPHERAL

`newPage() / endPage()`  
`write(text)`  
`getPageSize() / getCursorPos() / setCursorPos(x, y)`  
`getPaperLevel() / getInkLevel()`

#### REDNET API

`open / close / isOpen(side)` *(for networking)*  
`send(receiverId, message, [protocol])`  
`broadcast(message, [protocol])`  
`receive([protocolFilter], [timeout])`  
`host / unhost(protocol, hostname)`  
`lookup(protocol, [hostname])`

#### REDSTONE

`getSides()`  
`getInput(side)`  
`setOutput(side) / getOutput(side)`  
`getAnalogInput(side) / setAnalogOutput(side)`

#### SHELL API

`run(command, args...)`  
`openTab(command, args...)`  
`switchTab(tabID)`

#### TERM API

`write(text)`  
`clear()` / `clearLine()`  
`getCursorPos()` / `setCursorPos(x, y)`  
`isColor()`  
`getSize()`  
`scroll(n)`  
`redirect(target)`  
`current()` / `native()`  
`setTextColor(color)` / `setBackgroundColor(color)`  
`setTextScale(scale)` (*monitor only*)  
`window.setVisible(visible)`  
`window.getPosition()`  
*Returns: (x,y) of top-left*  
`window.reposition(x, y, [width, height])`

#### TEXTUTILS API

`formatTime(time, [twentyFourHour])`  
`serialize(data)` / `unserialize(serializedData)`  
`serializeJSON(data)`  
`urlEncode(urlUnsafeString)`

#### VECTOR API

`new(x, y, z)`  
`v1:dot(v2)` / `v1:cross(v2)`  
`v1:normalize()` / `v1:length()`  
`v1:round()`

#### WINDOW API

`create(parentTerm, x, y, width, height, [visible])`

#### LUA REFERENCE

`IF {condition} THEN {block} ELSE {block} END`  
`WHILE {condition} DO {block} END`  
`REPEAT {block} UNTIL {condition} END`  
`FOR i = begin, end, [step] DO {block} END`  
`FOR k,v IN pairs(table) DO {block} END`  
`FOR i,v IN ipairs(array) DO {block} END`  
`FUNCTION name(args) {block} END`  
`name = FUNCTION(args) DO {block} END`  
`table.__index` / `table.__newindex`  
Example values: `nil`, `True`, `False`, `{}`, `3`, `"yes"`, `{"yes", 3}`,  
`{a="a", b=3}`

#### SPECIAL THANKS

To the ComputerCraft Wiki for the recipe graphics, as well as excellent documentation.