Java 專案：物品

TYIC 桃高資訊社

物品和物品堆疊

物品(item)是構成 Minecraft 很重要的部分

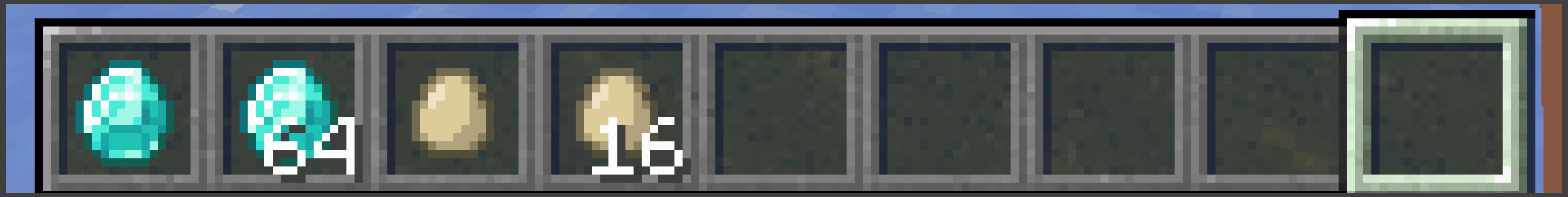
所有物品的類別為 net.minecraft.item.Item

物品堆疊(item stack)則是代表一種物品及擁有的數量(count)

其類別為 net.minecraft.item.ItemStack

如快捷欄(hotbar)的前四格為 4 個不同的物品堆疊

但前兩格物品堆疊的物品都是鑽石，後兩格物品堆疊的物品都是雞蛋



註冊表

幾乎所有東西都要向註冊表(registry)註冊(register)

如物品就需要註冊，但物品堆疊不需要註冊

其目的是為了讓遊戲知道有這東西，以便進行其他處理

每種註冊類別都有獨立的註冊表，並分為靜態註冊表和動態註冊表

靜態註冊表用於註冊永遠不變的項目，只能在遊戲初始化階段註冊項目

動態註冊表用於註冊可能會變更的項目，可以在任何時候註冊項目

註冊表中，不同項目會有唯一的註冊鍵(registry key)

用於區別及檢索註冊表中的不同項目

欲向註冊表註冊項目，需呼叫 `net.minecraft.registry.Registry` 介面的靜態方法 `register`，該方法會返回註冊的項目

所有靜態註冊表位於 `net.minecraft.registry.Registeries` 類別內

標識符

標識符(identifier)由命名空間(namespace)和路徑(path)組成
其中命名空間通常為模組 id
而路徑則由小寫英文、下滑線(_)、斜線(/)組成
在遊戲各處都會使用到標識符
標識符以字串表示為 "namespace:path"
如雞蛋物品的標識符以字串表示為 "minecraft:egg"
其在 Java 中處理的類別為 net.minecraft.util.Identifier
該類別的建構子為 private，需使用下列靜態方法創建實例：

```java
public static Identifier of(String namespace, String path)
```

註冊鍵

註冊鍵(registry key)由註冊表和值組成

其中的註冊表和值為標識符

註冊鍵以字串表示為 "ResourceKey[registry/value]"

如雞蛋物品的註冊鍵以字串表示為

"ResourceKey[minecraft:item/Minecraft:egg]"

註冊鍵除了註冊表相關事項之外幾乎不會使用到

註冊鍵在 Java 中處理的類別為

net.minecraft.registry.RegistryKey

該類別的建構子為 private，需呼叫靜態方法 of 創建實例

所有註冊鍵位於 net.minecraft.registry.RegistryKeys

物品類別

Item 類別只有一個建構子 Item(Item.Settings settings)
Item.Settings 是一個用來控制物品行為的類別
如最大堆疊大小(max stack size，預設為 64)、
描述文字(lore，預設為空)、註冊鍵(預設為空)等
其可以通過方法鏈式呼叫(method chaining)進行設定
用於註冊物品時，一定要設定註冊鍵，範例如下：

```java
RegistryKey<Item> registryKey =
        RegistryKey.of(RegistryKeys.ITEM, Identifier.of(TyicMod.MOD_ID, "example"));
Item.Settings settings =
        new Item.Settings().useCooldown(1.5f).maxCount(2).registryKey(registryKey);
```

簡單物品

最簡單的物品就是直接創建一個 Item 類別的實例並向遊戲註冊
但為了邏輯分離，因此將物品相關事項放在 item 套件下
物品註冊事項放在該套件下的 ModItems 類別
並將註冊過程做成一個函式：

1. 設定 Item.Setting 實例的註冊鍵
2. 向註冊表註冊項目，並將其返回值(註冊項目)作為函式返回值

之後若要引用該物品，如用於比較，只需使用此返回值即可
故註冊後會儲存在公開靜態欄位，方便之後使用

簡單物品

ModItems.registerAll
方法只是為了
加載 ModItems 類別

```java
package org.tyic;

import (...)

public class TyicMod implements ModInitializer {
    public static final String MOD_ID = "tyicmod";

    public static final Logger LOGGER = LoggerFactory.getLogger(MOD_ID);

    @Override
    public void onInitialize() {
        ModItems.registerAll();
    }
}
```

TyicMod.java

```java
package org.tyic.item;

import (...)

public class ModItems {
    public static final Item TyicLogo = register("tyic_logo", Item::new, new Item.Settings());

    public static Item register(String id, Function<Item.Settings, Item> itemFunction, Item.Settings settings) {
        RegistryKey<Item> registryKey = RegistryKey.of(RegistryKeys.ITEM, Identifier.of(TyicMod.MOD_ID, id));
        return Registry.register(Registries.ITEM, registryKey, itemFunction.apply(settings.registryKey(registryKey)));
    }

    public static void registerAll() {
        TyicMod.LOGGER.info("Registering Mod Items.");
    }
}
```

ModItems.java