

コマンドの可能性を無限に広げる /execute

[/execute]の概要

[/execute] は、ほかのコマンドを拡張するためのコマンドで、主な役割は「コマンドの実行者や基準座標を、別の何かに変更する」ことだ。たとえばコマンドブロックに記した内容は、そのままだと「実行者=コマンドブロック」「基準座標=コマンドブロックの座標」となる。だが[/execute]でエンティティや座標を指定してから続けてコマンドを入力すると、実行者や基準座標を指定したものに移動させた状態で実行されるようになるのだ。ほかにもコマンドの実行条件を設定し、条件に合ったときだけ実行させることも可能で、コマ

ンドを使った遊びかたを大きく広げてくれる。ただ、このコマンドは仕組みを理解するのがちょっと難しい。特にVer.1.19.50以降は、14種類のサブコマンドで制御する方式に変更されたため、さらに複雑になった。とはいえ、ギミックを作るには避けて通れないのも事実。そこで、ここでは多用するであろう4つのパターンを解説する。このコマンドはかなり異質なので、他のコマンドとちよつと異なる形式で解説しているが、これを入りに理解を深めていこう。

[as]——実行者を変更する

`/execute as @e[type=villager]`

コマンド サブコマンド

実行者を指定するターゲットセレクター。ここでは村人を指定

`run ride @s start_riding @e[type=mincart]`

サブコマンド。実行させたいコマンドの前に必ず入力する

実行させたいコマンド内容。ここでは実行者(@s=村人)をトロッコに乗らせる

Note:

「as」は、コマンドの実行者を別のエンティティに変更するサブコマンドだ。これ単体では基準座標は変化しない。上記の例をコマンドブロックに入力すると、基準座標はコマンドブロックの位置を維持したまま、実行者がコマンドブロックから村人に変更され、「run」以降のコマンド効果で村人がトロッコに乗るという結果になる。ちなみに、ほとんどのコマンドは効果対象を選べるため「at」と比べると使いどころは

少ない。たとえば上記の例では「@s」を「@e[type=villager]」に置き換えれば「execute」を使わずとも同じ結果を得られる。ただ、実行者を移しておくことと構文で「@s（実行者を意味するターゲットセレクター）」が使えるため、場合によってはコマンドの短縮につながる。また、実行者と基準座標をセットにしておくこと、その後の構文を考えやすいメリットもあるので覚えておこう。

[at]——基準座標を指定したエンティティの現在位置に変更する

`/execute at @e[type=armor_stand]`

コマンド サブコマンド

基準座標にしたいエンティティのターゲットセレクター。ここでは防具立てを指定

`run particle minecraft:crop_growth_emitter`

サブコマンド。実行させたいコマンドの前に必ず入力する

実行させたいコマンド内容。ここではパーティクルを表示する

Note:

「at」は、コマンドの基準座標を、指定したエンティティの現在地に変更するためのものだ。これ単体では実行者は変化しない。上記の例をコマンドブロックに実行させると、基準座標がコマンドブロックから防具立ての位置に移動し、防具立ての足元にパーティクルが表示される。また「at」は連鎖的に使うことができ、「run」

より手前を「/execute at @e[type=armor_stand] at @e[type=villager,c=1]」にすると、防具立てに最も近い村人の現在位置が基準座標になる。この基準座標の連鎖は、コマンドを思い通りに実行させるのにとても重要なので理解しておきたい。

[positioned]——基準座標を指定した座標に変更する

`/execute at @e[type=armor_stand]`

コマンド

基準座標を防具立てに変更

`at @e[type=villager,c=1] positioned ~ ~2 ~`

基準座標を防具立てに一番近い村人に変更

サブコマンドと座標。ここでは基準座標(村人)の2マス上を指定

`run particle minecraft:crop_growth_emitter`

サブコマンド。実行させたいコマンドの前に必ず入力する

実行させたいコマンド内容。ここではパーティクルを表示する

Note:

「positioned」は「at」と同じく基準座標を変更するためのサブコマンドで、こちらは座標そのものを指定したいときに使う。単体でも使えるが、例文のように「at」と連鎖させて使うことの方が多いかも。この例文では、基準座標がコマンドブロック→防具立て→防具立てに一番近い村人に移ったのち、「positioned」

による指定でその村人の頭上へと変更され、「run」以降のコマンドが実行（パーティクルを表示）される。ちなみに、派生形で「positioned at」というサブコマンドもあるが、こちらはエンティティの現在位置を指定するタイプであり、ただの「at」と内容が同じなので覚えなくてよい。

[if] [unless] 実行条件を指定する

`/execute as @e[type=cow] at @s if block ~~-1 ~`

コマンド 実行者と基準座標を牛に移動 ブロックを調べるサブコマンドとその座標。
ここでは基準座標の1マス下を調べる

`grass 0 run summon lightning_bolt`

調べたいブロックのIDとデータ値 実行させたいコマンド内容。ここでは実行者が雷を
召喚する。文頭に必ず「run」を入力すること

Note: 「if」 と 「unless」 は、コマンドの実行条件を設定できるサブコマンドだ。単体でも使えるが、例文のように他のサブコマンドと組み合わせることもできる。この例文は、実行者と基準座標を牛に移したのち、基準座標の1マス下が草ブロックだった場合のみ、コマンドを実行（牛が雷を召喚）するという内容だ。文中の「if」を「unless」に書き換えると実行条件が逆転し、草ブロックの上にはいない牛が雷を召喚するようになる。また「block」以外にもエンティティの有無で判定する「entity」、ブロック構成の差異で判定する「blocks」、スコアの比較で判定する「score」を用いることもできる。「entity」は「testfor」、「blocks」は「testforblocks」と似た性質をもつのでそれぞれのページを参考にしてほしい。「score」は少々複雑なので本書では割愛する。

■その他のサブコマンドの概要

サブコマンド	概要
align	座標の小数点以下を切り捨て、マスの中央を指定するためのサブコマンド。x, y, zのどれを切り捨てるかを選べる。 例) 自分のスポーンポイントを、現在位置の中央に定める。 /execute align xyz run spawnpoint @s ~ ~ ~
anchored	コマンド構文でローカル座標を用いるときに使うサブコマンド。ローカル座標の基準を目線の高さにするか、足元にするか選べる。 例) 実行者の目線の高さで、向いている方向に5ブロック分テレポートする。 /execute anchored eyes run teleport ^ ^ ^5
facing	指定した座標、またはエンティティの方向を向いているものとして実行させるサブコマンド。 例) 村人を最寄りのプレイヤーに向かって1ブロック近づかせる。 /execute as @e[type=villager] at @s facing entity @p eyes run tp @s ^ ^ ^1
in	コマンドを実行するディメンション(オーバーワールド、ネザー、ジ・エンド)を選択するサブコマンド。 例) 現在の座標を基準に、ジ・エンドにテレポートする。 /execute in the_end run tp ~ ~ ~
rotated	エンティティの向きを指定しつつコマンドを実行させられるサブコマンド。向きの指定は水平・垂直方向への角度を入力するか、もしくは基準座標を持つエンティティに合わせさせる。 例) 自分が向いている方向に、すべての羊を5ブロック分テレポートさせる。 /execute as @e[type=sheep] at @s rotated as @p run teleport @s ^ ^ ^5
run	実行させたいコマンド構文の直前に入れるサブコマンド。「run」の後ろに続く構文が、実際に実行される内容になる。 例) プレイヤー名「SOCYM」の正面にアレイを召喚する。 /execute at @a[name=SOCYM] run summon allay ^ ^1.5 ^1

■旧Bexecuteを再現したい場合の書式(実行者・基準座標の移動のみ)

`/execute as <origin:target> at @s positioned <position:x,y,z> run`

例) プレイヤー名「SOCYM」の正面にアレイを召喚する
旧 /execute @a[name=SOCYM] ~ ~ ~ summon allay ^ ^1.5 ^1
新 /execute as @a[name=SOCYM] at @s positioned ~ ~ ~ run summon allay ^ ^1.5 ^1

コマンド 11 /fillを使って特定のブロックで埋め尽くす

[/fill]の概要

「/fill」は、座標で指定した直方体の範囲を、指定したブロックで埋め尽くすコマンドだ。空気も指定できるので、大規模な整地にも役に立つ。また「/fill」にはモードコマンドが5種類(replace, destroy, hollow, keep, outline)あり、これらを使い分けることで埋め尽くすだけでなく、指定

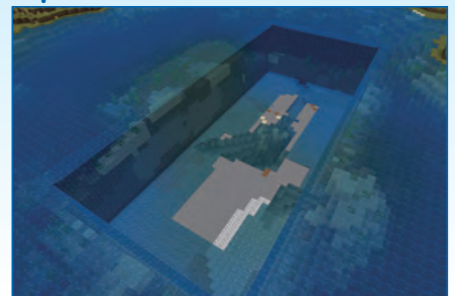
したブロックだけを別のブロックに置換したり、中身を空洞化して大きな箱を作ったりすることが可能だ。シンプルなコマンドの割に使い勝手がよく、とくに建築において役立つのでぜひマスターしよう。

replace(置換)



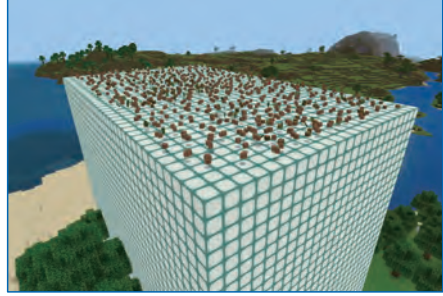
変更元と変更後のブロックを指定し置き換えるモードコマンド。建物を丸石から砂岩に変更したいなどの場面で役に立つ。

replace(省略)



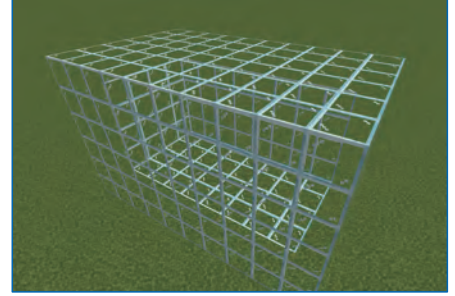
基本構文①で「replace」以下を省略すると指定範囲を指定ブロックで埋め尽くす。整地や水抜きなどに向けた構文だ。

destroy



指定範囲をすべて埋め尽くしつつ、範囲内にあるブロックをアイテム(エンティティ)として回収できるモードコマンド。

hollow



指定範囲の外側1マス分だけを指定ブロックで埋め尽くし、内側は空気に置換する。大きな箱型を作りたいときに活躍する。