--:! { turtle: []: (:smiley:, commands: []: (:smiley: } <- **Mock Functions Library** (for out-of-game test) -> muse/docs//lib/mock.md

--:neutral_face: mock: *Libraries to provide a limited simulation of turtle and command computer in-game operations.* -> mock

--:+ turtle: *Generally just returns success but allows for simulated blocking, fuel consumption, refueling.*

--:+ commands: *Mock returns from command computer and report success.*

--:# *Mock turtle has coal, torches, fences, dirt, ladders, wheat seeds, reeds, sand, and saplings.*

--:+ *As well as barrels, granite, potatoes, carrots, beetroot seeds, and kelp.*

--:+ *And advanced computers, ender modems, disk drives and floppies in its slots.*

--:# *Inspection shows dirt for down direction, otherwise nothing.*

--:# turtle.block(blocked: ^:) -> *Sets blocked status to* `blocked` *for debug* -> blocked: `^:`

--:# *Turns just report success, other motions decrement the simulated fuel and report success unless blocked.*

--:+ *Only forward motion can be blocked in simulation.*

--:# *Primitive turtle operations are mocked: attack, dig, place, detect, compare, drop, and suck.*

--:# *A few command computer operations are mocked:* `getBlockPosition`, `getBlockInfo`, *and* `exec`.

--:# commands.getBlockPosition(x: #:, y:, #:, z: #) -> *Mock position.* -> `x: #:, y:, #:, z: #`

--:# commands.getBlockInfo(_: _: , _: : ,: _: , resultTable: {:}) -> *Mock info.* -> `resultTable: {:}`

--:# commands.exec(_: _:, result: any) -> *Mock exec command.* -> `result: any`