Minecraft Data Packs

Notes & Code Snippets

Start

April 2, 2025

Author

Paul Beggs
PaulBeggs03@gmail.com

End

TBD

TABLE OF CONTENTS

1.1 Commands

Commands are the primary way to interact with the Minecraft game world. They allow players to perform various actions, such as changing the time of day, teleporting, or modifying the properties of entities. Commands can be executed in the game chat or through command blocks. They are typically prefixed with a forward slash (/) and can take various arguments to specify the action to be performed.

For each command, it is likely that you will have to use *target selectors*. These include `@a` (all players), `@e` (all entities), and `@r` (random player). As stated, these selectors are used with most commands. Other common syntax to use with other commands includes:

- `<target selector>` is used to specify the target of the command. This can be a player, entity, or location.

- `<item>` is used to specify the item or block to be affected by the command. This can be a specific item ID or a block name.

  - You will always want to prefix the item you are using with `minecraft:`. For custom items, you will want to prefix the item with `namespace:item_name`.

- `[nbt]` is used to specify the NBT (Named Binary Tag) data for the item or block. This can be used to specify custom properties for the item or block. We will touch on this in more detail in subsection 1.1.2.

Notice the use of square brackets. This indicates that the argument is optional. The other commands that have angled brackets are required arguments. For the following commands, each will be notated with either `<...>` or `[...]` to indicate whether the argument is required or optional.

1.1.1 Common Commands

The /give Command

The `/give` command is used to give an item to a player. It can be used to give any item in the game, including blocks, tools, and food. The syntax for the command is as follows:

```
/give <player> <item> [amount] [nbt]
```

For this command, the `<player>` argument is required, while the `<item>` argument is also required. Notice that the command specifies a *player*, and not a *target selector*. The other argument, `[amount]`, is optional and specifies the number of items to give. It defaults to

`1` if not specified. It used to be the case that to get a specific type of wood (for example, Spruce), you would have to specify the data value of the item. However, this is no longer the case. The command now looks like this:

```
/give @p minecraft:spruce_planks 64
```

The /summon Command

The `/summon` command is used to summon an entity in the game world. This can be used to create mobs, items, and other entities. The syntax for the command is as follows:

```
/summon <entity> [x] [y] [z] [nbt]
```

The `[x]`, `[y]`, and `[z]` arguments specify the coordinates where the entity will be summoned. If not specified, the entity will be summoned at the player's location. You can use tildas ($\sim$) to specify relative coordinates. For example, `/summon minecraft:creeper ~ ~1 ~` will summon a creeper one block above the player's location.

In a nutshell, here is a list of some of the most common commands you will use in Minecraft:

Command	Description	Example
/execute	Allows one command to be run *as* or *at* another entity, enabling complex logic and event-driven interactions.	

Table 1.1

1.1.2   Filters and NBTs

Notice from the previous section, we were talking about the entity selectors. The `@e` selector is used to target all entities in the game world. It would be quite annoying if for every time you wanted to remove a single entity, it removed all of them. Thus, this is where *filters* come in. Filters are used to narrow down the selection of entities based on specific criteria. For example, you can use filters to target entities based on their type, name, or other properties. This allows for more precise control over which entities are affected by a command.

A very common application of filters is to target entities based on their NBT (Named Binary Tag) data. NBT data is a way to store complex data structures in Minecraft, and it is used to define the properties of entities, items, and blocks. For example, you can use NBT data to specify the health of a mob, the enchantments on an item, or the custom name of an entity.

NBT data is typically represented in a JSON-like format, and it can be used in commands to filter entities based on their properties. For example, you can use the command `/kill @e[type=minecraft:pig, nbt={CustomName:"My Pig"}]` to kill only pigs with a custom name of "My Pig". This allows for very specific interactions with entities in the game.

1.2 Entities and Properties

1.2.1 Entities

Entities are objects that exist in the game world. They can be anything from players, mobs, and items to projectiles and vehicles. Each entity has its own set of properties that define its behavior and appearance. For example, a player entity has properties like health, position, and inventory, while a mob entity has properties like AI behavior, health, and drops.

These entities can be manipulated using commands and functions in Minecraft. For example, you can summon an entity using the `/summon` command, or you can modify an entity's properties using the `/data` command. Probably the most important entity for data pack coding is the armor stand (see section 2.1), which can be used to create floating text and other visual effects in the game.

2.1 Floating Text

According to this video, Floating Text in Minecraft, we need to identify the block that the armor stand will be standing on. The armor stand will always generate at the bottom of the block. (Note that the armor stand will generate as a floating entity, so it does not need a block to stand on.)

Look at a block, and type `/setblock` to get the coordinates of the block you're looking at. Copy those coordinates, and type:

```
summon minecraft:armor_stand [the coordinates] {Marker:true,
↪  Invisible:true, CustomNameVisibile:true, CustomName:{"text": "[your
↪  text]", "color": "[your color]"}}
```

For example, the code:

```
summon minecraft:armor_stand -67 104 55 {Marker:true, Invisible:true,
↪  CustomNameVisible:true, CustomName:{"text": "You won!", "color":
↪  "yellow"}}
```

will create something like this:
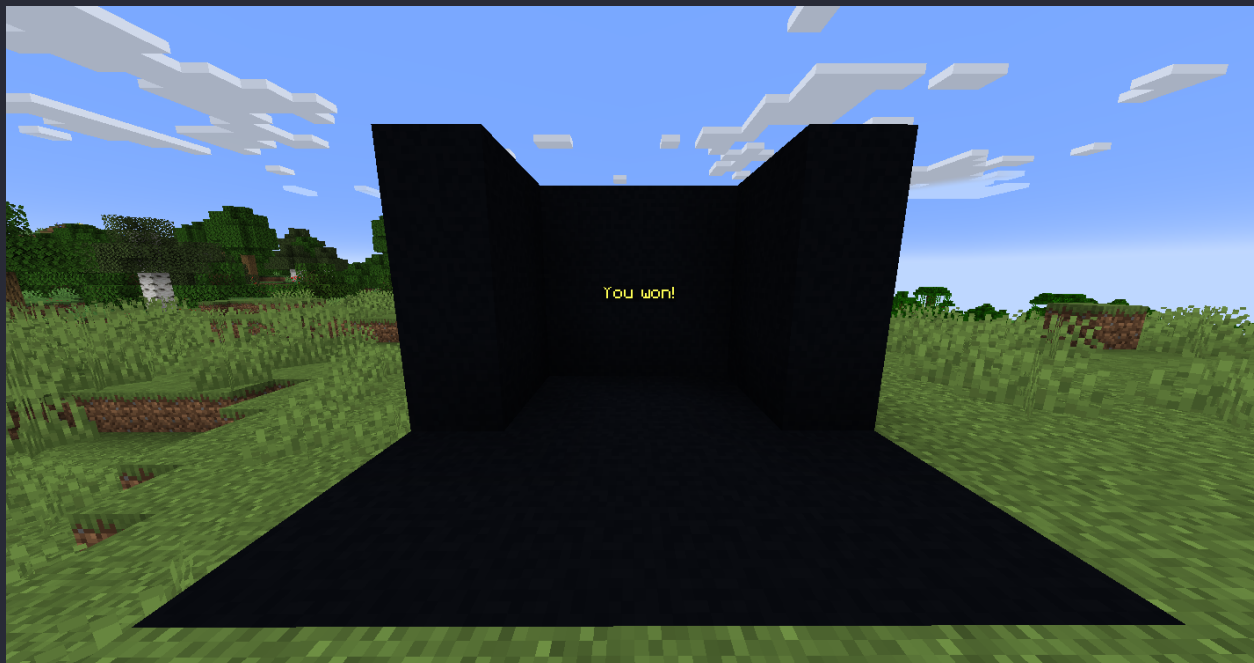
2.1.1   Adding Italics and Bold

To add italics and bold to the text, you can use the `italic` and `bold` tags. For example, the code:

```
summon minecraft:armor_stand -67 104 55 {Marker:true, Invisible:true,
↪  CustomNameVisible:true, CustomName:{"text": "You won!", "color":
↪  "yellow", "italic": true, "bold": true}}
```