



ユーザーガイド

# AWS DeepRacer 学生



# AWS DeepRacer 学生: ユーザーガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

# Table of Contents

AWS DeepRacer Student について教えてください .....	1
AWS AI & ML 奨学金プログラムのご紹介 .....	2
AWS DeepRacer Student を初めてお使いになるユーザー向けの情報 .....	2
AWS DeepRacer 学生リーグへの参加に興味がありますか？ .....	2
AWS AI & 機械学習スカラシッププログラムとは .....	3
AWS AI & ML 奨学金で得られるもの .....	3
AWS AI & 機械学習スカラシッププログラムへのサインアップと事前資格審査の方法 .....	4
AWS AI & 機械学習スカラシッププログラムと Udacity プラットフォームに関するよくある質問 (FAQ) .....	5
AWS Player アカウントとは .....	8
サポートされているサービスの AWS Player アカウントの作成 .....	8
Player AWS アカウントの削除 .....	8
はじめに .....	10
前提条件 .....	10
ステップ 1: AWS DeepRacer にサインアップする .....	10
ステップ 2: AWS DeepRacer Student のサインアップを完了する .....	11
ステップ 3: (オプション) AWS AI & ML 奨学金の検討を参照し、オプトインする .....	12
ステップ 4: プロファイルを更新する .....	13
ステップ 5: AWS DeepRacer Student を [Home] (ホーム) ページから探索する .....	14
強化学習モデルをトレーニングする .....	16
ステップ 1: AWS DeepRacer Student を使用して強化学習モデルをトレーニングする .....	16
ステップ 2: モデルに名前を付ける .....	16
ステップ 3: トラックを選択する .....	17
ステップ 4: アルゴリズムを選択する .....	17
ステップ 5: 報酬関数をカスタマイズする .....	18
ステップ 6: 期間を選択し、モデルをリーダーボードに送信する .....	19
ステップ 7: リーダーボードでモデルのパフォーマンスを表示する .....	20
ステップ 8: [Clone] (クローン) を使用してモデルを改善する .....	20
ステップ 9: (オプション) モデルをダウンロードする .....	21
レースに参加する .....	22
学生リーグレースに参加する .....	22
学生コミュニティレースに参加する .....	23
報酬関数をカスタマイズする .....	24
Python コードを編集して報酬関数をカスタマイズする .....	24

報酬関数の入力パラメータ .....	25
all_wheels_on_track .....	26
closest_waypoints .....	28
closest_objects .....	29
distance_from_center .....	30
heading .....	31
is_crashed .....	32
is_left_of_center .....	32
is_offtrack .....	32
is_reversed .....	32
objects_distance .....	32
objects_heading .....	33
objects_left_of_center .....	33
objects_location .....	33
objects_speed .....	34
progress .....	34
速度 .....	34
steering_angle .....	35
ステップ .....	37
track_length .....	37
track_width .....	38
x、y .....	39
ウェイポイント .....	40
セキュリティ .....	41
データ保護 .....	42
AWS DeepRacer Student ポータルでキャプチャしたデータ .....	42
AWS DeepRacer Student ポータルへの保管時の暗号化 .....	42
AWS DeepRacer Student ポータルへの転送中の暗号化 .....	43
Identity and Access Management .....	43
コンプライアンス検証 .....	43
耐障害性 .....	44
基盤セキュリティ .....	44
よくある問題に対するトラブルシューティング .....	45
AWS DeepRacer Student アカウントから自動的にサインアウトされたのはなぜですか？ .....	45
AWS AI & ML 奨学金プログラムからのオプトアウト方法を教えてください。 .....	45
AWS DeepRacer Student アカウントを削除できない .....	46

ドリップダウンリストに学校名が見つかりません。 .....	46
モデルのトレーニングが継続できません。 .....	46
「An account is registered with this email (この E メールで登録済みのアカウントがあります)」というエラーメッセージが表示されます .....	47
Gmail アカウントでサインアップしましたが認証コードが見つかりません .....	47
クォータ .....	48
アカウントの削除 .....	49
ドキュメント履歴 .....	51
.....	lii

# AWS DeepRacer Student について教えてください

AWS DeepRacer Student は、世界中の高校および大学に入学した学習者が機械学習 (ML) のスキルを身につけるための場所です。このサービスでは、学習素材にアクセスしたり、オプションの AWS AI & ML 奨学金プログラムに参加したりできます。また、AWS DeepRacer Student League に参加するために、強化学習 (RL) モデルのトレーニングとテストを行う機会なども得られます。開始するには、「[???](#)」セクションのトピックを参照してください。

## AWS DeepRacer Student の機能

- ホーム – 今後行われるイベントの詳細確認、RL モデルのトレーニングの実践、ML に関する学習コンテンツへのアクセス、およびモデルトレーニング時間の追跡が行えます。左のナビゲーションペインでは、AWS DeepRacer Student のプロフィールとアカウント情報を管理することもできます。
- [学習] – AWS のエキスパートが開発した (動画を含む) ML についてのコンテンツにアクセスできます。事前に経験がない学生は、理解しやすく、自分のペースで学習できる教材を使用して、ML の基礎を学ぶことができます。
- 実践 – トラック、アルゴリズム、および報酬関数を選択して、RL モデルを作成します。オプションで、Python で記述された報酬関数のコードに関するガイド付きチュートリアルを利用したり、そのコードをカスタマイズする選択も可能です。AWS DeepRacer のサービスを通じて、シミュレーションされた 3D レース環境の中でモデルのトレーニングを行います。最高のパフォーマンスを発揮するモデルのクローンを作成し報酬関数を繰り返すことで、AWS DeepRacer 学生リーグのランキングアップを目指していきます。
- 競技 – 毎月開催される仮想化レースにモデルを送り出して、アチーブメントや賞品を獲得するチャンスを手に入れましょう。地域のトップ学生レーサーは、Student World Championships に出場する機会を得ます。詳細については、「[用語と条件](#)」を参照してください。すべての学生には、AWS DeepRacer 学生リーグ向けの RL モデルのトレーニング用として、毎月 10 時間が与えられます。AWS DeepRacer Student を使用してトレーニングしたモデルのみが、AWS DeepRacer Student League に参加できます

## トピック

- [AWS AI & ML 奨学金プログラムのご紹介](#)
- [AWS DeepRacer Student を初めてお使いになるユーザー向けの情報](#)
- [AWS DeepRacer 学生リーグへの参加に興味がありますか？](#)

# AWS AI & ML 奨学金プログラムのご紹介

テクノロジーの恩恵を十分に受けていない、あるいは、その分野では自分は少数派に属するとお考えの方は、是非 AWS AI & ML 奨学金プログラムに参加してください。プログラムの採用基準、応募方法、獲得できる成果について詳細は、[AWS AI & 機械学習スカラシッププログラムとは](#) をご確認ください。

## AWS DeepRacer Student を初めてお使いになるユーザー向けの情報

AWS DeepRacer Student を初めて使用する方には、まず以下のセクションをお読みいただくことをお勧めします。

- [AWS DeepRacer Student の開始方法](#)
- [AWS DeepRacer Student レースに参加する](#)
- [AWS DeepRacer Student での強化学習モデルのトレーニング](#)

## AWS DeepRacer 学生リーグへの参加に興味がありますか？

AWS DeepRacer 学生リーグは AWS 主催の大会で、学生は 3 月から 9 月まで毎月開催される仮想化レースに参加して国や地域レベルで競い合い、10 月の学生世界選手権に出場する資格を得ることができます。学生は報酬や賞品を獲得する機会があります。詳細については、「[用語と条件](#)」を参照してください。

初めてのレースに出場するには、「[AWS DeepRacer Student レースに参加する](#)」を参照してください。

# AWS AI & 機械学習スカラシッププログラムとは

AWS AI & 機械学習スカラシッププログラムは、AWS DeepRacer Student の一部として立ち上げることができ、人工知能 (AI) と機械学習 (ML) の分野に人的多様性をもたらすことを目的としています。採用された応募者は、最大 2 つの Udacity の Nanodegrees を目指す機会を得ることができます。Udacity による Nanodegree は、学習とキャリア形成の間にあるギャップを埋めるために考案された、大規模なオープンオンラインコース (MOOC) です。詳細については、Udacity サポートドキュメントの「[What is a Nanodegree Program?](#)」を参照してください。採用された応募者は、特別なイベントに参加したり、指導を受けたりしながら、キャリアアップを目指していきます。詳細については、「[AWS AI & ML 奨学金で得られるもの](#)」を参照してください。

この奨学金は、テクノロジーの恩恵を十分に受けていなかったり、その分野では少数派に属する人々に焦点を当てています。これに応募する人は、年齢が16歳以上で、高校、大学、コミュニティカレッジのいずれかに在籍している必要があります。事前応募資格の詳細については、「[AWS AI & 機械学習スカラシッププログラムへのサインアップと事前資格審査の方法](#)」を参照してください。

AWS AI & 機械学習スカラシッププログラムへの参加は無料です。

AWS AI & 機械学習スカラシッププログラムは、2022 年 4 月 11 日に正式に開始されます。

AWS AI & 機械学習スカラシッププログラムでは、コホートベースのアプローチを採用しています。ここでは毎年、それぞれ 1,000 人の生徒からなるコホートが 2 組 (計 2,000 人) 選出されます。第 1 コホートの申請は 2024 年 5 月 31 日までに Udacity に届く必要があり、第 2 コホートの申請期日は 2024 年 9 月 30 日までとなっています。

## AWS AI & ML 奨学金で得られるもの

AWS AI & 機械学習スカラシッププログラムは、AWS が Udacity との協業により進めるものです。

AWS AI & 機械学習スカラシッププログラムは、毎年、2 つのコホート全体で、2,500 件の Udacity Nanodegree に関する奨学金を世界中に提供していきます。学習者の方には、全員、以下が提供されます。

- AI Programming with Python の Nanodegree への無料参加。このコースでは、AI 技術の使用開始に必要な基礎スキルを学べると同時に、プログラミング、線形代数、ニューラルネットワークに関するスキルを磨くことができます。



- 平日の営業時間には、Udacity のインストラクターにアクセスすることで、講義内容に関する疑問を解決できます。また、学習者は Udacity インストラクターの指導の下で毎週行われる、導入事例に基づいた演習にも参加できます。
- 業界の専門家によるセッション、履歴書編集ガイダンス、面接のヒントなど、メンターシップやキャリアリソースへの限定アクセスを提供します。

また、AI Programming with Python クラスの (習得度テストでのパフォーマンスに基づく) 上位 500 人の学生には、深層学習の基礎を学ぶためのさらに高度な Nanodegree への参加権が与えられ、そこで機械学習エンジニアになるために必要なスキルを獲得することも可能です。

奨学金の前提条件を満たし、Udacity の申請書を提出したすべての学生は、AWS と Udacity の共同制作による新しい生成系 AI コース「Introducing Generative AI with AWS」に無料でアクセスできます。1 か月コースは、生成系 AI 分野での学生のスキル向上に役立つオプションリソースです。このコースは、奨学金の申請書を提出したすべての学生が利用できるのに対し、Nanodegree スカラシップは奨学金受給者のみが利用できます。

## AWS AI & 機械学習スカラシッププログラムへのサインアップと事前資格審査の方法

AWS AI & 機械学習スカラシッププログラムは、この技術の恩恵を十分に受けておらず、そこでの活躍の場も限られている、16 歳以上の学生を対象としたプログラムです。技術の恩恵を受けておらず活躍の場が限られている学生には、女性、障害を持つ学生、非白人層 (アフリカ系、ラテン系、先住民)、LGBTQ+ コミュニティのメンバーが含まれます (ただし、これらに限定されない)。

まず、学生は AWS DeepRacer Student にサインアップしてアカウントを作成する必要があります。サインアップ時に、チェックボックスにチェックを入れ AWS AI & 機械学習スカラシッププログラムにオプトインします。アカウントを設定すると、[ホーム] ページに奨学金申請の事前資格審査タスクを追跡するためのタイルが表示されます。奨学金プログラムの申請書を提出するには、学生が AWS DeepRacer Student で 2 つの前提条件を満たしている必要があります。

AWS AI & ML 奨学金の事前資格を得る学生は、Udacity のサイトで引き換えて Udacity Nanodegree 奨学金申請フォームにアクセスできる、独自のアクセスコードを受け取ります。事前資格を持つ学生が Udacity Nanodegree 奨学金の授与を保証されているわけではありません。どの事前資格のある学生に Udacity Nanodegree 奨学金を授与するかは、Udacity が決定します。

本奨学金への事前参加資格を得るためには、以下の 2 つの基準があります。

学習課題をこなし、試験に合格する

事前参加資格を得るためには、すべての必須試験において、80% 以上のスコアを獲得する必要があります。各試験では複数の異なるチャプターが取り上げられ、これらのチャプターは、AWS DeepRacer Student ナビゲーションペインの [学習] セクションから参照できます。奨学金の申請は 2024 年 2 月 1 日に開始されます。

### 最速ラップタイムを達成する

2 月から 9 月までの毎月、新しいコースの導入とともに、新しいスコアボードが AWS DeepRacer League で公開されます。奨学金の事前資格を得るには、いずれか 1 つのレースでシングルラップタイムが 2 分以内を記録する必要があります。

学生が両方の前提条件を満たすと、AWS から一意のアクセスコードと Udacity Nanodegree スカラシップの申請リンクが記載されたメールが届きます。奨学金の受給資格を得るには、学生は申請書に必要な事項をすべて記入する必要があります。申請が完了すると、Udacity から学生に連絡があり、オプションとしてすべての応募者に提供される「Introducing Generative AI with AWS」コースへの登録手順についてお知らせします。Udacity は、Udacity Nanodegree スカラシップの受給対象として選ばれた応募者に連絡します。

## AWS AI & 機械学習スカラシッププログラムと Udacity プラットフォームに関するよくある質問 (FAQ)

AWS AI & ML 奨学金の正式な利用規約について教えてください。

正式な利用規約については、「[Official AWS AI & ML Scholarship program Terms and Conditions](#)」を参照してください。

この奨学金の対象になるのはどのような人が教えてください。

この AWS AI & 機械学習スカラシッププログラムは、この技術の恩恵を十分に受けておらず、そこでの活躍の場も限られている、16 歳以上の学生を対象としたプログラムです。技術の恩恵を受けておらず活躍の場が限られている学生には、女性、障害を持つ学生、非白人層 (アフリカ系、ラテン系、先住民族)、LGBTQ+ コミュニティのメンバーが含まれます (ただし、これらに限定されない)。

Nanodegree とは

Nanodegree とは、学習とキャリア能力の間にあるギャップを埋めるために役立つ、オンラインスキルベースの教育プログラムです。

AWS AI & ML 奨学金の事前資格を取得した場合、Udacity Nanodegree 奨学金を自動的に受け取ることができますか？

いいえ。AWS AI & ML 奨学金の事前資格を取得した学生は、Udacity Nanodegree 奨学金を申請する機会が与えられます。どの事前資格のある学生に Udacity Nanodegree 奨学金を授与するかは、Udacity が決定します。

AWS AI & ML 奨学金に選ばれる前に、入学証明書を提出する必要がありますか？

AWS AI & ML 奨学金を受け取る際に、入学証明書 (大学の成績証明書など) の提出を求められる場合があります。

AWS AI & 機械学習スカラシッププログラムへの応募のために、現在高校に入学していることを示す方法を教えてください。

高校に入学していることを示す方法については、[ステップ 4: プロファイルを更新する](#) Deepracer Student ユーザーガイドの「AWS」を参照してください。

個人を特定できる情報 (PII) データはどのように保護されますか？

個人データは AWS クラウドにより保護されます。詳細については、「[AWS プライバシー通知](#)」を参照してください。

AWS AI & ML 奨学金への応募に関して、年齢制限はありますか？

はい。AWS AI & 機械学習スカラシッププログラムへの応募者は、サインアップ時に 16 歳以上である必要があります。

AWS AI & ML 奨学金を受け取った場合はどのように通知されますか？

応募の受付期限が過ぎた後、ナノディグリー奨学金の応募に関する状況を通知する E メールが、Udacity から送られてきます。

AWS AI & ML 奨学金を受け取ったことはいつ通知されますか？

奨学金のステータスについては、Udacity から応募者に連絡します。奨学金に選ばれた学生には、重要な日付に関する情報が通知されます。

Udacity の AWS 生成系 AI コースにアクセスする方法

学生は、AWS DeepRacer Student で事前資格を取得し、Udacity で奨学金の申請書を提出する必要があります。Udacity は学生が申請書をすべて提出した後に、「Introducing Generative AI with AWS」コースへの登録手順をお知らせします。

AWS AI & 機械学習スカラシッププログラムに受け入れられた場合について教えてください。

AWS AI & ML 奨学金を受け取ることが通知された後、Udacity のオンボーディングチームの担当者から、参加申請を作成した際に Udacity に提出してある E メールアドレスに対して連絡が届きます。

Udacity Nanodegree 奨学金を受け取った場合、Nanodegree 研究にどれくらいの時間を費やすべきですか？

学生は、Udacity Nanodegree 研究に週あたり約 10 時間を費やすことが期待されています。

英語力は必要ですか？

英語力は必須ではありませんが、学習教材は英語です。十分に学ぶために、学生は英語の優れた読み書きのスキルを持っている必要があります。

人種/性別リストの中に、自分に適合する選択肢が見つからない場合の対処法を教えてください。

これらの項目には、[I prefer to self identify (Select to type)]、のオプションを使用できます。

# AWS Player アカウントとは

AWS Player アカウントは、によって作成された AWS DeepRacer マルチユーザーと AWS DeepRacer Student 向けのマネージド ID ソリューションです AWS。AWS Player アカウントは、これらの各 AWS サービスで作成されたすべてのリソースを保持します。

## サポートされているサービスの AWS Player アカウントの作成

[AWS DeepRacer マルチユーザー](#)または [AWS DeepRacer Student](#) のアカウントを作成すると、自動的に AWS Player アカウントが作成されます。これらのサービスで異なる機能を使用すると、新しいリソースが AWS Player アカウントに自動的に追加されます。AWS DeepRacer マルチユーザーと AWS DeepRacer Student の使用を開始するには、次のリンクを使用します。

### AWS DeepRacer Student アカウントの作成

AWS DeepRacer Student を使用するには、アカウントを作成して開始します。アカウントの作成方法については、「[学生ユーザーガイド](#)[ステップ 1: AWS DeepRacer にサインアップする](#)」の「」を参照してください。AWS DeepRacer

AWS DeepRacer マルチユーザーを使用して、1 つのアカウントで複数の参加者をスポンサーします。

AWS DeepRacer マルチユーザーモードは、管理者と参加者という 2 つの異なるユーザープロフィールをサポートします。両者のセットアップ要件は異なります。開始するには、「[AWS DeepRacer デベロッパーガイド](#)」の「[マルチユーザーモード](#)」を参照してください。

## Player AWS アカウントの削除

AWS Player アカウントを削除すると、サポートされているすべてのサービスにすぐにアクセスできなくなります。これには、獲得した成果 (バッジ、ポイント、アバターなど) も含まれます。

AWS Player アカウントを削除しても、AWS アカウントは削除されません。アカウントも削除する場合は、AWS「[アカウントを閉鎖する AWS](#)」で説明されているステップを使用します。

AWS プレイヤーアカウントを使用して AWS DeepRacer マルチユーザーでイベントを作成した場合は、AWS プレイヤーアカウントを削除することはできません。これは、作成したイベントの参加者に残念な経験をさせないようにするためです。管理者が AWS DeepRacer マルチユーザーモードでイベントを作成する方法の詳細については、次のトピックを参照してください。

## AWS DeepRacer マルチユーザーモードを使用したイベントの設定 (管理者)

マルチユーザーモードを使用してイベントを作成する方法については、「AWS DeepRacer デベロッパーガイド」の「[マルチユーザーモード \(管理者\) の設定](#)」を参照してください。

AWS プレイヤーアカウントは、サービスのアカウントで作成されたリソース以外の AWS リソースにアクセスできません。サービスアカウントの AWS Identity and Access Management ポリシーおよび関連リソースは、必要なリソースのみに制限されます。

# AWS DeepRacer Student の開始方法

このチュートリアルを使用して、AWS DeepRacer Student の使用を開始できます。このチュートリアルでは、AWS DeepRacer Student へのログイン、プロフィールの更新、AWS AI & ML 奨学金の検討へのオプトイン、機械学習 (ML) と強化学習 (RL) の無料コースの受講開始、AWS DeepRacer モデルの作成について説明します。アカウントの作成中に奨学金にオプトインしない場合は、後でプロフィールを更新するときに、オプトインできます。

## トピック

- [前提条件](#)
- [ステップ 1: AWS DeepRacer にサインアップする](#)
- [ステップ 2: AWS DeepRacer Student のサインアップを完了する](#)
- [ステップ 3: \(オプション\) AWS AI & ML 奨学金の検討を参照し、オプトインする](#)
- [ステップ 4: プロフィールを更新する](#)
- [ステップ 5: AWS DeepRacer Student を \[Home\] \(ホーム\) ページから探索する](#)

## 前提条件

AWS DeepRacer Student にアクセスして参加するには、次の要件を満たす必要があります。

- 16 歳以上で、高校、コミュニティカレッジ、大学のいずれかに入学している学生であること。
- または、高校、大学、コミュニティカレッジの学生のための教育者またはイベントオーガナイザーになります。
- 有効な E メールアドレス。

## ステップ 1: AWS DeepRacer にサインアップする

この手順にある URL から、AWS DeepRacer Student にサインアップできます。サインアップ後に、AWS Player アカウントを作成します。このアカウントは、他の AWS のサービスで使用できます。AWS Player アカウントを既にお持ちの場合、そのアカウントを AWS DeepRacer Student で使用できます。


1. <http://deepracerstudent.com/> から、ランディングページを開きます。
2. [開始する] を選択します。

3. まだ AWS Player アカウントを持っていない場合、[サインイン] ページで、[サインアップ] を選択します。

 Note

すでに AWS Player アカウントを持っている場合は、ここで情報を入力します。AWS Player アカウントの詳細については、「[AWS Player アカウントとは](#)」を参照してください。

4. [サインアップ] ページで、次の情報を入力します。
  - Email address (E メールアドレス)
  - [パスワード]
5. [サインアップ] を選択します。指定した E メールアドレスに、確認コードを記載した E メールが送信されます。
6. ポップアップ画面が表示されたら、確認番号を入力し、[検証] を選択します。
7. AWS DeepRacer Student の [サインイン] ページで、[E メールアドレス] と [パスワード] を入力し、[サインイン] を選択します。
8. [AWS DeepRacer Student へようこそ] ポップアップ画面で、[サインアップを完了] を選択します。

 Note

[後で行う] を選択できます。後でサインアップする場合は、[いったんサインアウトする] を選択します。

## ステップ 2: AWS DeepRacer Student のサインアップを完了する

AWS DeepRacer Student アカウントを作成するには、このセクションを完了させます。特に明記されていない限り、すべてのフィールドは必須です。

1. [個人情報を追加して AWS DeepRacer Student アカウントを作成します] セクションのフィールドに入力して、アカウントを作成します。
2. チェックボックスをオンにして、高校、大学、コミュニティカレッジのいずれかに在籍している学生であることを証明します。



**Note**

高校生の場合は、次の操作を実行します。

- [学校] で、[その他] を選択します。次に、[学校の名前を入力] フィールドに、高等学校の名前を追加します。
- [現在または将来の専攻] から、[未定] を選択するか、将来選択する可能性のある専攻をリストから選択します。

## ステップ 3: (オプション) AWS AI & ML 奨学金の検討を参照し、オプトインする

AWS AI & ML 奨学金についてと、その対象者については、[Do you want to be considered for the AWS AI & ML Scholarship program?] (AWS AI & ML 奨学金プログラムの対象者になるには) セクションを参照してください。

1. AWS AI & ML 奨学金の基準を満たしている場合は、オプトインを検討できます。スカラーシップの検討を確定するには、チェックボックスを選択します。

**Note**

スカラーシップの受け取りを検討しない、または未定である場合は、チェックボックスをオフのままにして、[I will do this later] (後で行う) を選択します。[Sign out for now] (いったんサインアウトする) でいったんサインアウトするか、[Submit] (送信) でオプトインせずに続行します。プロフィールを更新するときに、オプトインすることもできます。

2. (オプション) [Choose gender] (性別を選択) と [Choose race (US participants only)] (レースを選択 (米国の参加者のみ)) フィールドでは、ドロップダウンリストを使用して情報を入力します。
3. [Submit] (送信) を選択します。

## ステップ 4: プロファイルを更新する

[Your profile] (プロフィール) ページでプロフィールを更新します。AWS AI & ML 奨学金プログラムへのオプトインも選択できます。AWS DeepRacer Student League で受け取ったパフォーマンスの実績は、[プロフィール] ページからソーシャルメディアで共有できます。

ユーザープロフィールを更新するには

1. AWS DeepRacer Student の左にあるナビゲーションペインで、[プロフィール] を選択します。
2. [プロフィール] ページの [レーサー名] で、[レーサー名を変更] を選択します。

### Note

レーサーの名前は、2~24 文字です。文字、数字、ハイフン (-) を使用できます。

3. [レーサー名] モーダル画面で、レーサー名を入力し、[保存] を選択します。レーサーの名前を変更しない場合は、[キャンセル] を選択します。
4. [プロフィール情報] セクションでは、次のフィールドを変更できます。
  - 名前
  - レーサー名
  - 学校名
  - 専攻名
  - 卒業年
  - 居住国

### Note

AWS DeepRacer 学生リーグの仮想化レースに参加して賞品を受け取るには、居住国の選択が必要です。モデルをレースに提出すると、レースシーズンの間、居住国が固定されます。

5. (オプション) 「Do you want to be considered for the AWS AI & ML Scholarship program?」セクションで、スカラーシップについての情報と、申請方法を確認できます。
  - a. 奨学金プログラムの検討をオプトインするチェックボックスを選択します。

- b. (オプション) [Choose gender] (性別を選択) および [Choose race] (レースを選択) フィールドに情報を入力します。
- c. [Submit] (送信) を選択します。

## ステップ 5: AWS DeepRacer Student を [Home] (ホーム) ページから探索する

AWS DeepRacer Student の [Home] (ホーム) ページは、AWS DeepRacer Student が提供するすべての探索を開始するのに最適な場所です。[ホーム] ページでは、次の操作を実行できます。

機械学習 (ML) の基礎を学び始めましょう。

AWS DeepRacer Student の [学習] セクションで、無料コースが利用できます。この強力な講義は、AWS DeepRacer Student を使った機械学習ジャーニーの基礎を固めるのに役立ちます。

機械学習の知識を使って実践しましょう。

しばらくの間 [学習] コースで時間を費やせば、AWS DeepRacer モデルを作成してトレーニングできるようになります。詳細については、「[AWS DeepRacer Student での強化学習モデルのトレーニング](#)」を参照してください。

AWS DeepRacer Student レースで競争します。

最初の AWS DeepRacer モデルのトレーニングが完了すれば、レースに参加する準備が整います。学生リーグの月間レースに参加する場合は、[オープンレース] セクションに移動して [レースに参加する] を選択します。レースを完了するとリーダーボードでモデルの成績を確認し、モデルを変更して改善するために必要なデータも確認します。毎月の仮想化レースに参加するとポイントを獲得できます。これらのポイントにより、国および地域のシーズン順位が決まります。毎月上位 3 名のレーサーがメダルと賞品を獲得します。詳細については、「[AWS DeepRacer Student レースに参加する](#)」を参照してください。

モデルトレーニングの時間を確認します。

モデルのトレーニングとクローンの作成には、無料のモデルトレーニング時間の一部が使われます。モデルトレーニングの残り時間は、ホームページで確認できます。

AWS DeepRacer Student のレースカレンダーを確認します。

レースカレンダーを見て、レースをする日を計画しましょう。

AWS AI & ML 奨学金の詳細について説明します。

AWS AI & ML Scholarship について知ることができ、それを準備するために何ができるのかが分かります。詳細については、「[AWS AI & 機械学習スカラシッププログラムとは](#)」を参照してください。

その他のリソースを表示する

他にも、Discord チャンネルや AWS DeepRacer のウェブサイトなど、AWS DeepRacer Student の探索に役立つリソースを発見できます。これらのリソースから、レーサーやファンのコミュニティと繋がることができ、ヒントやインサイトを共有できます。

# AWS DeepRacer Student での強化学習モデルのトレーニング

このチュートリアルでは、AWS DeepRacer Student で最初のモデルをトレーニングする方法を説明します。また、経験を最大限に活用し、学習を迅速に進めることに役立つヒントを示します。

## ステップ 1: AWS DeepRacer Student を使用して強化学習モデルをトレーニングする

[Create model] (モデルの作成) ボタンの場所を確認して AWS DeepRacer Student を開始し、最初のモデルのトレーニングを開始します。モデルの作成とトレーニングは反復プロセスであることに注意してください。最適な結果を得るために、さまざまなアルゴリズムと報酬関数で実験します。

強化学習モデルをトレーニングするには

1. AWS DeepRacer Student [Home] (ホーム) ページで、[Create a model] (モデルの作成) を選択します。または、左のナビゲーションペインで [Your Models] (モデル) に移動します。[Models] (モデル) ページの [Your models] (モデル) で、[Create model] (モデルの作成) を選択します。
2. [Overview] (概要) ページで、強化モデルをトレーニングする方法を読んでください。プロセスの各ステップについて、このページで説明されています。読み終えたら、[Next] (次へ) をクリックします。

## ステップ 2: モデルに名前を付ける

モデルに名前を付けます。モデルを改良してクローンを作成する場合は、モデルを素早く見つけれられるよう、モデルに一意の名前を付けることをお勧めします。例えば、*yourinitials-date-version* のような命名規則を使用してモデルに名前を付けます。

モデルに名前を付けるには

1. [Name your model] (モデルに名前を付ける) ページで、[Model name] (モデル名) フィールドに名前を入力します。

**Note**

モデルのトレーニングを開始すると、モデルの名前は確定され、変更できなくなります。

2. [Next] を選択します。

## ステップ 3: トラックを選択する

シミュレーショントラックを選択します。トラックは環境として機能し、車両にデータを提供します。非常に複雑なトラックを選択すると、車両のトレーニングに必要な時間は全体として長くなり、使用する報酬関数は複雑になります。

トラック (環境) を選択するには

1. [Choose track] (トラックの選択) ページで、車両のトレーニング環境として機能するトラックを選択します。
2. [Next] を選択します。

## ステップ 4: アルゴリズムを選択する

AWS DeepRacer Student には 2 つのトレーニングアルゴリズムがあり、そこから選択します。異なるアルゴリズムによって、さまざまな方法で報酬が最大化されます。AWS DeepRacer Student の経験を最大限に活用するために、両方のアルゴリズムで実験します。アルゴリズムの詳細については、「[AWS DeepRacer Training Algorithms](#)」を参照してください。

トレーニングアルゴリズムを選択するには

1. [Choose algorithm type] (アルゴリズムタイプの選択) ページで、アルゴリズムタイプを選択します。次の 2 つのアルゴリズムタイプを使用できます。
  - Proximal Policy Optimization (PPO)。このアルゴリズムは、安定していますが、データが大量に必要です。トレーニングを反復するたびに一貫性のある実行が行われます。
  - Soft Actor Critic (SAC)。このアルゴリズムは、不安定ですが、データを効率的に使用します。トレーニングを反復するたびに一貫性のない実行を行う可能性があります。
2. [Next] を選択します。

## ステップ 5: 報酬関数をカスタマイズする

報酬関数は強化学習の中核です。報酬関数を使用して、トラック (環境) を探索するときに車両 (エージェント) が特定のアクションを実行するようにインセンティブを与えます。ペットの特定の行動を褒めたり叱ったりするように、このツールを使用して、車両ができるだけ早くラップを終えると報酬を得て、トラックから外れたりジグザグに走ったりすると報酬を失うようにすることができます。

最初のモデルをトレーニングするときは、デフォルトのサンプル報酬関数を使用したい場合があります。モデルを実験して最適化する準備ができたら、コードエディタでコードを編集して報酬関数をカスタマイズできます。報酬関数のカスタマイズの詳細については、「[報酬関数のカスタマイズ](#)」を参照してください。

報酬関数をカスタマイズするには

1. [Customize reward function] (報酬関数のカスタマイズ) ページで、サンプル報酬関数を選択します。カスタマイズできる報酬関数のサンプルには、次の 3 つがあります。
  - 中心線に従う。車線の中心線にできる限り近く自律的に運転すると、車に報酬を与えます。
  - 境界内に留まる。四輪すべてを車線の境界内に留めて自律的に運転すると、車両に報酬を与えます。
  - ジグザグ回避する。中心線の近くに留まることで車両に報酬を与えます。ステアリングの角度が大きかったり、車線から外れたりすると、車両にペナルティを与えます。

### Note

報酬関数をカスタマイズしない場合は、[Next] (次へ) をクリックします。

2. (オプション) 報酬関数のコードを変更します。
  - サンプル報酬関数を選択し、[Walk me through this code] (コードのウォークスルー) を選択します。
  - コードの各セクションについて、[+] をクリックすると、ポップアップテキストボックスに説明が表示されます。チュートリアルを先に進めるには、各ポップアップで [Next] (次へ) をクリックします。ポップアップテキストボックスを終了するには、コーナーの [X] をクリックします。チュートリアルを終了するには、[Finish] (完了) をクリックします。

**Note**

[Go with default code] (デフォルトコードを使用) を選択して、サンプル報酬関数コードを編集しないこともできます。

- 必要に応じて、サンプル報酬関数を選択し、[Edit sample code] (サンプルコードを編集) を選択して、サンプル報酬関数コードを編集します。コードを編集し、[Validate] (検証) をクリックしてコードを確認します。コードを検証できない場合、またはコードを元の状態にリセットする場合は、[Reset] (リセット) をクリックします。

3. [Next] を選択します。

## ステップ 6: 期間を選択し、モデルをリーダーボードに送信する

モデルトレーニングの期間は、パフォーマンスに影響があります。トレーニングの初期段階で実験を行うときは、このパラメータを小さい値から始め、徐々にトレーニング期間を長くします。

モデルトレーニングのこのステップでは、トレーニング済みモデルがリーダーボードに送信されます。チェックボックスの選択を解除すると、オプトアウトできます。

期間を選択し、モデルをリーダーボードに送信するには

1. [Choose duration] (期間の選択) ページの [Choose duration of model training] (モデルトレーニングの期間を選択) で期間を選択します。
2. [Model description] (モデルの説明) フィールドに、選択した内容を記憶するのに役立つモデルの説明を入力します。

**Tip**

報酬関数とアルゴリズムの現在の選択や変更、モデルのパフォーマンスに関する仮説など、モデルに関する情報を追加することをお勧めします。

3. チェックボックスをオンにすると、トレーニング完了後、モデルが AWS DeepRacer Student リーダーボードに自動的に送信されます。オプションで、チェックボックスの選択を解除して、モデルの入力をオプトアウトできます。



**i** Tip

モデルをリーダーボードに送信することをお勧めします。モデルを送信すると、モデルを他のモデルと比較するのに役立ち、フィードバックを得てモデルを改善できます。

4. [Train your model] (モデルのトレーニング) を選択します。
5. [Initializing model training] (モデルトレーニングの初期化) ポップアップで、[Okay] (OK) をクリックします。
6. [Training configuration] (トレーニング設定) ページで、モデルトレーニングのステータスと設定を確認できます。トレーニングの [Status] (ステータス) が [In progress] (進行中) のときに、選択したトラックでモデルトレーニングのビデオを表示することもできます。ビデオを見ることは、モデルの改善に使用できる貴重なインサイトを得るのに役立ちます。

## ステップ 7: リーダーボードでモデルのパフォーマンスを表示する

モデルをトレーニングしてリーダーボードに送信すると、そのパフォーマンスを表示できます。

モデルのパフォーマンスを表示するには

1. 左のナビゲーションペインで、[Compete] (競争) に移動して展開します。シーズンを選択します。[Leaderboard] (リーダーボード) ページのセクションにモデルとランクが表示されます。このページには、送信されたモデルのリストを表示する [Leaderboard] (リーダーボード) セクション、レースの詳細、および [Race details] (レースの詳細) セクションも含まれます。
2. リーダーボードが表示されているページで、プロフィールのセクションで、[Watch Video] (ビデオを見る) をクリックして、モデルのパフォーマンスのビデオを表示します。

## ステップ 8: [Clone] (クローン) を使用してモデルを改善する

モデルをトレーニングし、必要に応じてリーダーボードに送信した後、モデルのクローンを作成して改善できます。モデルのクローンを作成すると、新しいモデルのスタートポイントとして、以前にトレーニングしたモデルを使用することでステップが削減され、トレーニングの効率が向上します。

モデルのクローンを作成してモデルを改善するには

1. 左のナビゲーションペインの AWS DeepRacer Student で、[Your models] (モデル) に移動します。

2. [Your models] (モデル) ページで、モデルを選択し、[Clone] (クローン) を選択します。
3. [Name your model] (モデルに名前を付ける) フィールドで、クローンとして作成したモデルの新しい名前を入力し、[Next] (次へ) をクリックします。
4. [Customize a reward function] (報酬関数をカスタマイズ) ページで、報酬関数をカスタマイズし、[Next] (次へ) を選択します。報酬関数のカスタマイズの詳細については、「[ステップ 5: 報酬関数をカスタマイズする](#)」を参照してください。
5. [Choose duration] (期間を選択) ページで、[Choose duration of model training] (モデルトレーニングの期間を選択) フィールドに時間を入力し、[Model description] (モデルの説明) フィールド説明を入力し、チェックボックスをオンにしてクローンとして作成したモデルをリーダーボードに送信します。
6. [Train your model] (モデルのトレーニング) を選択します。トレーニングが初期化されます。[Training configuration] (トレーニング設定) ページが表示され、クローンとして作成されたモデルに関する情報が表示されます。トレーニングの [Status] (ステータス) が [In progress] (進行中) のときに、選択したトラックでモデルトレーニングのビデオを表示することもできます。
7. リーダーボードで最高のパフォーマンスを達成するために、事前トレーニングされたモデルのクローン作成と修正を続けます。

## ステップ 9: (オプション) モデルをダウンロードする

モデルをトレーニングし、必要に応じてリーダーボードに送信した後、将来の使用のためにモデルを AWS DeepRacer 物理デバイスにダウンロードすることができます。モデルは .tar.gz ファイルとして保存されます。

モデルをダウンロードするには

1. 左のナビゲーションペインの AWS DeepRacer Student で、[Your models] (モデル) に移動します。
2. [Your models] (モデル) ページで、モデルを選択し、[Download] (ダウンロード) を選択します。
3. ブラウザでモデルのダウンロードの進行状況を追跡します。モデルをダウンロードすると、ローカルハードドライブまたはその他の優先ストレージデバイスに保存できます。

AWS DeepRacer デバイスでの作業の詳細については、[AWS DeepRacer ガイド](#)の「AWS DeepRacer 車両の運転」を参照してください。

# AWS DeepRacer Student レースに参加する

シミュレーションでモデルのトレーニングと評価を完了したら、レースに参加して、モデルのパフォーマンスを他のレーサーのモデルと比較しましょう。レースは、モデルに関するフィードバックを受け、賞と賞品を獲得し、AWS DeepRacer Student コミュニティの他のメンバーとバーチャルに交流して、学習とスキル向上の機会に関する情報を得るとともに、楽しい時間を過ごすための素晴らしい方法です。Student レースには、学生リーグレースと学生コミュニティレースの2つのタイプがあります。

学生リーグレースは毎月開催されるバーチャルレースで、すべての学生が参加できます。学生コミュニティレースは教育者またはイベント主催者が AWS で作成するプライベートレースで、参加できるのは招待された学生のみとなります。

このセクションでは、AWS DeepRacer Student の学生リーグレースと学生コミュニティレースに参加する方法について説明します。

## 学生リーグレースに参加する

このセクションでは、AWS DeepRacer Student の学生リーグレースにモデルを送信する方法を学びます。毎月レースには、トレーニングされたモデルを学生リーダーボードに直接送信することによって参加できます。モデルのトレーニングに関する詳細については、「[AWS DeepRacer Student での強化学習モデルのトレーニング](#)」を参照してください。

### 学生リーグレースに参加する

1. AWS DeepRacer Student の左側のナビゲーションペインで [競技] を展開し、[学生リーグ] を選択します。
2. ページにはレースの詳細とリーダーボードが表示されます。[オープンレース] セクションで、[レースに参加する] を選択してモデルを送信するか、まだモデルを作成していない場合は [モデルを作成] を選択します。詳細については、「[AWS DeepRacer Student での強化学習モデルのトレーニング](#)」を参照してください。
3. [Choose a model to race] (レースするモデルを選択する) セクションにある [Choose a model] (モデルを選択する) フィールドで、ドロップダウンリストを使用してモデルを選択します。
4. [レースに参加する] を選択してモデルを送信します。
5. レース基準に照らしたモデルの評価が正常に完了したら、リーダーボードをチェックして、他の参加者と比較したモデルのランク付けを確認してください。

6. オプションで [Watch] (見る) を選択して、車のパフォーマンスの動画を視聴します。
7. [Race again] (再度レースする) を選択して、別のモデルを参加させます。AWS DeepRacer 学生リーグの各レースの提出枠は 50 です。

## 学生コミュニティレースに参加する

このセクションでは、AWS DeepRacer Student の学生コミュニティレースにモデルを送信する方法を学びます。学生コミュニティレースには、電子メールで教育者またはイベント主催者からの招待リンクを受け取ることによって参加できます。

### 学生コミュニティレースに参加する

1. 招待リンクに移動して、AWS DeepRacer Student アカウントにログインします。
2. サインインされたら、[Enter race] (レースに参加する) ボタンを選択します。
3. [Choose a model to race] (レースするモデルを選択する) ドロップダウンリストで、コミュニティレースで使用するモデルを選択します。
4. [Enter race] (レースに参加する) を選択してモデルを送信します。
5. レース基準に照らしたモデルの評価が正常に完了した場合は、リーダーボードをチェックして、他の参加者のモデルと比較したモデルのランク付けを確認してください。
6. オプションで [Watch] (見る) を選択して、車のパフォーマンスの動画を視聴します。
7. [Race again] (再度レースをする) を選択して、別のモデルを登録します。

# 報酬関数のカスタマイズ

報酬関数の作成は、インセンティブプランを設計するのに似ています。パラメータは、インセンティブプランの作成に使用できる値です。

インセンティブ戦略が異なると、車両の動作も異なります。車両が速く走るようにするには、ラップの終了に時間がかかりすぎたり、トラックから外れた際に負の値を付けてください。ジグザグな運転パターンを避けるために、ステアリング角度の範囲制限を定義して、トラックのまっすぐなセクションで積極的にステアリングを使用しない車両に報酬を与えるようにしてください。

トラックの中心線およびトラックの外側と内側のエッジに沿って配置された番号付きの指標であるウェイポイントを使用して、直線や曲線などのトラックの特徴と特定の運転動作を関連付けることができます。

効果的な報酬関数の作成は、クリエイティブで反復的なプロセスです。さまざまな戦略を試し、パラメータをミックスして一致させ、そして何より、楽しんでください。

トピックス

- [Python コードを編集して報酬関数をカスタマイズする](#)
- [AWS DeepRacer 報酬関数の入力パラメータ](#)

## Python コードを編集して報酬関数をカスタマイズする

AWS DeepRacer Student で、サンプルの報酬関数を編集し、モデルのカスタムレース戦略を作成できます。

報酬関数をカスタマイズするには

1. AWS DeepRacer Student の Create model (モデルの作成) エクスペリエンスの [Step 5: Customize reward function] (ステップ 5: 報酬関数をカスタマイズする) ページで、サンプルの報酬関数を選択します。
2. Python コードを使用して報酬関数の入力パラメータをカスタマイズするには、サンプルの報酬関数ピッカーの下にあるコードエディタを使用します。
3. [Validate] (検証) を選択して、コードが動作するかどうかをチェックします。または、[Reset] (リセット) を選択してやり直してください。
4. 変更が完了したら、[Next] (次へ) を選択します。

[AWS DeepRacer 報酬関数の入力パラメータ](#) を参照して、各パラメータを確認してください。報酬関数の例で、さまざまなパラメータがどのように使われているかを確認してください。

## AWS DeepRacer 報酬関数の入力パラメータ

AWS DeepRacer 報酬関数では、変数として渡された params 辞書オブジェクトを入力として受け取ります。

```
def reward_function(params) :  
  
    reward = ...  
  
    return float(reward)
```

params 辞書オブジェクトには、次のキーと値のペアが含まれています。

```
{  
    "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

}
```

次のリファレンスを参照して、AWS DeepRacer の入力パラメータに関する理解を深めてください。

## all\_wheels\_on\_track

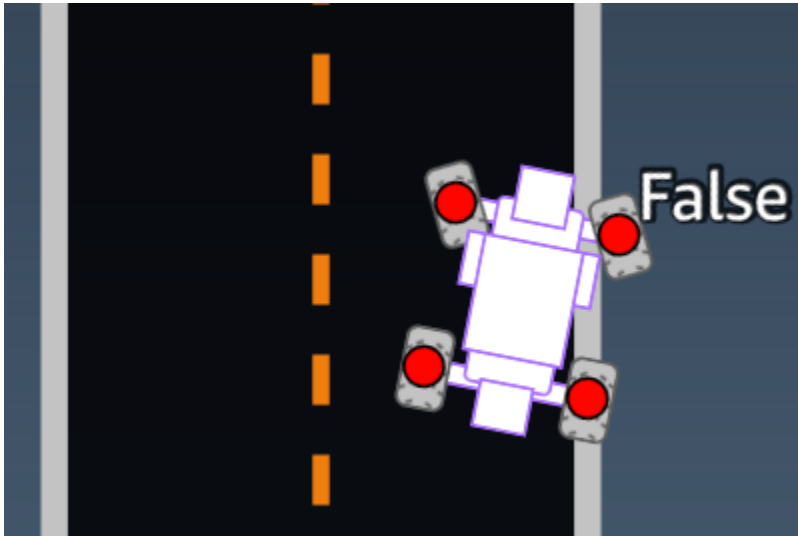
タイプ: Boolean

範囲: (True:False)

エージェントがトラック上にあるか否かを示す Boolean フラグ。車輪のいずれかがトラックの境界線の外側にある場合、エージェントはトラック上にありません (False)。4 つの車輪すべてがトラックの境界の内側にある場合、トラック上にあります (True)。次のイラストレーションは、エージェントがトラック上にあることを示しています。



次のイラストレーションでは、2つの車輪がトラックの境界の外側にあるため、エージェントはトラック上にありません。



例: `all_wheels_on_track` パラメータを試用した報酬関数

```
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

タイプ: [int, int]

範囲: [(0:Max-1), (1:Max-1)]

waypoint のエージェントの現在位置に最も近い 2 つが隣接する (x, y) のゼロベースのインデックス。距離は、エージェントの中心からのユークリッド距離によって測定されます。最初の要素は、エージェントの背後に最も近いウェイポイントを指し、2 番目の要素は、エージェントの前面にある最も近いウェイポイントを指します。Max は、ウェイポイントリストの長さです。[ウェイポイント](#) で示しているイラストレーションでは、closest\_waypoints は [16, 17] になります。

次の例の報酬関数は、waypoints と closest\_waypoints、および heading を使用して即時報酬を計算する方法を示しています。

AWS DeepRacer では次の Python ライブラリがサポートされています:

math、random、numpy、scipy、および shapely。これらの 1 つを使用するには、関数定義 `import supported library` の前に、インポートステートメント `def reward_function(params)` を追加します。

例: `closest_waypoints` パラメータを使用する報酬関数

```
# 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

タイプ: [int, int]

範囲: [(0:len(object\_locations)-1), (0:len(object\_locations)-1)]

エージェントの現在の位置 (x, y) に最も近い 2 つのオブジェクトのゼロから始まるインデックス。最初のインデックスは、エージェントの背後にある最も近いオブジェクトを参照し、2 番目のイ

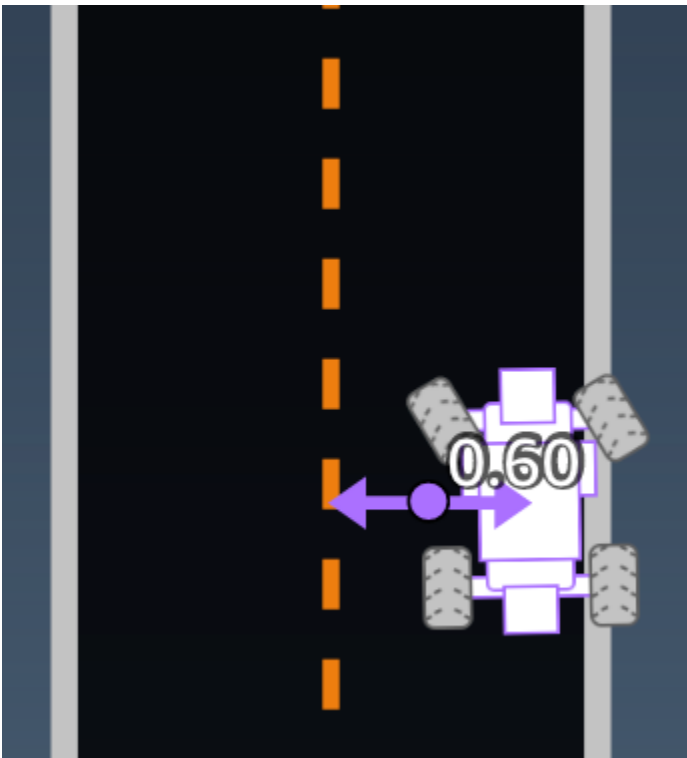
インデックスは、エージェントの前面にある最も近いオブジェクトを参照します。オブジェクトが 1 つしかない場合、両方のインデックスは 0 です。

## distance\_from\_center

タイプ: float

範囲: 0:~track\_width/2

エージェントの中心とトラックの中心との間の変位 (メートル単位)。観察可能な最大変位は、エージェントのいずれかの車輪がトラックの境界線の外側にあるときに発生し、トラックの境界線の幅に応じて、track\_width の半分よりわずかに小さいまたは大きい場合があります。



例: `distance_from_center` パラメータを使用する報酬関数

```
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

タイプ: float

範囲: -180:+180

座標の x 軸に対するエージェントの進行方向 (度単位)。



例: *heading* パラメータを使用する報酬関数

([=closest\\_waypoints](#)) 詳しくは、「」を参照してください。

## is\_crashed

タイプ: Boolean

範囲: (True:False)

エージェントが終了ステータスとして別のオブジェクトにクラッシュしたか (True) 否か (False) を示す Boolean フラグ。

## is\_left\_of\_center

タイプ: Boolean

範囲: [True : False]

エージェントがトラックの中心から左にあるか (True) 否か (False) を示す Boolean フラグ。

## is\_offtrack

タイプ: Boolean

範囲: (True:False)

エージェントの 4 つの車輪すべてがトラックの内側または外側の境界線の外を走っているか (True) 否か (False) を示す Boolean フラグ。

## is\_reversed

タイプ: Boolean

範囲: [True:False]

エージェントが時計回り (True) であるか反時計回り (False) のどちらで運転しているかを示す Boolean フラグ。

これは、エピソードごとに方向変更を有効にする場合に使用されます。

## objects\_distance

タイプ: [float, ... ]

範囲: [(0:track\_length), ... ]

開始線に対する環境内のオブジェクト間の距離のリスト。i 番目の要素は、i 番目のオブジェクトと、トラックの中心線に沿った開始線間の距離をメートルで測定します。

#### Note

$abs | (var1) - (var2) |$  = how close the car is to an object, WHEN  $var1 = ["objects\_distance"]$   
 $[index]$  and  $var2 = params["progress"] * params["track\_length"]$

車両の前面に最も近いオブジェクトと車両の背後に最も近いオブジェクトのインデックスを取得するには、closest\_objects パラメータを使用します。

## objects\_heading

タイプ: [float, ... ]

範囲: [(-180:180), ... ]

オブジェクトの見出しのリスト ( 度単位 )。i 番目の要素は、i 番目の オブジェクトの見出しを測定します。静止オブジェクトの見出しは 0 です。ポットカーの場合、対応する要素の値はその車両の見出しの角度です。

## objects\_left\_of\_center

タイプ: [Boolean, ... ]

範囲: [True|False, ... ]

Boolean フラグのリスト。i 番目の要素の値は、i 番目のオブジェクトがトラックの中心から左側 (True) か右側 (False) かを示します。

## objects\_location

タイプ: [(x,y), ...]

範囲: [(0:N,0:N), ...]

このパラメータには、すべてのオブジェクトの位置が格納されます。それぞれの位置は (x, y) のタプルです。

リストのサイズは、トラック上のオブジェクトの数と同じです。リストされるオブジェクトには、静止オブジェクトと動いているボットカーの両方が含まれます。

## objects\_speed

タイプ: [float, ... ]

範囲: [(0:12.0), ... ]

トラック上のオブジェクトの速度 (メートル/秒) のリスト。静止オブジェクトの場合、速度は 0 です。ボットカーの場合、値はトレーニングで設定した速度です。

## progress

タイプ: float

範囲: 0:100

トラック完走の割合。

例: *progress* パラメータを使用する報酬関数

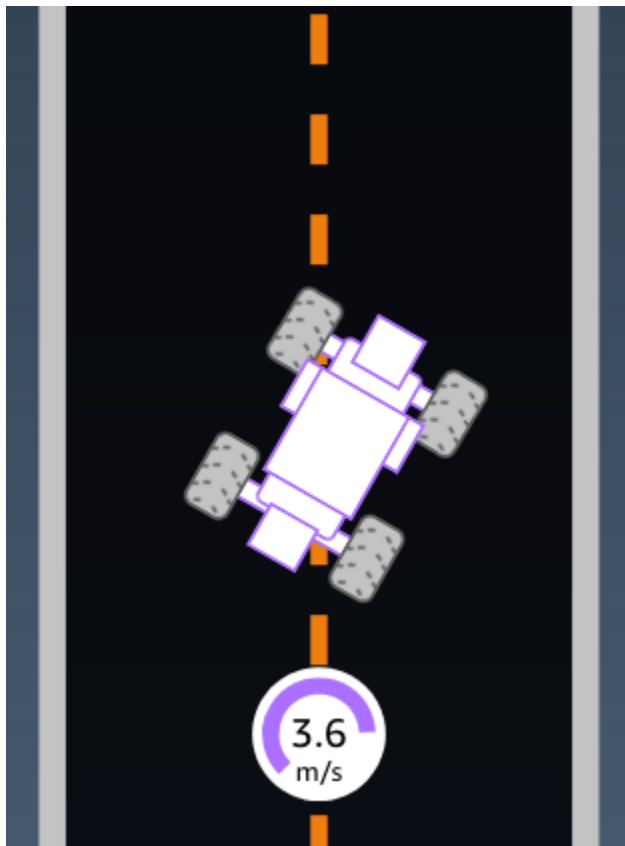
詳細については、「[ステップ](#)」を参照してください。

## 速度

タイプ: float

範囲: 0.0:5.0

エージェントの観測速度 (メートル/秒)。



例: *speed* パラメータを使用する報酬関数

詳細については、「[all\\_wheels\\_on\\_track](#)」を参照してください。

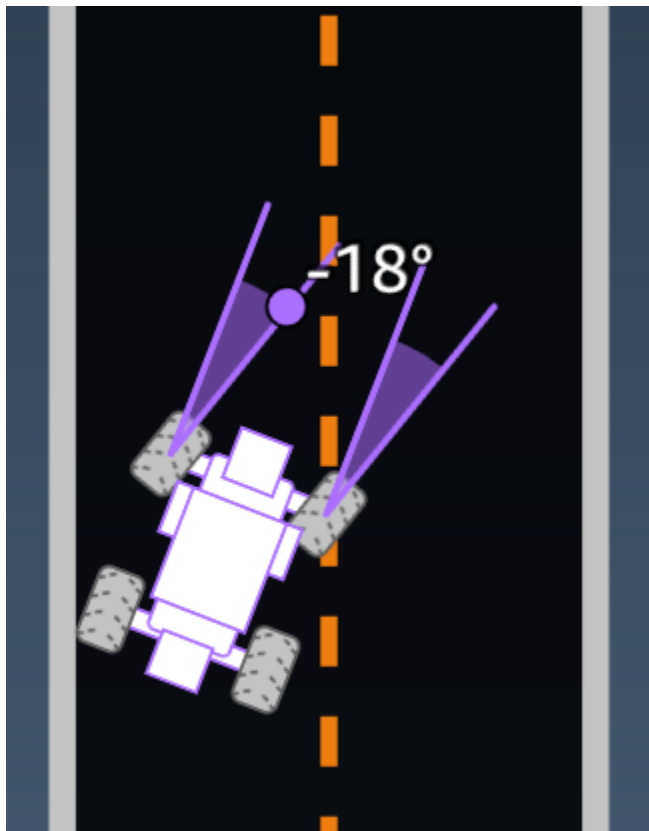
## steering\_angle

タイプ: float

範囲: -30:30

エージェントの中心線からの前輪のステアリング角 (度単位)。負の記号 (-) は右へのステアリングを意味し、正の (+) 記号は左へのステアリングを意味します。次の図に示すように、エージェントの中心線はトラックの中心線と必ずしも平行ではありません。





例: *steering\_angle* パラメータを使用する報酬関数

```
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)
```

## ステップ

タイプ: int

範囲:  $0:N_{\text{step}}$

完了した手順の数。手順は、現在のポリシーを使用してエージェントが行う観察アクションのシーケンスに対応します。

例: `steps` パラメータを使用する報酬関数

```
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

タイプ: float

範囲:  $[0:L_{\text{max}}]$

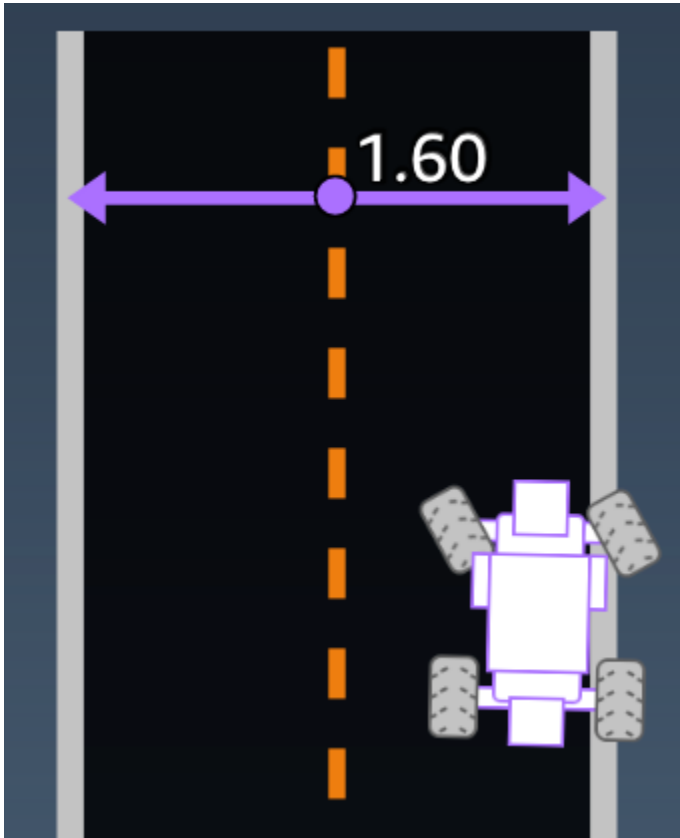
トラックの長さ (メートル単位)。  $L_{\text{max}}$  is track-dependent.

## track\_width

タイプ: float

範囲:  $0:D_{\text{track}}$

トラックの幅 (メートル)。



例: *track\_width* パラメータを使用する報酬関数

```
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

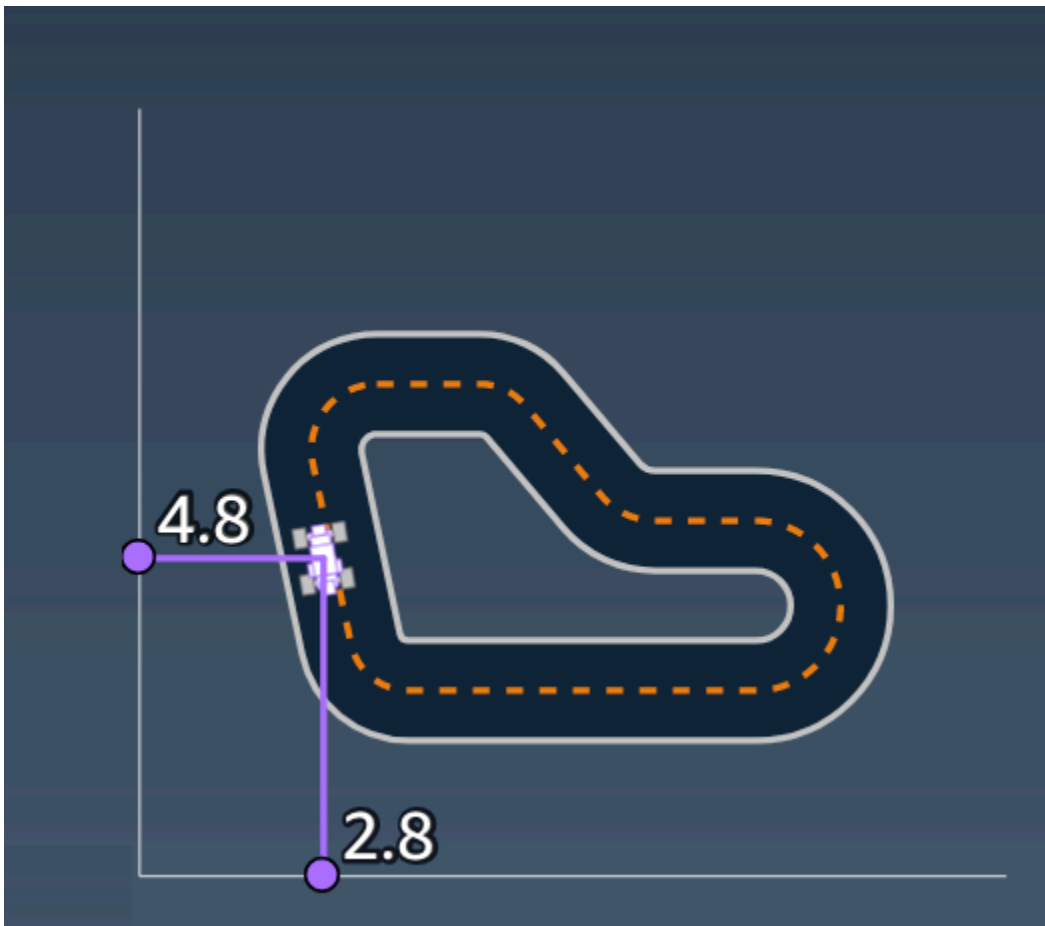
return float(reward)
```

x、y

タイプ: float

範囲: 0:N

トラックを含むシミュレーション環境における x 軸および y 軸上のエージェントの中心の位置 (メートル単位)。原点は、シミュレーション環境の左下隅にあります。

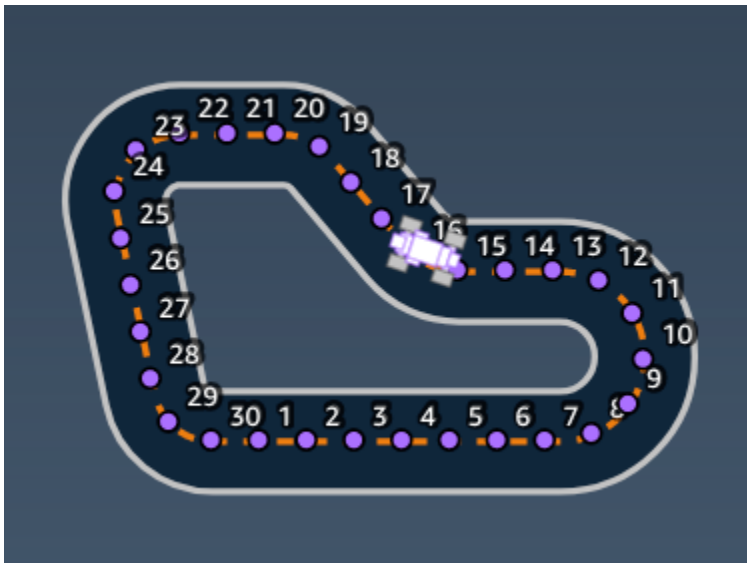


## ウェイポイント

タイプ: [float, float] の list

範囲:  $[[x_{w,0}, y_{w,0}] \dots [x_{w,Max-1}, y_{w,Max-1}]]$

トラックの中心に沿ったトラック依存 Max マイルストーンの順序付きリスト。各マイルストーンは、 $(x_{w,i}, y_{w,i})$  の座標で表されます。ループされたトラックの場合、最初と最後のウェイポイントは同じです。直線のトラックなどループされないトラックの場合、最初と最後のウェイポイントは異なります。



例 *waypoints* パラメータを使用する報酬関数

([=closest\\_waypoints](#)) 詳しくは、「」を参照してください。

# AWS DeepRacer Student のセキュリティ

AWS では、クラウドのセキュリティが最優先事項です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャから利点を得られます。

セキュリティは、AWS とお客様の間の共有責任です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS のサービスを実行するインフラストラクチャを保護する責任を負います。また、AWS は、使用するサービスを安全に提供します。[AWS コンプライアンスプログラム](#)の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。AWS DeepRacer Student に適用されるコンプライアンスプログラムの詳細については、「[コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)」を参照してください。
- クラウド内のセキュリティ - お客様の責任は、使用する AWS のサービスに応じて異なります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、AWS DeepRacer Student を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。ここでは、セキュリティとコンプライアンスの目的を満たすように AWS DeepRacer Student を設定する方法について説明します。また、AWS DeepRacer Student のリソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

## 目次

- [AWS DeepRacer Student でのデータ保護](#)
- [AWS DeepRacer Student 用の Identity and Access Management](#)
- [AWS DeepRacer Student のコンプライアンス検証](#)
- [AWS DeepRacer Student の耐障害性](#)
- [AWS DeepRacer Student のインフラストラクチャセキュリティ](#)

# AWS DeepRacer Student でのデータ保護

次のセクションでは、AWS DeepRacer Student がどのようなデータをキャプチャするのかと、AWS DeepRacer Student がデータを保護するためにどこでデータ暗号化を使用しているのかを説明します。

AWS DeepRacer Student アカウントを作成すると、AWS Player プレイヤーアカウントも作成されます。AWS DeepRacer Student アカウントで作成したリソースは、AWS Player プレイヤーアカウントに保存されます。AWS Player アカウントの詳細については、AWS DeepRacer Student ユーザーガイドの「[AWS Player アカウントとは](#)」を参照してください。

## トピック

- [AWS DeepRacer Student ポータルでキャプチャしたデータ](#)
- [AWS DeepRacer Student ポータルへの保管時の暗号化](#)
- [AWS DeepRacer Student ポータルへの転送中の暗号化](#)

## AWS DeepRacer Student ポータルでキャプチャしたデータ

AWS DeepRacer Student ポータルを使用すると、AWS Player アカウントに必要なデータが保存されます。AWS DeepRacer Student ポータルでキャプチャしたデータは、サービス改善のためには使用されません。

AWS DeepRacer Student がキャプチャしたデータ。

AWS DeepRacer Student で作成し、AWS Player アカウントに保存されるデータの概要は次の通りです。

- アカウント登録に使用した E メールアドレスとパスワード。
- レーサーの名前
- Student League リーダーボードの順位
- トレーニングしたデータ
- 報酬機能コード

## AWS DeepRacer Student ポータルへの保管時の暗号化

AWS DeepRacer Student ポータルでキャプチャしたデータは、デフォルトで暗号化されています。

AWS Player アカウントは、Amazon Cognito を使用して、AWS DeepRacer Student へのログインに使用した E メールとパスワードを暗号化して保存します。詳細については、「[Amazon Cognito でのデータ保護](#)」を参照してください。

AWS DeepRacer Student でキャプチャしたデータは、すべてクラウドへの保管時に、AWS が所有するキーによって暗号化されます。キーは AES-GCM の AWS Key Management Service を使用し、256 ビットのキーを使用します。このデータは、Amazon Simple Storage Service (S3) と Amazon DynamoDB に保存され、暗号化されます。

## AWS DeepRacer Student ポータルへの転送中の暗号化

登録された E メールアドレスと、承認された E メールアドレスは、クライアント側の暗号化によって暗号化されます。その他の [AWS DeepRacer Student でキャプチャしたデータ](#) はすべてアカウントからコピーされ、AWS システム内部で処理されます。デフォルトでは、AWS DeepRacer Student は、HTTPS 経由のセキュリティで保護された接続を使用して、転送中のデータを暗号化します。

## AWS DeepRacer Student 用の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者がリソースへのアクセスを安全に制御するのに役立つ AWS のサービスです。IAM の管理者は、誰を認証 (サインイン) し、誰に AWS リソースの使用を許可する (アクセス許可を持たせる) かを制御できます。AWS DeepRacer Student は、AWS リソースへのユーザーアクセスを制御するために IAM と直接統合しません。その代わりに、AWS DeepRacer Student では、AWS DeepRacer が管理する認証済のプロキシ API を使用して、ユーザーリソースが保護されます。

## AWS DeepRacer Student のコンプライアンス検証

AWS DeepRacer Student のセキュリティとコンプライアンスは、AWS のさまざまなコンプライアンスプログラムの一環として、第三者の監査機関によって評価されます。

特定のコンプライアンスプログラムの対象となる AWS サービスのリストについては、「[コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)」を参照してください。一般的な情報については、「[AWS コンプライアンスプログラム](#)」を参照してください。

サードパーティーの監査レポートをダウンロードするには、AWS Artifact を使用します。詳細については、[AWS Artifact におけるレポートのダウンロード](#) を参照してください。

AWS では、コンプライアンスに役立つ、次のリソースを提供しています。



- [セキュリティおよびコンプライアンスのクイックスタートガイド](#) – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境を AWS でデプロイするための手順を説明します。
- [AWSコンプライアンスのリソース](#) – このワークブックおよびガイドのコレクションは、お客様の業界と拠点に適用されるものである場合があります。
- AWS Configデベロッパーガイドの[ルールでのリソースの評価](#) – AWS Configは、リソース設定が、社内のプラクティス、業界のガイドラインそして規制にどの程度適合しているのかを評価します。
- [AWS Security Hub](#) – AWSのこのサービスは、AWS内でのユーザーのセキュリティ状態に関する包括的な見解を提供し、業界のセキュリティ標準、およびベストプラクティスに対するコンプライアンスを確認するために役立ちます。

## AWS DeepRacer Student の耐障害性

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティゾーンを中心として構築されます。リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立および隔離されたアベイラビリティゾーンがあります。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティゾーンの詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

## AWS DeepRacer Student のインフラストラクチャセキュリティ

マネージドサービスである AWS DeepRacer Student は、ホワイトペーパー「[Amazon Web Services のセキュリティプロセスの概要](#)」に記載されている AWS グローバルネットワークセキュリティの手順で保護されています。

# AWS DeepRacer 学生に関する一般的な問題のトラブルシューティング

## トピック

- [AWS DeepRacer Student アカウントから自動的にサインアウトされたのはなぜですか？](#)
- [AWS AI & ML 奨学金プログラムからのオプトアウト方法を教えてください。](#)
- [AWS DeepRacer Student アカウントを削除できない](#)
- [ドロップダウンリストに学校名が見つかりません。](#)
- [モデルのトレーニングが継続できません。](#)
- [「An account is registered with this email \(この E メールで登録済みのアカウントがあります\)」というエラーメッセージが表示されます](#)
- [Gmail アカウントでサインアップしましたが認証コードが見つかりません](#)

## AWS DeepRacer Student アカウントから自動的にサインアウトされたのはなぜですか？

AWS セキュリティポリシーに従って、30 日後に AWS DeepRacer Student アカウントから自動的にサインアウトされます。

- サービスを引き続き使用するには、[AWS DeepRacer 「学生サインイン」ページ](#)に移動し、認証情報を使用してサインインし直します。

## AWS AI & ML 奨学金プログラムからのオプトアウト方法を教えてください。

AWS AI & ML 奨学金プログラムはオプションのサービスであり、この技術の恩恵を十分に受けていない、あるいは少数派に属している、16 歳以上の学生を対象としています。AWS DeepRacer Student にサインアップすると、デフォルトでは AWS AI & ML 奨学金プログラムに登録されません。

参加するには、まず、「AWS AI & ML 奨学金プログラムの対象になるかどうか」セクションのチェックボックスをオンにしてオプトインする必要があります。これは、サイトの左側のナビゲー

シヨンペインからアクセスできるプロフィールページから AWS DeepRacer Student 以降にサインアップするときに行います。

- プログラムにオプトインした場合のみ、応募プロセスにアクセスできるようになります。この場合でも、応募は必須ではありません。

## AWS DeepRacer Student アカウントを削除できない

AWS DeepRacer Student アカウントを削除できない場合は、AWS DeepRacer マルチユーザーイベントを作成したかどうかを確認します。AWS プレイヤーアカウントは、AWS DeepRacer マルチユーザーと AWS DeepRacer Student AWS 用に によって作成されたマネージド ID ソリューションです。AWS Player アカウントは、これらの各 AWS サービスで作成されたすべてのリソースを保持します。

- 作成したイベントの参加者が壊れたエクスペリエンスのままにならないようにするには、AWS DeepRacer マルチユーザーイベントのリソースが含まれている AWS DeepRacer Student アカウントを削除することはできません。

## ドロップダウンリストに学校名が見つかりません。

特に高校は、ドロップダウンリストにすべての学校が表示されない場合があります。

- ご自身の学校がドロップダウンリストにない場合は、[その他] を選択し、学校名を入力します。

## モデルのトレーニングが継続できません。

モデルトレーニング時間が毎月の上限を超えている可能性があります。

- [ホームページ] を開き、[モデルトレーニングの残り時間] セクションにある、[消費したトレーニング時間数] を確認します。モデルトレーニング時間が上限を超過している場合は、時間がリセットされるまで待つてトレーニングを再開します。

## 「An account is registered with this email (この E メールで登録済みのアカウントがあります)」というエラーメッセージが表示されます

このエラーメッセージは、AWS プレイヤーアカウントのサインアップページに確認コードを入力し、AWS DeepRacer マルチユーザー経由で AWS プレイヤーアカウントにサインアップするために同じ E メールアドレスを既に使用している場合に表示されます。このエラーは、以前に同じ E メールアドレスを使用して Student AWS DeepRacer アカウントにサインアップした場合にも表示されません。

- 以前に作成した認証情報を使用して [AWS DeepRacer Student のサインインページ](#) にサインインするか、パスワードフィールドでパスワードを忘れた場合を選択します。

## Gmail アカウントでサインアップしましたが認証コードが見つかりません

Gmail アカウントを使用して AWS Player アカウントにサインアップしていて、検証コードメッセージを見つけられない場合は、間違ったフォルダに配信されている可能性があります。

- Gmail アカウントにサインインし、[プロモーション] フォルダで、「Your AWS Player profile verification code (AWS Player プロファイル認証コード)」というタイトルのメッセージを探します。

# AWS DeepRacer Student のクォータ

AWS DeepRacer Student に参加するすべての学生には、毎月10時間のモデルトレーニング用コンピューティングリソースと 5 GB のストレージが無料で提供されます。

# AWS DeepRacer Student アカウントの削除

AWS DeepRacer Student ポータルは、以下の情報を AWS Player アカウントに保存します。

- [E メールアドレス]
- パスワード
- レーサーの名前
- Student League でのリーダーボードランク

収集されたデータの詳細については、「学生ユーザーガイド[AWS DeepRacer Student でのデータ保護](#)」の「」を参照してください。AWS DeepRacer

AWSのサーバーからこの情報を削除する場合は、次の手順を使用して AWS DeepRacer Student ポータルアカウントを削除します。AWS DeepRacer Student アカウントを削除すると、AWS Player アカウントと関連するすべてのリソースも削除されます。

AWS Player アカウントの詳細については、「」を参照してください。 [AWS Player アカウントとは](#)

## Note

AWS DeepRacer マルチユーザーイベントを作成した場合は、AWS プレイヤーアカウントを削除することはできません。詳細については、「学生ユーザーガイド[AWS DeepRacer Student アカウントを削除できない](#)」の「」を参照してください。AWS DeepRacer

AWS DeepRacer Student プレイヤーアカウントを削除するには

## Important

AWS DeepRacer Student アカウントの削除は、元に戻すことができないアクションです。AWS DeepRacer Student を削除すると、AWS Player アカウントと関連するすべてのリソースも削除されます。

AWS DeepRacer Student アカウントを削除すると、Player アカウントのリソースは 1 AWS 年以内にサーバーから削除されます。

1. Student AWS DeepRacer ランディングページを開きます: <https://student.deepracer.com/signIn>。
2. プロンプトが表示されたら、Student AWS DeepRacer アカウントにログインします。
3. [Your account] (アカウント) を選択します。
4. [Your account] (アカウント) ページで、[Delete your account] (アカウントの削除) を選択します。
5. [To confirm deletion, type Delete in the field] (削除を確認するには、フィールドに Delete と入力します) に、**Delete** と入力します。
6. [削除] を選択します。

アカウントが正常に削除されると、アカウントが正常に削除されたというメッセージが表示され、AWS DeepRacer 学生ログインページに戻ります。

アカウントも削除する場合は、AWS 「アカウント [を閉鎖する AWS](#)」で説明されているステップを使用します。

お客様がプライバシーとデータセキュリティについて懸念されるのは当然のことです。不正なアクセスやカスタマーコンテンツの開示を防ぐように設計された、高度な技術的および物理的な制御を行っています。お客様の信頼を維持することは継続的なコミットメントです。AWS データプライバシーのコミットメントの詳細については、[「プライバシー通知」](#) ページを参照してください。

# AWS DeepRacer Student ユーザーガイドのドキュメント履歴

次の表は、AWS DeepRacer Student のドキュメントリリースの一覧です。

変更	説明	日付
<a href="#">2024 年プログラムの最新情報</a>	2024 年 AWS AI & 機械学習スカラシッププログラムと AWS DeepRacer Student リーグに関するトピックを更新しました。AWS AI & 機械学習スカラシップに申請するすべての学生が利用できる新しいコースに関する情報を追加しました。	2023 年 11 月 1 日
<a href="#">2023 AWS DeepRacer 学生リーグの更新</a>	2023 AWS DeepRacer 学生リーグに関する複数のトピックを更新しました。詳細については、「 <a href="#">用語と条件</a> 」を参照してください。	2023 年 3 月 1 日
<a href="#">初回リリース</a>	AWS AI & ML Scholarship プログラムと、AWS DeepRacer Student League のサポートを含む、AWS DeepRacer Student ユーザーガイドの初回リリース。	2021 年 12 月 1 日



翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。