



使用者指南

AWS DeepRacer 學生



AWS DeepRacer 學生: 使用者指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS DeepRacer 學生？	1
您對 AWS AI & ML Scholarship 計劃感興趣嗎？	1
您是第一次使用 AWS DeepRacer 學生嗎？	1
您是否有興趣在 AWS DeepRacer 學生聯盟中競爭？	2
什麼是 AWS AI & ML Scholarship 計畫？	3
您在 AWS AI & ML Scholarship 中獲勝的項目	3
如何註冊並預先符合 AWS AI & ML Scholarship 計畫的資格？	4
關於 AWS AI & ML Scholarship 計畫和 Udacity 平台的常見問答集 (FAQs)	5
什麼是 AWS 玩家帳戶？	7
為支援的服務建立 AWS 玩家帳戶	7
刪除 AWS 玩家帳戶	7
開始使用	8
先決條件	8
步驟 1：註冊 for AWS DeepRacer 學生	8
步驟 2：完成註冊 AWS DeepRacer 學生	9
步驟 3：(選用) 閱讀並選擇加入 AWS AI & ML 學術考量	10
步驟 4：更新您的設定檔	10
步驟 5：從首頁 Explore AWS DeepRacer 學生	11
訓練強化學習模型	13
步驟 1：使用 AWS DeepRacer 學生訓練強化學習模型	13
步驟 2：為您的模型命名	13
步驟 3：選擇您的賽道	13
步驟 4：選擇演算法	14
步驟 5：自訂獎勵函數	14
步驟 6：選擇持續時間，並將模型提交至排行榜	15
步驟 7：在排行榜上檢視模型的效能	16
步驟 8：使用複製來改善模型	16
步驟 9：(選用) 下載模型	17
參加競賽	18
參加學生聯盟競賽	18
參加學生社群競賽	18
自訂獎勵函數	20
編輯 Python 程式碼以自訂獎勵函數	20
獎勵函數輸入參數	20

all_wheels_on_track	22
closest_waypoints	24
closest_objects	25
distance_from_center	25
heading	27
is_crashed	27
is_left_of_center	28
is_offtrack	28
is_reversed	28
objects_distance	28
objects_heading	29
objects_left_of_center	29
objects_location	29
objects_speed	29
進度	30
speed	30
steering_angle	31
steps	32
track_length	33
track_width	33
x, y	34
waypoints	35
安全	37
資料保護	37
在 AWS DeepRacer 學生入口網站中擷取的資料	38
靜態加密 in AWS DeepRacer 學生入口網站	38
傳輸中加密 in AWS DeepRacer 學生入口網站	38
身分與存取管理	39
法規遵循驗證	39
恢復能力	39
基礎設施安全性	40
排解常見問題	41
為什麼我自動登出 my AWS DeepRacer 學生帳戶？	41
如何選擇退出 AWS AI & ML Scholarship 計畫？	41
我無法刪除 my AWS DeepRacer 學生帳戶	41
我在下拉式清單中找不到我的學校名稱	42

我無法繼續訓練我的模型	42
我收到「帳戶已使用此電子郵件註冊」錯誤訊息	42
我註冊了 Gmail 帳戶，但找不到我的驗證碼	42
配額	43
刪除帳戶	44
文件歷史紀錄	46
.....	xlvii

什麼是 AWS DeepRacer 學生？

AWS DeepRacer Student 是全球高中和大學註冊學習者開發機器學習 (ML) 技能的地方。它提供存取教育資料、選用的 AWS AI 和 ML 學術計劃，以及為 AWS DeepRacer 學生聯盟訓練和測試強化學習 (RL) 模型的機會。若要開始使用，請參閱 [???](#) 章節中的主題。

AWS DeepRacer 學生功能

- 首頁 – 尋找近期活動的詳細資訊、練習訓練 RL 模型、存取 ML 教育內容，以及追蹤模型訓練時數。您也可以從左側導覽窗格管理 AWS DeepRacer 學生設定檔和帳戶資訊。
- 了解 – 存取由 AWS 專家開發的 ML 內容，包括影片。沒有先前經驗的學生可以從 easy-to-understand 的自定進度資料中學習 ML 基礎知識。
- 練習 – 選擇軌跡、演算法和獎勵函數來建立 RL 模型。或者，進行獎勵函數 Python 程式碼的引導演練，然後選擇自訂。使用 AWS DeepRacer 服務，在模擬的 3D 競賽環境中訓練您的模型。複製效能最佳的模型，並在其獎勵函數上反覆運算，以爬升 AWS DeepRacer 學生聯盟排行榜。
- 競爭 – 將您的模型提交至每月虛擬競賽，以獲得成就和獲得獎勵的機會。最佳區域學生競賽者有機會參加學生世界冠軍賽。如需詳細資訊，請參閱 [條款與條件](#)。所有學生每個月都有 10 小時來訓練 AWS DeepRacer 學生聯盟的 RL 模型。只有使用 AWS DeepRacer 學生訓練的模型才能用於 AWS DeepRacer 學生聯盟。

主題

- [您對 AWS AI & ML Scholarship 計劃感興趣嗎？](#)
- [您是第一次使用 AWS DeepRacer 學生嗎？](#)
- [您是否有興趣在 AWS DeepRacer 學生聯盟中競爭？](#)

您對 AWS AI & ML Scholarship 計劃感興趣嗎？

如果您自我識別為服務不足或技術代表性不足，請選擇加入 AWS AI & ML Scholarship 計畫。若要進一步了解誰符合資格、如何申請，以及您獲勝的內容，請參閱 [什麼是 AWS AI & ML Scholarship 計畫？](#)

您是第一次使用 AWS DeepRacer 學生嗎？

如果您是第一次使用 AWS DeepRacer 學生，我們建議您先閱讀以下章節：

- [AWS DeepRacer 學生入門](#)
- [參加 a AWS DeepRacer 學生競賽](#)
- [在 AWS DeepRacer 學生中訓練強化學習模型](#)

您是否有興趣在 AWS DeepRacer 學生聯盟中競爭？

AWS DeepRacer 學生聯盟是一項 AWS 贊助的競賽，學生可以在 3 月到 9 月之間參加每月虛擬競賽，以在國家和區域層級進行競賽，並有機會在 10 月參加學生世界冠軍賽。學生有機會獲得獎勵和獎勵。如需詳細資訊，請參閱 [條款與條件](#)。

若要參加您的第一個競賽，請參閱 [參加 a AWS DeepRacer 學生競賽](#)。

什麼是 AWS AI & ML Scholarship 計畫？

AWS AI & ML Scholarship 計畫作為 AWS DeepRacer Student 的一部分推出，旨在透過為成功的申請人提供機會來獲得最多兩個 Udacity Nanodegrees，將多樣性帶入人工智慧 (AI) 和機器學習 (ML) 領域。Udacity Nanodegree 是大量開放的線上課程 (MOOCs)，旨在彌補學習和職業目標之間的差距。如需詳細資訊，請參閱 Udacity 支援文件中[什麼是 Nanodegree 計畫？](#)。成功的申請人也可以存取獨家活動和指導，以協助他們進一步發展自己的事業。如需詳細資訊，請參閱[您在 AWS AI & ML Scholarship 中獲勝的項目](#)。

此學術課程著重於技術領域中服務不足和代表性不足的人員。申請人必須至少 16 歲，且目前已註冊高中、大學或社區大學。如需預先資格的詳細資訊，請參閱[如何註冊並預先符合 AWS AI & ML Scholarship 計畫的資格？](#)

參與 AWS AI & ML Scholarship 計畫是免費的。

AWS AI & ML Scholarship 計畫已於 2022 年 4 月 11 日正式推出。

AWS AI & ML Scholarship 計畫適用於以群組為基礎的方法。每年會選取兩個 1,000 名學生的群組（總計 2,000 名）。第一個群組的申請日期是 2024 年 5 月 31 日 Udacity，第二個群組的申請日期是 2024 年 9 月 30 日。

您在 AWS AI & ML Scholarship 中獲勝的項目

對於 AWS AI 和 ML 學術計畫，AWS 正在與 Udacity 協作。

每年，AWS AI 和 ML 學術計畫將在全球提供 2,500 個 Udacity Nanodegree 學術，分佈在兩個不同的群組。所有學生都會收到下列項目：

- 免費使用 Python Nanodegree 的 AI 程式設計。本課程教導開始使用 AI 技術所需的基礎技能，並在程式設計、線性代數和神經網路中開發您的技能。
- 在一週內與 Udacity 講師一起存取上班時間，以協助回答有關課程內容的問題。此外，學生可以參與 Udacity 講師主持的每週案例研究練習。
- 獨家存取指導和職業資源，包括業界專家的工作階段、繼續編輯指導和面試秘訣。

此外，AI Programming with Python 類別的前 500 名學生（根據課程評估的效能）有權存取更進階的 Nanodegree，涵蓋深度學習的基礎知識，並為您提供成為機器學習工程師所需的技能。

所有完成學費先決條件並提交 Udacity 應用程式的學生都可以免費存取新的生成式 AI 課程、使用介紹生成式 AI AWS、由 AWS 和 Udacity 共同建立。為期一個月的課程是選用的資源，可協助學生在生成式 AI 領域中進一步提升技能。該課程適用於所有提交學費申請的學生，而 Nanodegree 學費僅適用於學費收件人。

如何註冊並預先符合 AWS AI & ML Scholarship 計畫的資格？

AWS AI & ML Scholarship 計畫適用於 16 歲以上的未受關心和代表性不足的學生。代表性不足和服務不足的學生包括（但不限於）女性、有障礙的人、有色人種（黑人、拉丁人和原住民），以及 LGBTQ+ 社群的成員。

若要開始，學生需要註冊並建立 on AWS DeepRacer 學生帳戶。在註冊時，您可以勾選核取方塊，選擇加入 AWS AI & ML Scholarship 計畫。設定您的帳戶時，您會在首頁上看到一個圖磚，用於追蹤您的學費應用程式預先資格審查任務。學生需要完成兩個先決條件 in AWS DeepRacer 學生，才能提交其學費計劃的申請。

符合 AWS AI & ML Scholarship 資格的學生會收到可在 Udacity 網站上兌換的唯一存取碼，以存取 Udacity Nanodegree 學術申請表。不保證具備預先資格的學生可獲得 Udacity Nanodegree 學費。Udacity 決定哪些符合資格的學生獲得 Udacity Nanodegree 學費。

獲得學費資格的預先資格取決於兩個條件：

檢閱課程和通過評估

若要預先符合資格，您必須在所有必要的評估中取得 80% 或更高的分數。每個評估都以不同的章節為基礎，您可以在 AWS DeepRacer 學生導覽窗格中的學習區段中找到。學術應用程式將於 2024 年 2 月 1 日開啟。

達到最短單圈時間

每個月從 2 月到 9 月，AWS DeepRacer 聯盟都會推出新的排行榜，並推出新的賽道。若要預先符合獲得學費的資格，您需要在任何單一競賽中達到兩分鐘的單圈時間。

學生完成這兩個先決條件後，會收到來自的電子郵件，AWS 其中包含唯一的存取碼和 Udacity Nanodegree 學費的應用程式連結。若要考慮獲得學費，學生需要完整填寫應用程式。當應用程式完成時，Udacity 會聯絡學生，說明註冊選用的介紹生成式 AI 的步驟，並提供 AWS 課程給所有申請人。Udacity 會聯絡獲選獲得 Udacity Nanodegree 學費的申請人。

關於 AWS AI & ML Scholarship 計畫和 Udacity 平台的常見問答集 (FAQs)

什麼是官方 AWS AI 和 ML 學術條款與條件？

若要檢視官方條款與條件，請參閱[官方 AWS AI & ML Scholarship 計畫條款與條件](#)。

該學費的對象是誰？

此 AWS AI & ML Scholarship 計畫適用於世界各地 16 歲以上未受關心和代表性不足的學生。代表性不足和服務不足的學生包括（但不限於）女性、有障礙的人、有色人種（黑人、拉丁人和原住民），以及 LGBTQ+ 社群的成員。

什麼是奈米級？

nanodegree 是一種線上技能型教育計畫，有助於彌補學習和職業技能之間的差距。

如果我預先符合 AWS AI 和 ML 學術資格，我是否會自動獲得 Udacity Nanodegree 學術資格？

否。預先符合 AWS AI 和 ML 學費資格的學生可以存取 Udacity Nanodegree 學費的應用程式。Udacity 決定哪些符合資格的學生獲得 Udacity Nanodegree 學費。

在被選入 AWS AI 和 ML 學術之前，是否需要提供註冊證明？

您可能需要提供註冊證明（例如大學文字記錄），才能獲得 AWS AI 和 ML 學費。

申請 AWS AI & ML 學術計畫時，如何表示我目前已註冊高中？

若要表示您已註冊高中，請參閱 AWS DeepRacer 學生使用者指南[步驟 4：更新您的設定檔](#)中的。

如何保護我的個人身分識別資訊 (PII) 資料？

您的資料在 AWS 雲端受到保護。如需詳細資訊，請參閱[AWS 隱私權聲明](#)。

申請 AWS AI 和 ML 學術是否需要存留期？

是。若要申請 AWS AI & ML Scholarship 計畫，註冊時您必須至少滿 16 歲。

如果我收到 AWS AI 和 ML 學費，如何收到通知？

在應用程式截止日期過後，您會收到來自 Udacity 的電子郵件，告知您的奈米級學費應用程式的狀態。

如果我收到 AWS AI 和 ML 學費，何時會收到通知？

Udacity 將聯絡申請者，告知其學費狀態。獲選獲得學費的學生將獲得重要日期的相關資訊。

如何存取 Udacity 上的 AWS 生成式 AI 課程？

學生需要預先符合 on AWS DeepRacer 學生的資格，並在 Udacity 上提交學費申請。Udacity 在提交完整的應用程式後聯絡學生，並為學生提供註冊使用 AWS 的生成式 AI 課程的步驟。

如果我被 AWS AI & ML Scholarship 計劃接受，會發生什麼情況？

收到 AWS AI & ML Scholarship 的通知後，Udacity 加入團隊的代表將使用您在完成申請時提供給 Udacity 的電子郵件與您聯絡。

如果我獲得 Udacity Nanodegree 學費，我應該預期在奈米級研究上花費多少時間？

學生預期每週會花大約 10 小時在 Udacity nanodegree 研究上。

是否需要英文能力？

雖然英文熟練度並非必要，但學習資料為英文。若要成功，學生應具備良好的英文閱讀和撰寫技能。

如果在競賽/性別清單中找不到我使用 識別的選項，該怎麼辦？

對於兩者，您可以使用我偏好自我識別的選項（選取以輸入）。

什麼是 AWS 玩家帳戶？

AWS 玩家帳戶是由建立的 AWS DeepRacer 多使用者和 AWS DeepRacer 學生的受管身分解決方案 AWS。您的 AWS 玩家帳戶會保留在這些服務中建立的所有資源 AWS。

為支援的服務建立 AWS 玩家帳戶

當您為 [AWS DeepRacer 多使用者](#) 或 [AWS DeepRacer 學生](#) 建立帳戶時，會自動建立 AWS 玩家帳戶。當您在這些服務中使用不同的功能時，新的資源會自動新增至您的 AWS 玩家帳戶。若要開始使用 AWS DeepRacer 多使用者和 AWS DeepRacer 學生，請使用以下連結。

建立 AWS DeepRacer 學生帳戶

若要使用 AWS DeepRacer 學生，請從建立帳戶開始。若要了解如何建立帳戶，請參閱 [AWS DeepRacer 學生使用者指南 步驟 1：註冊 for AWS DeepRacer 學生](#) 中的。

使用 AWS DeepRacer 多使用者在一個帳戶中贊助多個參與者。

AWS DeepRacer 多使用者模式支援兩種不同的使用者設定檔：管理員和參與者。兩者都有不同的設定需求。若要開始使用，請參閱 [AWS DeepRacer 開發人員指南](#) 中的 [多使用者模式](#)。

刪除 AWS 玩家帳戶

如果您刪除 AWS 玩家帳戶，您立即無法存取所有支援的服務。這包括您獲得的任何成就（徽章、點數、頭像等）。

刪除您的 AWS 玩家帳戶不會刪除 AWS 您的帳戶。如果您也想要刪除 AWS 帳戶，請使用 [關閉 AWS 帳戶](#) 中所述的步驟。

如果您已使用 AWS 玩家帳戶在 AWS DeepRacer 多使用者中建立事件，則無法刪除您的 AWS 玩家帳戶。這是為了確保您建立的事件參與者不會留下中斷的體驗。若要進一步了解管理員如何在 AWS DeepRacer 多使用者模式中建立事件，請使用下列主題。

使用 AWS DeepRacer 多使用者模式設定事件（管理員）

若要了解如何使用多使用者模式建立事件，請參閱 [AWS DeepRacer 開發人員指南](#) 中的 [設定多使用者模式（管理員）](#)。

AWS 玩家帳戶無法存取服務帳戶中建立的資源以外的任何 AWS 資源。服務帳戶中的任何 AWS Identity and Access Management 政策和相關資源僅限於必要的資源。

AWS DeepRacer 學生入門

使用此教學課程來開始使用 AWS DeepRacer 學生。本教學課程說明如何登入 to AWS DeepRacer 學生、更新您的設定檔、選擇加入 AWS AI 和 ML 學術考量、開始在機器學習 (ML) 和強化學習 (RL) 中學習免費課程，以及建立 AWS DeepRacer 模型。如果您在建立帳戶期間未選擇加入該學費，您可以在稍後更新設定檔時選擇加入。

主題

- [先決條件](#)
- [步驟 1：註冊 for AWS DeepRacer 學生](#)
- [步驟 2：完成註冊 AWS DeepRacer 學生](#)
- [步驟 3：\(選用\) 閱讀並選擇加入 AWS AI & ML 學術考量](#)
- [步驟 4：更新您的設定檔](#)
- [步驟 5：從首頁 Explore AWS DeepRacer 學生](#)

先決條件

若要存取 AWS DeepRacer 學生並參與，您需要：

- 為至少 16 歲且目前註冊高中、社區大學或大學的學生。
- 或者是高中、大學或社區大學學生的教育工作者或活動主辦人。
- 有效的電子郵件地址。

步驟 1：註冊 for AWS DeepRacer 學生

您可以使用此程序中提供的 URL 註冊 for AWS DeepRacer 學生。註冊時，您可以建立 AWS 玩家帳戶。此帳戶可與某些其他 AWS 服務搭配使用。如果您已經有 AWS 玩家帳戶，您可以將該帳戶與 AWS DeepRacer Student 搭配使用。

1. 開啟 <http://deepracerstudent.com/> 登陸頁面。
2. 選擇開始使用。
3. 在登入頁面上，如果您還沒有 AWS 玩家帳戶，請選擇註冊。

Note

如果您已有 AWS 玩家帳戶，請在此處輸入您的資訊。如需 AWS 玩家帳戶的詳細資訊，請參閱 [什麼是 AWS 玩家帳戶？](#)

4. 在註冊頁面上，輸入下列資訊：
 - 電子郵件地址
 - 密碼
5. 選擇註冊。包含確認碼的電子郵件會傳送至您指定的電子郵件地址。
6. 在出現的快顯視窗中，輸入您的驗證號碼，然後選擇驗證。
7. 在 AWS DeepRacer 學生登入頁面上，輸入您的電子郵件地址和密碼，然後選擇登入。
8. 在歡迎使用 AWS DeepRacer 學生快顯視窗上，選擇完成註冊。

Note

您可以選擇我稍後會執行此操作。如果您稍後想要註冊，請立即登出。

步驟 2：完成註冊 AWS DeepRacer 學生

完成 區段以建立您的 AWS DeepRacer 學生帳戶。除非另有說明，否則所有欄位都是必要欄位。

1. 填寫 新增個人資訊以建立 AWS DeepRacer 學生帳戶區段中的欄位以建立您的帳戶。
2. 選取核取方塊，以證明您是高中或大學或社區學院註冊的學生。

Note

如果您是高中學生，請執行下列動作：

- 針對學校，選擇其他。然後，將高中的名稱新增至 欄位 輸入您的學校名稱。
- 對於目前或潛在主要，請選擇未決定，或從清單中選擇可能的潛在主要。

步驟 3：（選用）閱讀並選擇加入 AWS AI & ML 學術考量

閱讀 AWS AI & ML 學術計劃的相關資訊，以及其目標的相關資訊，請參閱 AWS AI & ML 學術計劃嗎？一節。

1. 如果您符合 AWS AI & ML Scholarship 的條件，您可以選擇加入考慮。選擇核取方塊，以確認您想要考慮獲得學費。

Note

如果您不想考慮獲得學費或未決定，請保持未選取核取方塊，然後選擇我稍後會這樣做。現在登出。現在登出，或提交以繼續而不選擇加入。您也可以更新設定檔時選擇加入。

2. （選用）使用下拉式清單在選擇性別和選擇競賽（僅限美國參與者）欄位中輸入您的資訊。
3. 選擇提交。

步驟 4：更新您的設定檔

若要更新您的設定檔，請使用您的設定檔頁面。您也可以選擇加入 AWS AI & ML Scholarship 計畫。如果您在 AWS DeepRacer 學生聯盟中收到效能的成就，您可以從您的設定檔頁面在社交媒體上共用這些成就。

更新您的設定檔


1. In AWS DeepRacer 學生，在左側導覽窗格中，選擇您的設定檔。
2. 在設定檔頁面的競賽者名稱中，選擇變更您的競賽者名稱。

Note

您的競賽者名稱可以有 2 到 24 個字元。允許使用字母、數字和連字號 (-)。

3. 在競賽者名稱模式中，輸入您的競賽者名稱，然後選擇儲存。如果您決定不想變更您的競賽者名稱，請選擇取消。
4. 在設定檔資訊區段中，您可以變更下列欄位：
 - 名稱
 - 競賽者名稱

- 學校名稱
- 主要名稱
- 畢業年份
- 居住國家/地區

 Note

需要選取居住國家/地區，才能進入 AWS DeepRacer 學生聯盟虛擬競賽並獲得獎勵。將模型提交至競賽後，您的居住國家/地區會在競賽季節鎖定在 中。

5. (選用) 在是否要考慮加入 AWS AI 和 ML 學術計劃？區段中，您可以檢視有關學術以及如何申請的資訊。
 - a. 選取核取方塊以選擇加入，以便在學術計畫中納入考量。
 - b. (選用) 在選擇性別和選擇競賽欄位中輸入您的資訊。
 - c. 選擇提交。

步驟 5：從首頁 Explore AWS DeepRacer 學生

AWS DeepRacer 學生首頁是開始探索所有 AWS DeepRacer 學生提供項目的理想位置。在 首頁中，您可以執行下列動作：

開始學習機器學習 (ML) 的基礎知識

您可以使用 AWS DeepRacer 學生學習區段中可用的免費課程。此強大產品可協助您透過 AWS DeepRacer 學生為機器學習旅程打下基礎。

練習使用您的機器學習知識

使用 Learn 課程花一些時間後，您就可以建立和訓練 AWS DeepRacer 模型。如需詳細資訊，請參閱在 [AWS DeepRacer 學生中訓練強化學習模型](#)。

參加 in AWS DeepRacer 學生競賽

完成第一個 AWS DeepRacer 模型的訓練後，您就可以參加競賽。如果您在學生聯盟每月競賽中競賽，請前往開放競賽區段，然後選擇參加競賽。當您完成競賽時，請檢查模型在排行榜上的表現，以及進行變更和改善模型所需的資料。當您參加每月虛擬競賽時，可以獲得點數。這些點決定了您的

國家和區域賽季排名。每月前 3 名的參賽者會獲得獎章和獎項。如需詳細資訊，請參閱[參加 a AWS DeepRacer 學生競賽](#)。

檢查模型訓練時數

當您訓練和複製模型時，您會使用一部分的免費模型訓練時間。您可以查看首頁上剩餘的模型訓練時數。

檢查 AWS DeepRacer 學生競賽行事曆

檢視競賽行事曆並開始規劃您的競賽日。

了解 AWS AI 和 ML 學術

您可以了解 AWS AI & ML Scholarship 以及您可以做哪些準備。如需詳細資訊，請參閱[什麼是 AWS AI & ML Scholarship 計畫？](#)

檢視其他資源

您可以探索可協助您探索 AWS DeepRacer 學生的其他資源，例如 Discord 頻道和 AWS DeepRacer 網站。這些資源可協助您連線到競賽者和粉絲的社群，分享秘訣和洞見。

在 AWS DeepRacer 學生中訓練強化學習模型

本演練示範如何在 AWS DeepRacer 學生中訓練您的第一個模型。它還提供一些有用的提示，協助您充分利用您的體驗並快速追蹤您的學習。

步驟 1：使用 AWS DeepRacer 學生訓練強化學習模型

透過了解在何處尋找建立模型按鈕並開始訓練您的第一個模型，在 AWS DeepRacer 學生中開始您的旅程。請記住，建立和訓練模型是一個反覆程序。使用不同的演算法和獎勵函數進行實驗，以獲得最佳結果。

訓練強化學習模型

1. 在 AWS DeepRacer 學生首頁中，選擇建立模型。或者，在左側導覽窗格中導覽至您的模型。在模型頁面的模型中，選擇建立模型。
2. 在概觀頁面中，閱讀如何訓練強化模型。此頁面會說明程序中的每個步驟。閱讀完成後，請選擇下一步。

步驟 2：為您的模型命名

為您的模型命名。當您想要改善和複製模型時，提供模型唯一名稱以快速找到個別模型是很好的做法。例如，您可能想要使用命名慣例來命名模型，例如：*yourinitials-date-version*。

為您的模型命名

1. 在為您的模型命名頁面上，在模型名稱欄位中輸入名稱。

Note

當您開始訓練模型時，模型的名稱會變成固定且無法再變更。

2. 選擇下一步。

步驟 3：選擇您的賽道

選擇您的模擬軌跡。賽道做為環境，並提供資料給您的汽車。如果您選擇非常複雜的賽道，您的車輛需要較長的總訓練時間，而且您使用的獎勵函數更複雜。

選擇您的賽道（環境）

1. 在選擇賽道頁面上，選擇要做為您車輛訓練環境的賽道。
2. 選擇下一步。

步驟 4：選擇演算法

AWS DeepRacer 學生有兩個訓練演算法可供選擇。不同的演算法會以不同的方式最大化獎勵。若要充分利用您的 AWS DeepRacer 學生體驗，請試用這兩種演算法。如需演算法的詳細資訊，請參閱 [AWS DeepRacer 訓練演算法](#)。

選擇訓練演算法

1. 在選擇演算法類型頁面上，選取演算法類型。有兩種演算法類型可用：
 - 近端政策最佳化 (PPO)。此穩定但資料忙碌的演算法會在訓練反覆運算之間持續執行。
 - Soft Actor Critic (SAC)。這種不穩定但資料效率高的演算法可能會在訓練反覆運算之間不一致地執行。
2. 選擇下一步。

步驟 5：自訂獎勵函數

獎勵函數是強化學習的核心。使用它來激勵您的汽車（代理程式）在探索賽道（環境）時採取特定動作。就像您鼓勵和不鼓勵寵物中的某些行為一樣，您可以使用此工具來鼓勵您的汽車盡快完成單圈，並阻止它離開賽道和鋸齒狀。

訓練第一個模型時，您可能想要使用預設範例獎勵函數。當您準備好實驗和最佳化模型時，您可以透過在程式碼編輯器中編輯程式碼來自訂獎勵函數。如需自訂獎勵函數的詳細資訊，請參閱 [自訂獎勵函數](#)。

自訂獎勵函數

1. 在自訂獎勵函數頁面上，選擇範例獎勵函數。您可以自訂 3 個範例獎勵函數：
 - 遵循中心線。當汽車自動行駛到盡可能接近賽道的中心線時，獎勵汽車。
 - 保持在邊界內。當汽車自動駕駛時獎勵汽車，四個車輪都停留在賽道邊界內。
 - 防止 zig-zag。獎勵您的汽車保持在中心線附近。如果車輛使用高轉向角度或偏離賽道，則會懲罰您的車輛。

Note

如果您不想自訂獎勵函數，請選擇下一步。

2. (選用) 修改獎勵函數程式碼。

- 選取範例獎勵函數，然後選擇逐步解說此程式碼。
- 對於程式碼的每個區段，您可以選取 + 以顯示具有說明文字的快顯文字方塊，以檢視更多資訊。在每個快顯視窗中選擇下一步，進行程式碼演練。若要退出快顯文字方塊，請選擇角落的 X。若要結束演練，請選擇完成。

Note

您可以選擇不編輯範例獎勵函數程式碼，方法是選取 Go with default code。

- 或者，選擇範例獎勵函數並選擇編輯範例程式碼，以編輯範例獎勵函數程式碼。編輯程式碼，然後選取驗證以檢查您的程式碼。如果您的程式碼無法驗證，或您想要將程式碼重設為其原始狀態，請選擇重設。

3. 選擇下一步。

步驟 6：選擇持續時間，並將模型提交至排行榜

模型訓練的持續時間會影響其效能。在訓練的早期階段進行實驗時，您應該從此參數的小數值開始，然後逐步訓練較長的時間。

在此訓練模型的步驟中，您的訓練模型會提交至排行榜。您可以透過取消選取核取方塊來選擇退出。

選擇持續時間並將模型提交至排行榜

1. 在選擇持續時間頁面上，選擇模型訓練持續時間中選擇一個時間。
2. 在模型描述欄位中，輸入模型的實用描述，協助您記住所做的選擇。

Tip

最佳實務是新增模型的相關資訊，例如獎勵函數和演算法的目前選擇和修改，以及您對模型將如何執行的假設。

3. 選取核取方塊，讓您的模型在訓練完成後自動提交至 AWS DeepRacer 學生排行榜。或者，您可以透過取消選取核取方塊來選擇不輸入模型。

Tip

建議您將模型提交至排行榜。提交模型可協助您了解模型與其他模型的比較情況，並為您提供意見回饋，以便改善模型。

4. 選擇訓練您的模型。
5. 在初始化模型訓練快顯視窗中，選擇確定。
6. 在訓練組態頁面上，您可以檢閱模型的訓練狀態和組態。當訓練狀態為進行中時，您也可以選取的賽道上檢視模型訓練的影片。觀看影片可協助您開發可用來改善模型的寶貴洞見。

步驟 7：在排行榜上檢視模型的效能

訓練模型並將其提交至排行榜後，您可以檢視其效能。

檢視模型的效能

1. 在左側導覽窗格中，導覽至 **並展開競爭**。選擇季節。在排行榜頁面上，您的模型和排名會出現在區段中。此頁面也包含排行榜區段，其中包含提交的模型清單、競賽詳細資訊和競賽詳細資訊區段。
2. 在顯示排行榜的頁面中，在您的設定檔的 **區段** 中，選取觀看影片以檢視模型效能的影片。

步驟 8：使用複製來改善模型

完成訓練並選擇性地將模型提交至排行榜後，您可以複製模型以改善模型。複製模型可節省您的步驟，並使用先前訓練過的模型作為新模型的起點，讓訓練更有效率。

複製和提升模型

1. In AWS DeepRacer 學生，在左側導覽窗格中，導覽至您的模型。
2. 在模型頁面上，選取模型，然後選擇複製。
3. 在為您的模型命名欄位中，為複製的模型提供新名稱，然後選擇下一步。
4. 在自訂獎勵函數頁面上，自訂獎勵函數，然後選擇下一步。如需自訂獎勵函數的詳細資訊，請參閱 [步驟 5：自訂獎勵函數](#)。

5. 在選擇持續時間頁面，在選擇模型訓練持續時間欄位中輸入時間，在模型描述欄位中輸入描述，然後選取核取方塊，將複製的模型提交至排行榜。
6. 選擇訓練您的模型。您的訓練已初始化。隨即出現訓練組態頁面，其中包含複製模型的相關資訊。當訓練狀態為進行中時，您也可以選取的賽道上檢視模型訓練的影片。
7. 繼續複製和修改預先訓練的模型，以在排行榜上達到最佳效能。

步驟 9：（選用）下載模型

訓練模型並選擇性地提交至排行榜後，您可能想要下載模型以供日後在 a AWS DeepRacer 實體裝置上使用。您的模型會儲存為 `.tar.gz` 檔案。

下載模型

1. In AWS DeepRacer 學生，在左側導覽窗格中，導覽至您的模型。
2. 在模型頁面上，選取模型，然後選擇下載。
3. 在瀏覽器中追蹤模型下載的進度。下載模型時，您可以將其儲存至本機硬碟或其他偏好的儲存裝置。

若要進一步了解如何使用 AWS DeepRacer 裝置，請參閱 [AWS DeepRacer 指南中的操作 YourDeepRacer 車輛](#)。AWS DeepRacer

參加 a AWS DeepRacer 學生競賽

在模擬中成功訓練和評估模型後，透過參加競賽來比較模型的效能與其他賽車手的模型。賽車是取得模型意見回饋、獲得獎勵和獎勵、虛擬認識其他 AWS DeepRacer 學生社群成員、了解學習和改善技能的機會，以及享受樂趣的絕佳方式。學生競賽有兩種類型：學生聯盟和社群。

學生聯盟競賽是所有學生都可以參加的每月虛擬競賽。學生社群競賽是由教育工作者或事件組織者在 AWS 主控台中建立的私有競賽，學生只能透過邀請參加。

本節會討論如何參加 AWS DeepRacer 學生聯盟競賽和學生社群競賽。

參加學生聯盟競賽

在本節中，了解如何將模型提交至 AWS DeepRacer 學生聯盟競賽。您可以將訓練過的模型直接提交至學生排行榜，以每月參加一次競賽。如需訓練模型的詳細資訊，請參閱 [在 AWS DeepRacer 學生中訓練強化學習模型](#)。

參加學生聯盟競賽

1. 在左側導覽窗格中的 AWS DeepRacer 學生，展開競爭並選擇學生聯盟。
2. 頁面會顯示競賽詳細資訊和排行榜。在開啟競賽區段中，選擇輸入競賽以提交模型，或者如果您尚未建立模型，請選擇建立模型。如需詳細資訊，請參閱 [在 AWS DeepRacer 學生中訓練強化學習模型](#)。
3. 在選擇要競爭的模型區段中，使用下拉式清單在選擇模型欄位中選擇模型。
4. 選擇加入競賽以提交您的模型。
5. 成功根據競賽條件評估模型後，請查看排行榜，以了解模型與其他參與者的排名。
6. 或者，選擇觀看以檢視汽車效能的影片。
7. 再次選擇競賽以輸入另一個模型。每個 AWS DeepRacer 學生聯盟競賽的提交配額為 50。

參加學生社群競賽

在本節中，了解如何將您的模型提交至 a AWS DeepRacer 學生社群競賽。您可以透過電子郵件接收來自您的教育工作者或活動主辦者的邀請連結，來參加學生社群競賽。

參加學生社群競賽

1. 前往邀請連結並登入您的 AWS DeepRacer 學生帳戶。

2. 登入後，請選擇輸入競賽按鈕。
3. 在選擇要競賽的模型下拉式清單中，選取要在社群競賽中使用的模型。
4. 選擇進入競賽以提交您的模型。
5. 如果您的模型成功根據競賽條件進行評估，請查看排行榜，查看您的模型與其他參與者的模型的排名。
6. 或者，選擇觀看以檢視汽車效能的影片。
7. 再次選擇競賽以輸入另一個模型。

自訂獎勵函數

建立獎勵函數就像設計獎勵計畫一樣。參數是可用來開發獎勵計畫的值。

不同的獎勵策略會產生不同的車輛行為。為了鼓勵車輛更快駕駛，請嘗試在車輛花太長時間才能完成單圈或離開賽道時授予負值。為了避免鋸齒狀駕駛模式，請嘗試定義轉向角度範圍限制，並獎勵車輛在賽道的直線區段上較不積極地轉向。

您可以使用路徑點，其是沿著軌道中心線和外部邊緣和內部邊緣放置的編號標記，以協助您將特定駕駛行為與軌道的特定功能建立關聯，例如直線和曲線。

製作有效的獎勵函數是一個創意且反覆的程序。嘗試不同的策略、混合和比對參數，最重要的是，享受樂趣！

主題

- [編輯 Python 程式碼以自訂獎勵函數](#)
- [AWS DeepRacer 獎勵函數的輸入參數](#)

編輯 Python 程式碼以自訂獎勵函數

In AWS DeepRacer 學生，您可以編輯範例獎勵函數，為您的模型制定自訂競賽策略。

自訂獎勵函數

1. 在 AWS DeepRacer 學生建立模型體驗的步驟 5：自訂獎勵函數頁面上，選取範例獎勵函數。
2. 使用範例獎勵函數選擇器下方的程式碼編輯器，使用 Python 程式碼自訂獎勵函數的輸入參數。
3. 選取驗證以檢查程式碼是否可運作。或者，選擇重設以重新開始。
4. 完成變更後，請選取下一步。

使用 [AWS DeepRacer 獎勵函數的輸入參數](#) 了解每個參數。了解獎勵函數範例中如何使用不同的參數。

AWS DeepRacer 獎勵函數的輸入參數

AWS DeepRacer 獎勵函數接受以變數 傳遞的字典物件params做為輸入。

```
def reward_function(params) :

    reward = ...

    return float(reward)
```

params 字典物件包含下列鍵/值對：

```
{
    "all_wheels_on_track": Boolean,          # flag to indicate if the agent is on the
    track
    "x": float,                             # agent's x-coordinate in meters
    "y": float,                             # agent's y-coordinate in meters
    "closest_objects": [int, int],          # zero-based indices of the two closest
    objects to the agent's current position of (x, y).
    "closest_waypoints": [int, int],        # indices of the two nearest waypoints.
    "distance_from_center": float,          # distance in meters from the track center
    "is_crashed": Boolean,                  # Boolean flag to indicate whether the agent
    has crashed.
    "is_left_of_center": Boolean,           # Flag to indicate if the agent is on the
    left side to the track center or not.
    "is_offtrack": Boolean,                 # Boolean flag to indicate whether the agent
    has gone off track.
    "is_reversed": Boolean,                 # flag to indicate if the agent is driving
    clockwise (True) or counter clockwise (False).
    "heading": float,                       # agent's yaw in degrees
    "objects_distance": [float, ],          # list of the objects' distances in meters
    between 0 and track_length in relation to the starting line.
    "objects_heading": [float, ],           # list of the objects' headings in degrees
    between -180 and 180.
    "objects_left_of_center": [Boolean, ],  # list of Boolean flags indicating whether
    elements' objects are left of the center (True) or not (False).
    "objects_location": [(float, float),], # list of object locations [(x,y), ...].
    "objects_speed": [float, ],             # list of the objects' speeds in meters per
    second.
    "progress": float,                       # percentage of track completed
    "speed": float,                           # agent's speed in meters per second (m/s)
    "steering_angle": float,                  # agent's steering angle in degrees
    "steps": int,                             # number steps completed
    "track_length": float,                    # track length in meters.
    "track_width": float,                     # width of the track
    "waypoints": [(float, float), ]          # list of (x,y) as milestones along the
    track center
```

```
}
```

使用下列參考，進一步了解 AWS DeepRacer 輸入參數。

all_wheels_on_track

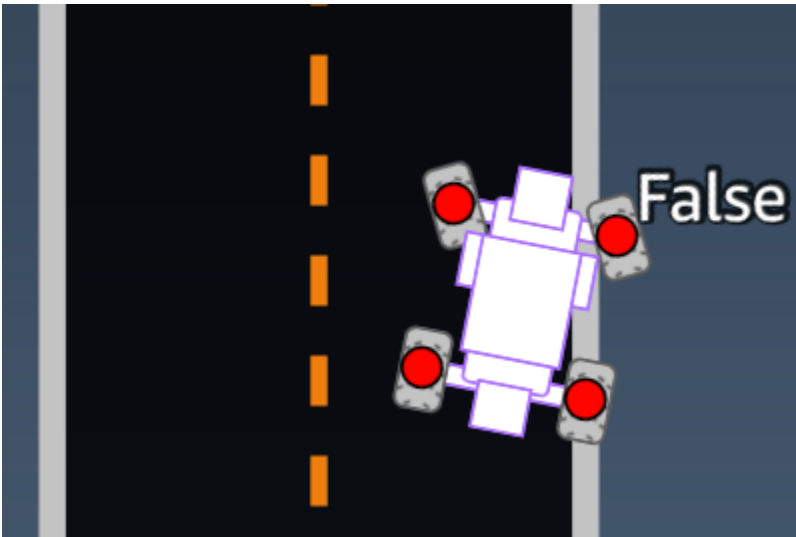
Type (類型) : Boolean

Range (範圍) : (True:False)

指出代理程式是否在正軌上的 Boolean 旗標。如果代理程式的任何一個輪子位於軌道邊界之外，則代理程式不在軌道上 (False)。如果這四個輪子都在內部和外部賽道邊界內，則表示它在賽道上 (True)。下圖顯示正在追蹤的代理程式。



下圖顯示 代理程式，因為兩個輪子位於軌道邊界之外，所以不在軌道上。



範例：使用 `all_wheels_on_track` 參數的獎勵函數

```
def reward_function(params):
    #####
    ...
    Example of using all_wheels_on_track and speed
    ...

    # Read input variables
    all_wheels_on_track = params['all_wheels_on_track']
    speed = params['speed']

    # Set the speed threshold based your action space
    SPEED_THRESHOLD = 1.0

    if not all_wheels_on_track:
        # Penalize if the car goes off track
        reward = 1e-3
    elif speed < SPEED_THRESHOLD:
        # Penalize if the car goes too slow
        reward = 0.5
    else:
        # High reward if the car stays on track and goes fast
        reward = 1.0

    return float(reward)
```

closest_waypoints

Type (類型) : [int, int]

Range (範圍) : [(0:Max-1), (1:Max-1)]

兩個與代理程式目前位置 ((x, y)) 最接近相鄰 waypoint 以零為基礎的索引。距離會以從代理程式中心計算的歐幾里得距離測量。第一個元素是代理程式後方最接近的導航點，第二個元素則是代理程式前方最接近的導航點。Max 則是導航點清單的長度。在 [中顯示的圖例中 waypoints](#)，closest_waypoints 是 [16, 17]。

以下範例獎勵函數會示範如何使用 waypoints 和 closest_waypoints，以及 heading 來計算立即性獎勵。

AWS DeepRacer 支援下列 Python 程式庫：math、scipy、random numpy 和 shapely。若要使用，請在函數定義 import *supported library* 之前新增匯入陳述式 def reward_function(params)。

範例：使用 *closest_waypoints* 參數的獎勵函數。

```
# Place import statement outside of function (supported libraries: math, random, numpy,
    scipy, and shapely)
# Example imports of available libraries
#
# import math
# import random
# import numpy
# import scipy
# import shapely

import math

def reward_function(params):
    #####
    ...
    Example of using waypoints and heading to make the car point in the right direction
    ...

    # Read input variables
    waypoints = params['waypoints']
    closest_waypoints = params['closest_waypoints']
    heading = params['heading']
```

```
# Initialize the reward with typical value
reward = 1.0

# Calculate the direction of the centerline based on the closest waypoints
next_point = waypoints[closest_waypoints[1]]
prev_point = waypoints[closest_waypoints[0]]

# Calculate the direction in radius, arctan2(dy, dx), the result is (-pi, pi) in
radians
track_direction = math.atan2(next_point[1] - prev_point[1], next_point[0] -
prev_point[0])
# Convert to degree
track_direction = math.degrees(track_direction)

# Calculate the difference between the track direction and the heading direction of
the car
direction_diff = abs(track_direction - heading)
if direction_diff > 180:
    direction_diff = 360 - direction_diff

# Penalize the reward if the difference is too large
DIRECTION_THRESHOLD = 10.0
if direction_diff > DIRECTION_THRESHOLD:
    reward *= 0.5

return float(reward)
```

closest_objects

Type (類型) : [int, int]

Range (範圍) : [(0:len(object_locations)-1), (0:len(object_locations)-1)]

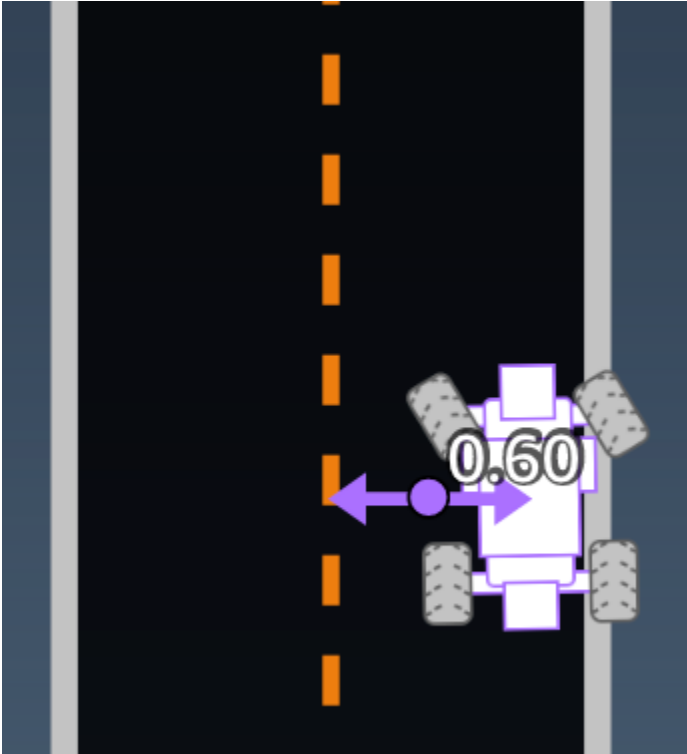
兩個與代理程式目前位置 (x, y) 最接近物件的以零為基礎索引。第一個索引是代理程式後方最接近的物件，第二個索引則是代理程式前方最接近的物件。如果只有一個物件，則兩個索引都將會是 0。

distance_from_center

Type (類型) : float

Range (範圍) : $0:\sim\text{track_width}/2$

代理程式中心與賽道中心之間的置換，以公尺為單位。可觀察到的最大偏移會在代理程式的任何一個輪胎位於軌道邊界外圍時發生，取決於軌道邊界的寬度，這可能會略小或略大於 `track_width` 的一半。



範例：使用 `distance_from_center` 參數的獎勵函數

```
def reward_function(params):
    #####
    ...
    Example of using distance from the center
    ...

    # Read input variable
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # Penalize if the car is too far away from the center
    marker_1 = 0.1 * track_width
    marker_2 = 0.5 * track_width

    if distance_from_center <= marker_1:
        reward = 1.0
```

```
elif distance_from_center <= marker_2:  
    reward = 0.5  
else:  
    reward = 1e-3 # likely crashed/ close to off track  
  
return float(reward)
```

heading

Type (類型) : float

Range (範圍) : -180:+180

代理程式相對於座標系統 x 軸的方向，以度為單位。



範例：使用 *heading* 參數的獎勵函數

如需詳細資訊，請參閱[closest_waypoints](#)。

is_crashed

Type (類型) : Boolean

Range (範圍) : (True:False)

指出代理程式是否已當機進入另一個物件 (True) 或未當成終止狀態 (False) 的 Boolean 旗標。

is_left_of_center

Type (類型) : Boolean

Range (範圍) : [True : False]

指出代理程式是否位於軌道中心 (True) 或非軌道中心 () 的左側的 Boolean 旗標 False。

is_offtrack

Type (類型) : Boolean

Range (範圍) : (True:False)

指出所有四個代理器輪子是否在軌道內部或外部電路板 (True) 外駕駛的 Boolean 標記 () False。

is_reversed

Type (類型) : Boolean

Range (範圍) : [True:False]

指出代理程式是順時針駕駛 (True) 還是逆時針駕駛 () 的 Boolean 旗標 False。

此標記會在您為每個回合啟用方向變更時使用。

objects_distance

Type (類型) : [float, ...]

Range (範圍) : [(0:track_length), ...]

環境中物件與起始線相關的距離清單。第 i 個元素會測量沿著賽道中心線，第 i 個物件與起始線之間的距離 (公尺)。

Note

$\text{abs} | (\text{var1}) - (\text{var2}) |$ = how close the car is to an object, WHEN $\text{var1} = [\text{"objects_distance"}]$
 $[\text{index}]$ and $\text{var2} = \text{params}[\text{"progress"}] * \text{params}[\text{"track_length"}]$

若要取得車輛前方最接近物件和車輛後方最接近物件的索引，請使用 `closest_objects` 參數。

objects_heading

Type (類型) : [float, ...]

Range (範圍) : [(-180:180), ...]

物件所面對方向的清單 (角度)。第 i 個元素會測量第 i 個物件所面對的方向。靜態物件的標題為 0。對於機器人車，對應的元素值是機器人車的前進角度。

objects_left_of_center

Type (類型) : [Boolean, ...]

Range (範圍) : [True|False, ...]

Boolean 旗標清單。第 i 個元素值指出第 i 個物件位於軌跡中心的左側 (True) 或右側 (False)。

objects_location

Type (類型) : [(x,y), ...]

Range (範圍) : [(0:N,0:N), ...]

此參數存放所有物件位置。每個位置都是 [\(x, y\)](#) 的元組。

清單的大小與賽道上的物件數量相等。列出的物件包括固定障礙物和移動的機器人車。

objects_speed

Type (類型) : [float, ...]

Range (範圍) : [(0:12.0), ...]

賽道上物件的速度清單 (單位為每秒的公尺數)。針對固定物件，其速度為 0。對於機器人車輛，值是您在訓練中設定的速度。

進度

Type (類型) : float

Range (範圍) : 0:100

完成軌道的百分比。

範例：使用 *progress* 參數的獎勵函數

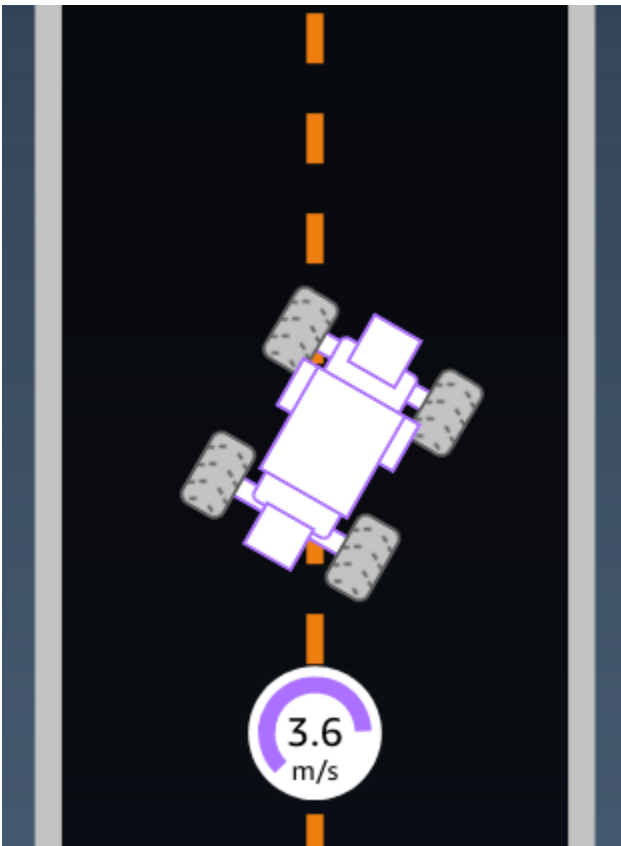
如需詳細資訊，請參閱[梯度](#)。

speed

Type (類型) : float

Range (範圍) : 0.0:5.0

代理程式所觀察到的速度，單位為公尺/每秒 (m/s)。



範例：使用 *speed* 參數的獎勵函數

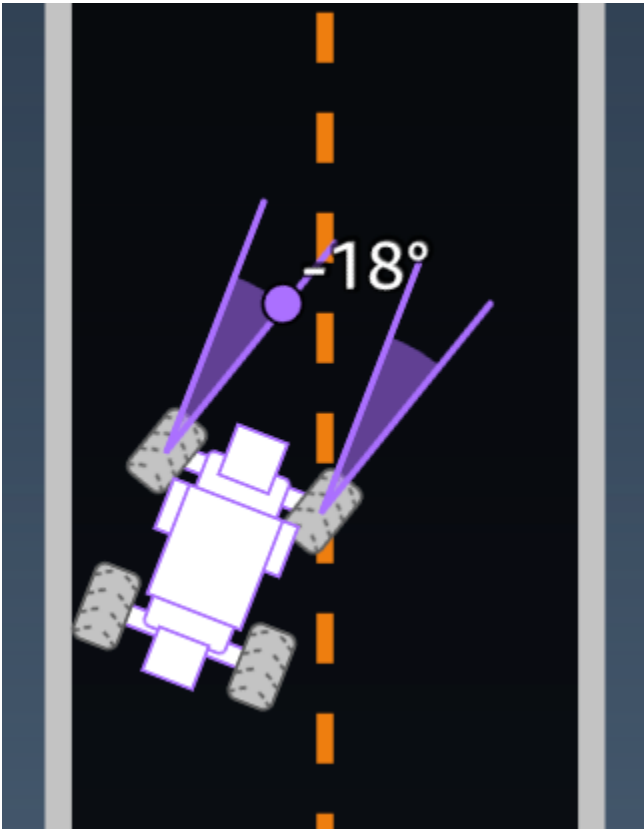
如需詳細資訊，請參閱 [all_wheels_on_track](#)。

steering_angle

Type (類型) : float

Range (範圍) : -30:30

前輪偏移代理程式中線的方向盤角度 (度)。負號 (-) 表示向右偏移，正號 (+) 表示向左偏移。代理程式的中心線不一定與軌道中心線平行，如下圖所示。



範例：使用 *steering_angle* 參數的獎勵函數

```
def reward_function(params):  
    ...  
    Example of using steering angle  
    ...  
  
    # Read input variable  
    abs_steering = abs(params['steering_angle']) # We don't care whether it is left or  
    right steering
```

```
# Initialize the reward with typical value
reward = 1.0

# Penalize if car steer too much to prevent zigzag
ABS_STEERING_THRESHOLD = 20.0
if abs_steering > ABS_STEERING_THRESHOLD:
    reward *= 0.8

return float(reward)
```

steps

Type (類型) : int

Range (範圍) : 0:N_{step}

完成的步驟數目。步驟對應至客服人員使用目前政策完成的一個觀察動作序列。

範例：使用 *steps* 參數的獎勵函數

```
def reward_function(params):
    #####
    ...
    Example of using steps and progress
    ...

    # Read input variable
    steps = params['steps']
    progress = params['progress']

    # Total num of steps we want the car to finish the lap, it will vary depends on the
    track length
    TOTAL_NUM_STEPS = 300

    # Initialize the reward with typical value
    reward = 1.0

    # Give additional reward if the car pass every 100 steps faster than expected
    if (steps % 100) == 0 and progress > (steps / TOTAL_NUM_STEPS) * 100 :
        reward += 10.0

    return float(reward)
```

track_length

Type (類型) : float

Range (範圍) : $[0:L_{\max}]$

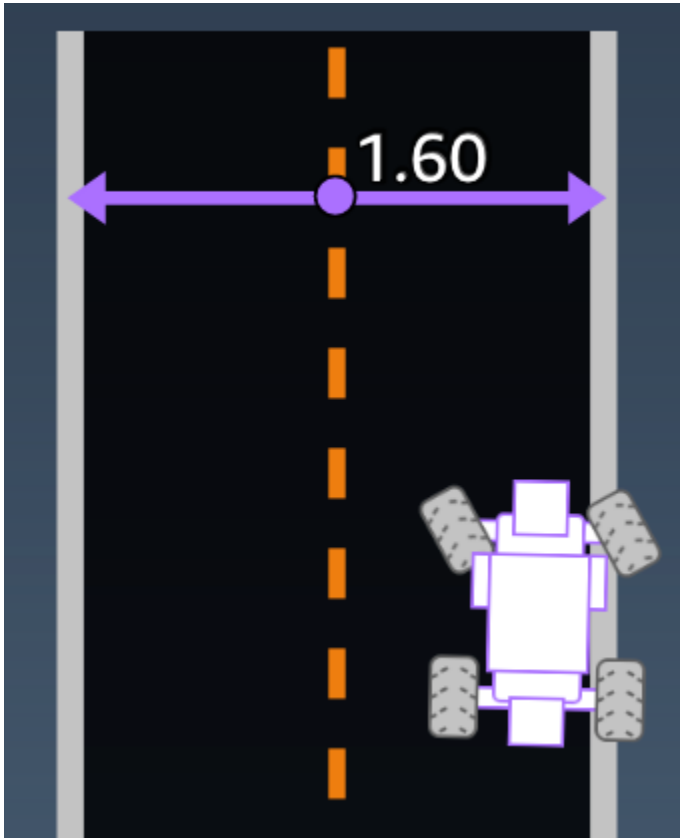
以公尺為單位的賽道長度。 L_{\max} is track-dependent.

track_width

Type (類型) : float

Range (範圍) : $0:D_{\text{track}}$

軌道寬度 (公尺)。



範例：使用 *track_width* 參數的獎勵函數

```
def reward_function(params):  
    #####  
    ...
```

```
Example of using track width
'''

# Read input variable
track_width = params['track_width']
distance_from_center = params['distance_from_center']

# Calculate the distance from each border
distance_from_border = 0.5 * track_width - distance_from_center

# Reward higher if the car stays inside the track borders
if distance_from_border >= 0.05:
    reward = 1.0
else:
    reward = 1e-3 # Low reward if too close to the border or goes off the track

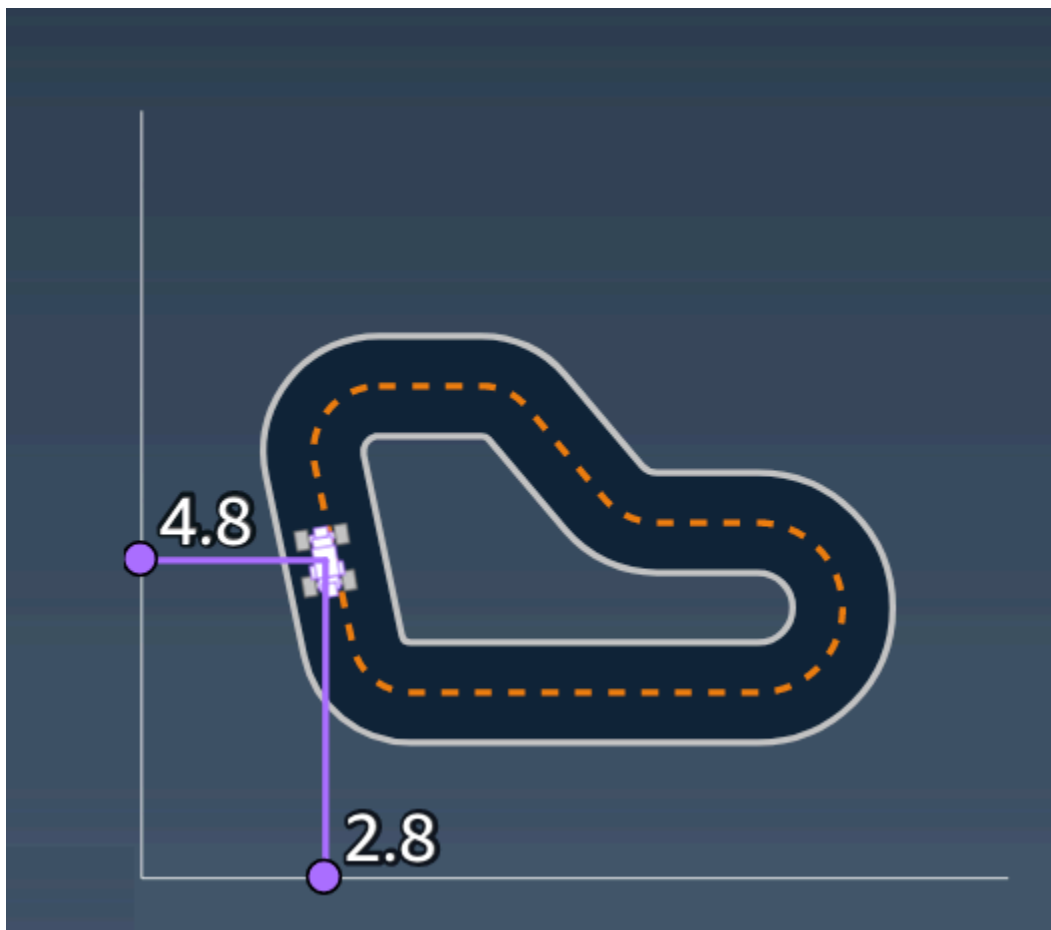
return float(reward)
```

X, y

Type (類型) : float

Range (範圍) : 0:N

代理程式中心沿著包含軌道之模擬環境的 x 軸和 y 軸的位置，以公尺為單位。原點位於模擬環境的左下角。



waypoints

Type (類型) : [float, float] 的 list

Range (範圍) : $[[x_{w,0}, y_{w,0}] \dots [x_{w,Max-1}, y_{w,Max-1}]]$

沿著賽道中心的相依賽道 Max 里程碑排序清單。每個里程碑都由 $(x_{w,i}, y)$ 的座標描述 $w_{,i}$ 。對於環狀軌道，第一個和最後一個路徑點是相同的。對於直線軌道或其他非環狀軌道，第一個和最後一個路徑點則不同。



範例 使用 *waypoints* 參數的獎勵函數

如需詳細資訊，請參閱[closest_waypoints](#)。

Security in AWS DeepRacer 學生

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，該架構專為符合最安全敏感組織的需求而建置。

安全性是 AWS 與您之間共同責任。[共同責任模型](#)將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。作為[AWS 合規計劃](#)的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 AWS DeepRacer 學生的合規計劃，請參閱[合規計劃的 AWS 服務範圍](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規

本文件可協助您了解如何在使用 AWS DeepRacer 學生時套用共同責任模型。它說明如何設定 AWS DeepRacer 學生以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS DeepRacer 學生資源。

目錄

- [資料保護 in AWS DeepRacer 學生](#)
- [適用於 AWS DeepRacer 學生的身分和存取管理](#)
- [AWS DeepRacer 學生的合規驗證](#)
- [彈性 in AWS DeepRacer 學生](#)
- [基礎設施安全 in AWS DeepRacer 學生](#)

資料保護 in AWS DeepRacer 學生

以下各節說明 AWS DeepRacer 學生擷取哪些資料，以及 AWS DeepRacer 學生在何處使用資料加密來保護您的資料。

當您建立 a AWS DeepRacer 學生帳戶時，您也會建立 AWS 玩家帳戶。在您的 AWS DeepRacer 學生帳戶中建立的資源會存放在您的 AWS 玩家帳戶中。如需 AWS 玩家帳戶的詳細資訊，請參閱 AWS DeepRacer 學生使用者指南[什麼是 AWS 玩家帳戶？](#)中的。

主題

- [在 AWS DeepRacer 學生入口網站中擷取的資料](#)

- [靜態加密 in AWS DeepRacer 學生入口網站](#)
- [傳輸中加密 in AWS DeepRacer 學生入口網站](#)

在 AWS DeepRacer 學生入口網站中擷取的資料

若要使用 AWS DeepRacer 學生入口網站，所需的資料會存放在您的 AWS 玩家帳戶中。在 AWS DeepRacer 學生入口網站中擷取的資料不會用於協助改善服務。

擷取資料 in AWS DeepRacer 學生。

以下是在 AWS DeepRacer Student 中建立並存放在 AWS 玩家帳戶中的資料摘要。

- 用來註冊帳戶的電子郵件地址和密碼。
- 您的競賽者名稱
- 您在學生聯盟排行榜上的站立
- 您訓練過的模型
- 獎勵函數程式碼

靜態加密 in AWS DeepRacer 學生入口網站

根據預設，AWS DeepRacer 學生入口網站擷取的資料會加密。

AWS 玩家帳戶使用 Amazon Cognito 來加密和存放用於登入 AWS DeepRacer 學生的電子郵件和密碼。如需詳細資訊，請參閱 [Amazon Cognito 中的資料保護](#)。

使用 AWS 擁有的金鑰透過 AWS Key Management Service AES-GCM AWS DeepRacer 並使用大小為 256 位元的金鑰，在雲端中靜態加密擷取的所有其他資料。此資料儲存在 Amazon Simple Storage Service (S3) 和 Amazon DynamoDB 中並加密。

傳輸中加密 in AWS DeepRacer 學生入口網站

您的註冊和授權電子郵件地址會使用用戶端加密進行加密。擷取到 [in AWS DeepRacer Student 的所有其他資料](#) 都會從您的帳戶複製，並在內部 AWS 系統中處理。根據預設，AWS DeepRacer 學生透過 HTTPS 使用安全連線來加密傳輸中的資料。

適用於 AWS DeepRacer 學生的身分和存取管理

AWS Identity and Access Management (IAM) 是一項 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證（登入）和授權（具有許可）來使用 AWS 資源。AWS DeepRacer Student 不會直接與 IAM 整合，以控制使用者對 AWS 資源的存取。反之，AWS DeepRacer Student 會使用由 AWS DeepRacer 管理的已驗證代理 API 來保護使用者資源。

AWS DeepRacer 學生的合規驗證

在多個合規計畫中，第三方稽核人員會評估 AWS DeepRacer Student 的安全性和 AWS 合規性。

如需特定合規計畫範圍內 AWS 的服務清單，請參閱[合規計畫的 AWS 服務合規](#)。如需一般資訊，請參閱[AWS 合規計畫](#)。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載報告下載 AWS Artifact](#)。

AWS 提供下列資源以協助合規：

- [安全與合規快速入門指南](#)：這些部署指南討論架構考量，並提供在 AWS 上部署以安全及合規為重心之基準環境的步驟。
- [AWS 合規資源](#) – 此工作手冊和指南集合可能適用於您的產業和據點。
- 《AWS Config 開發人員指南》中的[使用規則評估資源](#) AWS Config：評估資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub CSPM](#) – AWS 此服務提供內安全狀態的完整檢視 AWS，協助您檢查是否符合安全產業標準和最佳實務。

彈性 in AWS DeepRacer 學生

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱[AWS 全球基礎設施](#)。

基礎設施安全 in AWS DeepRacer 學生

作為受管服務，AWS DeepRacer 學生受到 [Amazon Web Services : 安全程序概觀](#) 白皮書中所述 AWS 的全球網路安全程序的保護。

疑難排解 common AWS DeepRacer 學生問題

主題

- [為什麼我自動登出 my AWS DeepRacer 學生帳戶？](#)
- [如何選擇退出 AWS AI & ML Scholarship 計畫？](#)
- [我無法刪除 my AWS DeepRacer 學生帳戶](#)
- [我在下拉式清單中找不到我的學校名稱](#)
- [我無法繼續訓練我的模型](#)
- [我收到「帳戶已使用此電子郵件註冊」錯誤訊息](#)
- [我註冊了 Gmail 帳戶，但找不到我的驗證碼](#)

為什麼我自動登出 my AWS DeepRacer 學生帳戶？

為了遵循 AWS 安全政策，您在 30 天後會自動登出 AWS DeepRacer 學生帳戶。

- 若要繼續使用服務，請導覽至 [AWS DeepRacer 學生登入頁面](#)，並使用您的登入資料重新登入。

如何選擇退出 AWS AI & ML Scholarship 計畫？

AWS AI & ML Scholarship 計畫是選用的，適用於 16 歲以上未受關心和代表性不足的學生。當您註冊 for AWS DeepRacer 學生時，預設不會註冊 AWS AI & ML Scholarship 計畫。

若要參與，您必須先勾選 AWS AI & ML Scholarship 計畫？區段中的方塊，以便在註冊 AWS DeepRacer 學生時或之後從您的設定檔頁面選擇加入，該頁面可從網站的左側導覽窗格存取。

- 選擇加入計畫只會讓您存取應用程式程序。您仍然可以选择不套用。

我無法刪除 my AWS DeepRacer 學生帳戶

如果您無法刪除您的 AWS DeepRacer 學生帳戶，請檢查您是否已建立 AWS DeepRacer 多使用者事件。AWS 玩家帳戶是由 AWS for AWS DeepRacer 多使用者和 AWS DeepRacer 學生建立的受管身分解決方案。您的 AWS 玩家帳戶會保留在這些服務中建立的所有資源 AWS。

- 為了確保您建立的事件參與者不會因體驗中斷而留下，如果其包含適用於 an AWS DeepRacer 多使用者事件的資源，則無法刪除您的 AWS DeepRacer 學生帳戶。

我在下拉式清單中找不到我的學校名稱

您可能無法在下拉式清單中找到所有學校，尤其是高中。

- 如果您的學校不在下拉式清單中，請選擇其他，然後輸入您的學校名稱。

我無法繼續訓練我的模型

您可能已超過每月模型訓練時數限制。

- 前往首頁，在模型訓練剩餘時數區段中檢查已使用的訓練時數。如果您已超過模型訓練時數，請等到您的時數重設後再開始訓練。

我收到「帳戶已使用此電子郵件註冊」錯誤訊息

當您在 AWS 玩家帳戶註冊頁面中輸入確認碼，且您已使用相同的電子郵件地址透過 AWS DeepRacer 多使用者註冊 AWS 玩家帳戶時，會收到此錯誤訊息。當您先前使用相同的電子郵件地址註冊 an AWS DeepRacer 學生帳戶時，也會收到此錯誤。

- 在密碼欄位下，使用您先前建立的登入資料登入 [AWS DeepRacer 學生登入頁面](#)，或選取忘記密碼？來請求密碼重設。

我註冊了 Gmail 帳戶，但找不到我的驗證碼

如果您已使用 Gmail 帳戶註冊 AWS 玩家帳戶，但找不到您的驗證碼訊息，則可能已傳送至錯誤的資料夾。

- 登入您的 Gmail 帳戶，並檢查提升資料夾是否有名為「您的 AWS 播放器設定檔驗證碼」的訊息。

AWS DeepRacer 學生配額

參與 AWS DeepRacer 學生的每個學生都會獲得每月 10 小時的免費模型訓練運算資源和 5GB 的儲存空間。

刪除您的 AWS DeepRacer 學生帳戶

AWS DeepRacer 學生入口網站會將下列資訊存放在您的 AWS 玩家帳戶中：

- 電子郵件地址
- 您的密碼
- 您的競賽者名稱
- 您的學生聯盟排行榜排名

若要進一步了解收集的資料，請參閱 AWS DeepRacer 學生使用者指南 [資料保護 in AWS DeepRacer 學生](#) 中的。

如果您想要從 AWS 的伺服器移除此資訊，請使用下列程序刪除您的 AWS DeepRacer 學生入口網站帳戶。刪除您的 AWS DeepRacer 學生帳戶也會刪除您的 AWS 玩家帳戶和所有相關聯的資源。

若要進一步了解 AWS 玩家帳戶，請參閱 [什麼是 AWS 玩家帳戶？](#)

Note

如果您已建立 AWS DeepRacer 多使用者事件，則無法刪除您的 AWS 玩家帳戶。如需詳細資訊，請參閱 AWS DeepRacer 學生使用者指南 [我無法刪除 my AWS DeepRacer 學生帳戶](#) 中的。

刪除您的 AWS DeepRacer 學生玩家帳戶

Important

刪除您的 AWS DeepRacer 學生帳戶是無法復原的動作。當您刪除 AWS DeepRacer 學生時，也會刪除您的 AWS 玩家帳戶和所有相關聯的資源。
當您刪除您的 AWS DeepRacer 學生帳戶時，您的 AWS 玩家帳戶中的資源會在一年內從我們的伺服器中移除。

1. 開啟 AWS DeepRacer 學生登陸頁面：<https://student.deepracer.com/signIn>。
2. 如果出現提示，請登入您的 AWS DeepRacer 學生帳戶。

3. 選擇您的帳戶。
4. 在您的帳戶頁面上，選擇刪除您的帳戶。
5. 在確認刪除下，在欄位中輸入 Delete，輸入 **Delete**。
6. 選擇 刪除。

成功刪除您的帳戶時，帳戶刪除的訊息會成功顯示，而且您會回到 AWS DeepRacer 學生登入頁面。

如果您也想要刪除 AWS 帳戶，請使用[關閉 AWS 帳戶](#)中所述的步驟。

我們知道客戶非常重視隱私權和資料安全，並實作負責任且複雜的技術和實體控制，旨在防止未經授權存取或揭露客戶內容。維護客戶信任是持續的承諾。您可以在我們的隱私權[通知](#)頁面上進一步了解 AWS 資料隱私權承諾。

AWS DeepRacer 學生使用者指南的文件歷史記錄

下表說明 for AWS DeepRacer 學生的文件版本。

變更	描述	日期
2024 年計劃的更新	更新參考 2024 AWS AI & ML 學術計劃和 AWS DeepRacer 學生聯盟的主題。已新增所有申請 AWS AI & ML Scholarship 學生可用的新課程相關資訊。	2023 年 11 月 1 日
2023 AWS DeepRacer 學生聯盟的更新	更新參考 2023 AWS DeepRacer 學生聯盟的多個主題。如需詳細資訊，請參閱 條款與條件 。	2023 年 3 月 1 日
初始版本	AWS DeepRacer 學生使用者指南的初始版本，包括對 AWS AI 和 ML 學術計劃和 AWS DeepRacer 學生聯盟的支援。	2021 年 12 月 1 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。