2021 一小時程式設計 (時間世界) 編碼解決方案

下方介紹的編碼解決方案會為每一個時間分裂，以 **Block** 和 **Python** 的形式，各提供一種初級解決方案和進階解決方案。但是，學生可能會找到多種解決方案，從而達到相同的成功結果。



# 2021 一小時程式設計：編碼解決方案 (MakeCode 方塊)

下方介紹的編碼解決方案會為每一個時間分裂，各提供一種初級解決方案和進階解決方案。但是，學生可能會找到多種解決方案，從而達到相同的成功結果。

**改良程式碼 (Blocks)**

如果你希望 Time Agent 重複執行某個指令，該怎麼辦？

雖然學生可以使用相同的 MakeCode 方塊 15 次，甚至可以複製和貼上以節省時間，但還有一種更有效的編碼方式。在編碼中使用迴圈將能為學生節省大量時間，同時也能讓他們更解編程。

從「迴圈」抽屜中選擇一個「重複」方塊，輸入你希望這些指令重複的次數，然後將這些指令放入重複方塊中。確認方塊的順序正確，並且僅包含你想要重複的方塊！

|  |  |
| --- | --- |
| **時間分裂 1：爵士大樂隊**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  | 初級解決方案也是最進階的解決方案，不過學生可能會找到各種解決方案，達到相同的結果。 |

|  |  |
| --- | --- |
| **時間分裂 2：吉薩大金字塔**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 3：登月任務**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  | 初級解決方案也是最進階的解決方案，不過學生可能會找到各種解決方案，達到相同的結果。 |

|  |  |
| --- | --- |
| **時間分裂 4：中國萬里長城**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 5：蒙娜麗莎**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 6：首次飛行**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 7：第一位電腦科學家**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 8：人類最好的朋友**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 9：古生物學謎題**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  |  |

|  |  |
| --- | --- |
| **時間分裂 10：發現元素**  學生可能會找到多種時間分裂的解決方案，從而達到相同的成功結果。 | |
| 初級解決方案範例 | 更多進階解決方案範例 |
|  | 初級解決方案也是最進階的解決方案，不過請注意，學生可能會找到各種解決方案，達到相同的結果。 |

# 2021 一小時程式設計：編碼解決方案

**(Azure Notebooks Python 程式碼)**

下方介紹的編碼解決方案會為每一個時間分裂，各提供一種初級解決方案和中級解決方案。但是，學生可能會找到多種解決方案，從而達到相同的成功結果。

對於中級代碼，請注意 Python 對標籤和空格非常敏感。「for I in range」迴圈中的所有重複指令**必須** 以完全相同的方式縮排。請使用一個標籤來確保程式碼有效，而不要使用空格鍵上的多個空格。

**爵士大樂隊**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
| C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\jazz_beginner.png | C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\jazz_intermediate.png |

**吉薩大金字塔**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**登月任務**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**中國萬里長城**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
| C:\Users\birving\Documents\HourofCode2021\python_code_screenshots\china_beginner.png |  |

**蒙娜麗莎**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**首次飛行**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**第一位電腦科學家**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  | 由於此時間分裂中移動不易預測，因此建議採用直線序列 (如初級程式碼中所述)。 |

**人類最好的朋友**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**古生物學謎題**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |

**發現元素**

|  |  |
| --- | --- |
| **初級程式碼** | **中級程式碼** |
|  |  |