

Java 專案：NBT

TYIC 桃高資訊社

NBT

NBT(named binary tag)

是 **Minecraft** 中幾乎所有資料的儲存格式

NBT 最常見的形式是以字串呈現的 **SNBT(stringified NBT)**

兩者可以互相轉換

NBT 共有 **13** 種資料型別

幾乎可以與 **Java** 中的資料型別對應

SNBT 的寫法也與 **JSON** 非常相似

SNBT

NBT 資料型別	Java 資料型別	備註
位元組(byte)	byte	SNBT 中使用 "<number>b" 表示，如 1b
布林(boolean)	boolean	SNBT 中的 false、true 可以分別與位元組 0b、1b 互通
短整數(short)	short	SNBT 中使用 "<number>s" 表示，如 5s
整數(int)	int	如 666
長整數(long)	long	如 -987651
單倍精度浮點數(float)	float	如 2.718f
雙倍精度浮點數(double)	double	如 3.14159265358
字串(string)	java.lang.String	SNBT 中可使用 一對單引號(' ')表示字串 如 'tyic'、"tysh"

SNBT

NBT 資料型別	Java 資料型別	備註
位元組陣列(byte array)	<code>byte[]</code>	SNBT 使用 <code>[B;byte1,byte2,...]</code> 表示 如 <code>[B;-10b,false,true]</code>
整數陣列(int array)	<code>int[]</code>	SNBT 使用 <code>[I;int1,int2,...]</code> 表示 如 <code>[I;1111,10,-5]</code>
長整數陣列(long array)	<code>long[]</code>	SNBT 使用 <code>[L;long1,long2,...]</code> 表示 如 <code>[L;1314,-520,888]</code>
串列(list)	<code>java.util.List<T></code>	SNBT 使用 <code>[element1,element2,...]</code> 表示 如 <code>[1b,true,false]</code> 串列中所有元素的型別需相同 須注意此與上方的任何陣列不相等
複合資料(compound)	<code>java.util.Map<String,?></code>	SNBT 使用 <code>{key1:value1,key2:value2,...}</code> 表示 鍵類似於字串，但可不加引號 值可以為任意型別 如 <code>{School:'TYSH',"Since":1941S}</code>

SNBT

SNBT 範例如下：

```
{lit_total_time: 1600s, cooking_time_spent: 159s, x: -1530, y: 79, z: -2005, cooking_total_time: 200s, Items: [{count: 62, Slot: 0b, id: "minecraft:raw_iron"}, {count: 63, Slot: 1b, id: "minecraft:coal"}, {count: 2, Slot: 2b, id: "minecraft:iron_ingot"}], id: "minecraft:furnace", lit_time_remaining: 1042s, RecipesUsed: {"minecraft:iron_ingot_from_smelting_raw_iron": 2}}
```

snbt

物品堆疊元件

物品堆疊元件(`item stack component`、`data component`)

是用於儲存物品堆疊固定(已知)會有的額外資料

如界伏盒內的物品、耐久度、最大耐久度、其餘自訂資料等

物品堆疊元件為鍵值映射

且鍵需要被註冊，值則為符合指定資料型別的任何值

物品堆疊元件的字串型式為 `[key1=value1,key2=value2,...]`

但其最終會被轉換為 `NBT` 的複合資料

以 `SNBT` 表示為 `{key1:value1,key2:value2,...}`

物品堆疊元件介面為 `net.minecraft.component.ComponentType<T>`

物品堆疊元件

範例：製作一個物品「**TNT 遙控器**」

對一個 **TNT 方塊** 右鍵後便會綁定該 **TNT**

再對**空氣**或其他非 **TNT 方塊** 右鍵後

便會點燃之前所綁定的 **TNT**

因 **TNT 遙控器** 需紀錄所綁定的 **TNT**

故需要使 **TNT 遙控器** 有能記錄 **TNT 座標** 的物品堆疊元件

物品堆疊元件

ComponentType<T> 建造者由呼叫靜態方法 **<T>builder()** 取得
並且需要設定編解碼器(**codec**)
以用於資料**序列化(serialize)**和**反序列化(deserialize)**
大多數**類別**中都有靜態欄位 **CODEC** 可供使用

```
package org.tyic.tyicmod;

import (...)

public class ModDataComponentTypes {
    public static final ComponentType<BlockPos> BLOCK_POS = register("block_pos", BlockPos.CODEC);

    public static <T> ComponentType<T> register(String id, Codec<T> codec) {
        return Registry.register(Registries.DATA_COMPONENT_TYPE, Identifier.of(TyicMod.MOD_ID, id),
            ComponentType.<T>builder().codec(codec).build());
    }

    public static void init() {
        TyicMod.LOGGER.info("Registering mod data component types.");
    }
}
```

ModDataComponentTypes.java

方塊實體

方塊實體(block entity、tile entity)

是用於儲存方塊狀態之外的任意資料

並且每刻(tick, 20 刻 = 1 秒)皆會更新

因此常用於一些能存放東西的方塊，或每刻都要執行功能的方塊

如儲物箱、熔爐、釀造台、日光感應器、海靈核心等

方塊實體類別為 `net.minecraft.block.entity.BlockEntity`

方塊實體類型類別為

`net.minecraft.block.entity.BlockEntityType<T>`

我們需要註冊的是方塊實體類型

方塊實體