Projets

Astro Pi: Mission Zero

Prépare-toi pour la Mission Zero

Étape 1Introduction

Égaye la routine quotidienne des astronautes de la station spatiale internationale en leur affichant un message et l'humidité relative à bord, à l'aide du Sense HAT de l'ordinateur Astro Pi.

Tu utiliseras l'émulateur en ligne Sense HAT pour créer ton programme, aucun matériel supplémentaire n'est nécessaire - tout se fait dans un navigateur Web.

Ton programme achevé sera exécuté dans l'espace dans la Station spatiale internationale (ISS) ! Tu recevras aussi un certificat spécial indiquant exactement où se trouvait l'ISS lors de l'exécution de ton programme !

Ce que tu feras

Voici un exemple du type de programme que tu peux créer. Clique sur **Run** (Exécuter) pour le voir en action.

Ce que tu vas apprendre

Tu découvriras l'unité Astro Pi et apprendras à l'utiliser et à faire les actions suivantes :

- afficher des messages et ajouter des couleurs
- créer des images
- mesurer l'humidité à l'aide d'un Sense HAT

Ce projet couvre des éléments des sections suivantes du **Raspberry Pi Digital Making Curriculum** (http://rpf.io/curriculum):

- **Utilise des concepts de base de programmation pour créer des programmes simples** (https://curriculum.raspberrypi.org/programming/creator/)

Ce qu'il te faut

Matériel informatique

- Un ordinateur avec une connexion Internet

Logiciels

- Un navigateur Web (par exemple Google Chrome) pour ouvrir **https://trinket.io/mission-zero (https://trinket.io/mission-zero)**

Notes pour les enseignants et les mentors

Cette activité peut être réalisée en une après-midi. Organisez vos élèves en équipes de quatre personnes et laissez-nous les guider dans l'écriture d'un petit programme en Python qui affiche un message personnel et l'humidité relative sur l'Astro Pi.

Lisez le **document des directives officielles (https://astro-pi.org/wp-content/uploads/2018/09/Astro_Pi_Mission_Zero_Guidelines_2018_19_V12_pages.pdf)** pour la Mission Zero.

Vous devrez vous inscrire au défi Mission Zero pour permettre à votre ou vos équipes de participer.

- Allez à la page **Emulateur Trinket Mission Zero (https://trinket.io/mission-zero/register)**.
- Remplissez le formulaire et cliquez sur **Submit (Soumettre)***.

\ * Veuillez noter que ce formulaire d'inscription est disponible en anglais uniquement.

Les champs du formulaire incluent :
Nom de l'enseignant/mentor
Nom d'équipe
Nombre de membres de l'équipe
Noms et âges des membres de l'équipe
Dans quelle langue avez-vous accédé aux directives de Mission Zero ?

- Un compte Trinket sera créé pour vous (si vous n'en avez pas déjà un, ou si vous n'êtes pas connecté). Vous pouvez créer un compte par adresse e-mail. Chaque compte a son propre **code de salle de classe**, et vous devez donner ce code à votre ou vos équipes quand elles sont prêtes à soumettre leurs programmes.
- Affichez votre code de salle de classe quelque part où il sera visible, par exemple sur un tableau blanc ou en utilisant un projecteur, et commencez l'activité.

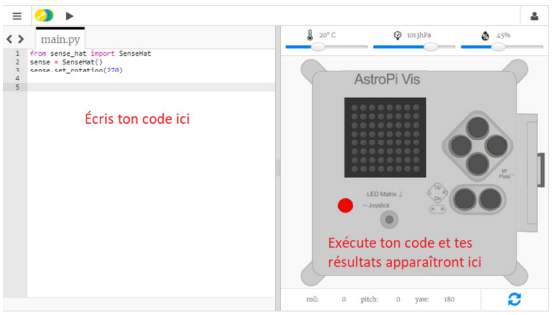
  Nous avons créé **un document imprimable de deux pages (https://astro-pi.org/astro_pi_mission_zero_project_print_out_v10_print/)** qui couvre les points clés de la Mission Zero et que les étudiants et les jeunes peuvent utiliser avec ce projet en ligne.

Étape 2 Qu’est-ce qu’un Astro Pi ?

Un Astro Pi est un ordinateur Raspberry Pi encapsulé dans un boîtier spécialement conçu pour les conditions spatiales. Il possède également une carte supplémentaire appelée Sense HAT, spécialement conçue pour la mission Astro Pi. Le Sense HAT est doté d’un joystick, d’un écran LED et de capteurs pour enregistrer la température, l’humidité, la pression et l’orientation.

Voici une vraie unité Astro Pi dans la station spatiale internationale, exécutant un code écrit par des élèves. C’est là que ton code sera finalement exécuté !

Pour cette mission, tu utiliseras l’émulateur Sense HAT. L’émulateur est un logiciel qui simule toutes les fonctions de l’Astro Pi dans ton navigateur Web.



Il existe quelques différences entre le Sense HAT réel et celui qui est émulé :

- Sur l’émulateur, tu peux définir toi-même la température, la pression et l’humidité à l’aide de curseurs, tandis que le véritable Sense HAT de l’Astro Pi utilise des capteurs pour mesurer les valeurs de ces paramètres dans leur environnement.

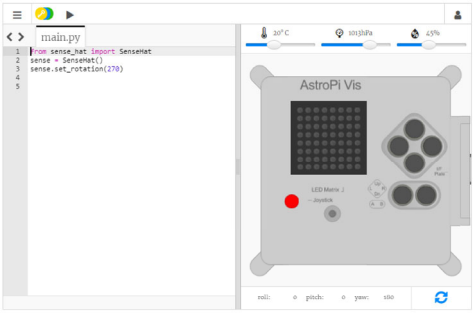
- Tu peux utiliser utiliser la souris pour cliquer et faire glisser le Sense HAT émulé pour le déplacer et le faire pivoter, afin de simuler des changements d’orientation ; le véritable Astro Pi (et son Sense HAT) peut se déplacer dans le monde réel, et les capteurs d’orientation du Sense HAT détectent quand et comment il s’est déplacé.

Étape 3 Afficher un message

Ouvre **l'émulateur Sense HAT** [(https://trinket.io/mission-zero)](https://trinket.io/mission-zero) pour le projet Mission Zero.

Tu vas constater que trois lignes de code ont été ajoutées automatiquement:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.set_rotation(270)
```



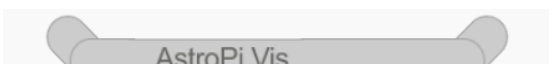
Ce code établit la connexion à l'Astro Pi et assure que l'affichage LED de l'Astro Pi s'effectue dans le bon sens. Laisse ce code ici car tu en auras besoin.

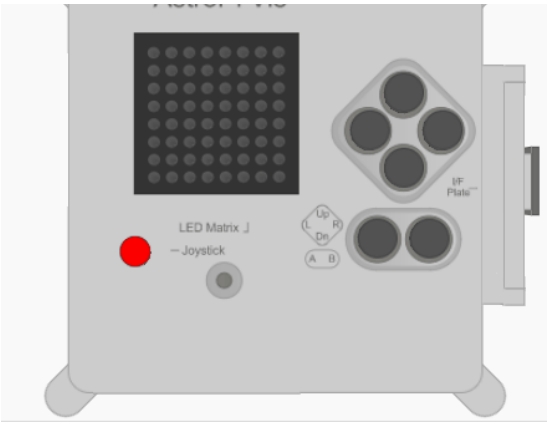
Peut-être tu pourrais laisser un message de salutation amical pour les astronautes de la station spatiale internationale (ISS) qui travaillent près de l'Astro Pi ? On va faire défiler un message sur l'écran.

Ajoute cette ligne en-dessous de l'autre ligne de code :

```
sense.show_message("Astro Pi")
```

Appuie sur le bouton **Run** (Exécuter) et regarde comme le message `Astro Pi` défile sur l'écran LED.





Pour afficher un autre message tu peux écrire ce que tu veux entre les guillemets ("").

Tu peux aussi modifier la vitesse de défilement du message sur l'écran. Ajoute un `scroll_speed` (vitesse de défilement) à ta ligne de code, comme cela :

```
sense.show_message("Astro Pi", scroll_speed=0.05)
```

La vitesse par défaut du message est `0.1` . En réduisant le nombre, tu fais défiler le message plus rapidement et en augmentant le nombre tu fais défiler le message plus lentement.

Étape 4 Ajouter de la couleur

Les LED de l'Astro Pi peuvent également afficher des couleurs. Tu peux spécifier une couleur en créant une variable et en lui attribuant une valeur de couleur RGB.

Tu peux apprendre ici comment créer toutes les couleurs en utilisant différentes proportions de rouge, de vert et de bleu :

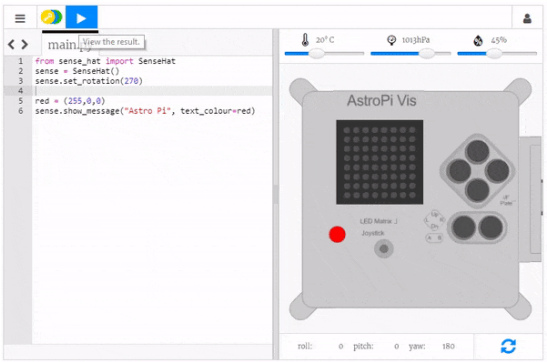
Choisis une couleur et découvre la valeur RGB de cette couleur. Tu peux utiliser un **sélecteur de couleur** [(https://www.w3schools.com/colors/colors_rgb.asp)](https://www.w3schools.com/colors/colors_rgb.asp) pour t'aider.

Crée une variable qui servira à stocker la couleur que tu as choisi. Par exemple, si tu as choisi la couleur rouge, tu vas écrire cette ligne de code :

```
red = (255,0,0)
```

Tu peux maintenant afficher ton texte dans la couleur de ton choix ! Pour indiquer au programme d'utiliser la couleur que tu as créée, ajoute le paramètre `text_colour` (couleur du texte) à la ligne de code qui affiche ton texte :

```
red = (255,0,0)
sense.show_message("Astro Pi", text_colour=red)
```



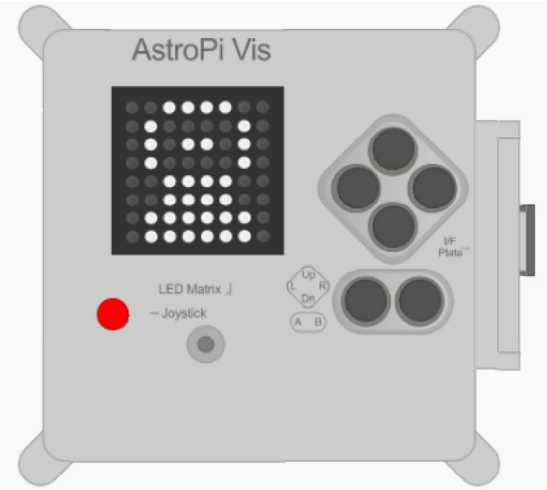
Tu peux également modifier la couleur de fond de l'écran. Choisis une autre couleur et crée une autre variable pour cette couleur. Pour indiquer au programme d'utiliser la couleur de fond que tu as choisie, ajoute le paramètre `back_colour` (couleur de fond) à ton code :

```
red = (255,0,0)
green = (0,255,0)
sense.show_message("Astro Pi", text_colour=red, back_colour=green)
```

Modifie le texte et la couleur du message de salutation - quel message veux-tu envoyer aux astronautes à bord de l'ISS ?

Étape 5 Afficher une image

Tu peux afficher des images sur la matrice LED de l'Astro Pi. Ton message de salutation pour les astronautes pourrait peut-être inclure une image ou un motif avec le texte, ou à la place du texte ?



En bas de ton programme, crée des variables pour les couleurs que tu veux définir pour dessiner une image. Tu peux utiliser autant de couleurs que tu veux mais dans cet exemple, nous nous limiterons à deux couleurs - blanc (`w`) et noir (`b`).

```
w = (255, 255, 255)
b = (0, 0, 0)
```

Remarque : Cette fois-ci, il est conseillé de donner aux variables définies pour les couleurs des noms se limitant à une lettre, car cela permettra de gagner du temps à l'étape suivante, quand tu les saisiras de nombreuses fois. De plus, en utilisant des noms à une seule lettre tu pourras voir plus facilement l'image que tu vas dessiner.

Sous tes nouvelles variables, crée une liste de 64 éléments. Chaque élément représente un pixel de la matrice du LED et correspond à l'une des variables de couleur que tu as définies. Dessine ton image en mettant une variable à l'endroit où tu veux que la couleur de cette variable apparaisse. Nous avons dessiné un astronaute en utilisant les pixels noirs (`b`) pour l'arrière-plan et les pixels blancs (`w`) pour dessiner la combinaison spatiale de l'astronaute :

```
picture = [
    b, b, w, w, w, w, b, b,
    b, w, b, b, b, b, w, b,
    b, w, b, w, w, b, w, b,
    b, w, b, b, b, b, w, b,
    b, b, w, w, w, w, b, b,
    b, b, w, w, w, w, b, b,
    b, w, w, w, w, w, w, b,
    b, w, w, w, w, w, w, b
]
```

Ajoute une ligne de code pour afficher ton image sur l'écran LED.

```
sense.set_pixels(picture)
```

Appuie sur **Run** (Exécuter) pour afficher ton image.

Tu peux ajouter du code pour inclure une courte attente (ou `sleep`) après que l'image soit affichée. Cela donnera aux astronautes le temps de voir ton image avant l'affichage de la partie suivante de ton message. En haut de ton programme, rajoute :

```
from time import sleep
```

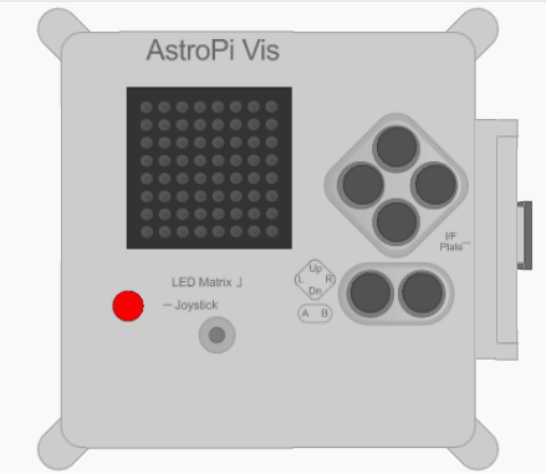
Ensuite, sur la ligne qui suit celle qui affiche ton image, ajoute ce code pour créer un moment d'attente de deux secondes :

```
sleep(2)
```

Crée ta propre image ou ton propre motif que tu veux afficher pour les astronautes !

Étape 6 Mesurer l'humidité

Le capteur d'humidité de l'Astro Pi peut mesurer l'humidité de l'air qui l'entoure, c'est une fonction utile pour t'aider à collecter des données sur les conditions dans l'espace.



L'Astro Pi mesure l'humidité dans l'ISS en pourcentage de concentration d'eau dans l'air.

Une partie de ta mission est de contribuer à la vie quotidienne de l'équipage à bord de l'ISS, afin de leur faire savoir que l'humidité à bord de la station spatiale est dans les limites normales, cela les rassurera.

Ajoute ce code pour mesurer l'humidité :

```
humid = sense.humidity
```

Cette ligne mesure l'humidité actuelle et stocke la valeur mesurée dans la variable `humid`.

L'humidité est enregistrée de manière très précise, c'est-à-dire que la valeur stockée aura un grand nombre de décimales. Tu peux arrondir la valeur à n'importe quel nombre de décimales. Dans l'exemple, nous avons arrondi à une décimale, mais pour avoir un autre niveau de précision remplace le nombre `1` par le nombre de décimales que tu souhaites.

```
humid = round( sense.humidity, 1 )
```

Pour afficher l'humidité actuelle sous la forme d'un message défilant à l'écran, ajoute cette ligne de code :

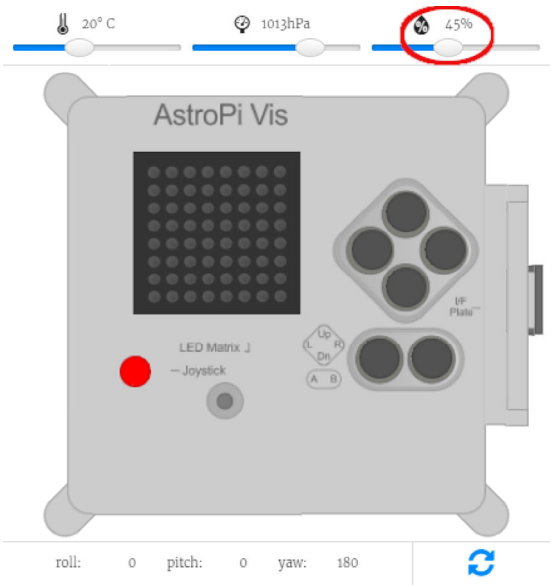
```
sense.show_message( str(humid) )
```

La partie `str()` convertit l'humidité d'un nombre en texte afin que l'Astro Pi puisse l'afficher.

Tu peux également afficher l'humidité dans un autre message en joignant les parties de ton message avec un +.

```
sense.show_message( "C'est " + str(humid) + " %" )
```

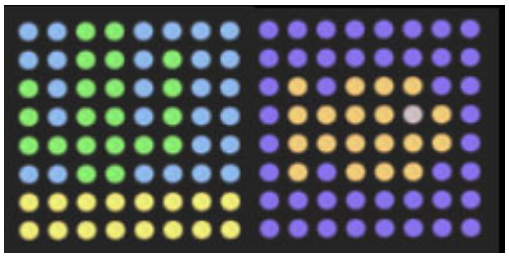
Le vrai Astro Pi mesurera l'humidité autour de lui, mais tu peux déplacer le curseur d'humidité sur l'émulateur Sense HAT pour simuler les changements d'humidité et tester ton code.



Remarque : Tu te demandes peut-être pourquoi le curseur d'humidité affiche l'humidité sous forme d'un nombre entier, mais le résultat de la mesure que tu obtiens est en décimales. L'émulateur simule la légère imprécision du capteur réel, de sorte que le résultat de la mesure de l'humidité que tu vois peut être très légèrement supérieure ou inférieure à la valeur que tu as définie avec le curseur.

Étape 7Afficher l'humidité

Tu peux combiner ta mesure d'humidité avec une image pour aussi indiquer l'humidité d'une manière graphique. Par exemple, tu peux afficher un océan pour une humidité élevée, et un désert pour une faible humidité :



En bas de ton programme, crée d'autres variables pour les couleurs que tu veux utiliser pour dessiner tes images. Tu en as peut-être déjà défini certaines lors d'une étape précédente.

```
o=(255,130,0)
b=(0,0,255)
c=(0,150,255)
e=(80,80,80)
g=(0,255,0)
y=(255,255,0)
```

Comme précédemment, dessine tes images en créant d'abord une liste pour chacune d'entre elles, puis en indiquant la couleur que tu veux donner aux pixels de chaque élément de la liste.

```
wet = [
  b, b, b, b, b, b, b, b,
  b, b, b, b, b, b, b, b,
  b, o, b, o, o, o, b, b,
  b, o, o, o, o, e, o, b,
  b, o, o, o, o, o, o, b,
  b, o, b, o, o, o, b, b,
  b, b, b, b, b, b, b, b,
  b, b, b, b, b, b, b, b
]


dry = [
  c, c, g, g, c, c, c, c,
  c, c, g, g, c, g, c, c,
  g, c, g, g, c, g, c, c,
  g, c, g, g, c, g, c, c,
  g, g, g, g, g, g, c, c,
  c, c, g, g, c, c, c, c,
  y, y, y, y, y, y, y, y,
  y, y, y, y, y, y, y, y
]
```

Ajoute du code pour obtenir l'humidité :

```
humid = sense.humidity
```

Décide maintenant quelle image tu veux afficher. Pour cet exemple, nous afficherons l'image **`humide`** si la lecture de l'humidité est de 40% ou plus, et l'image `sec` si l'humidité est inférieure à 40%.

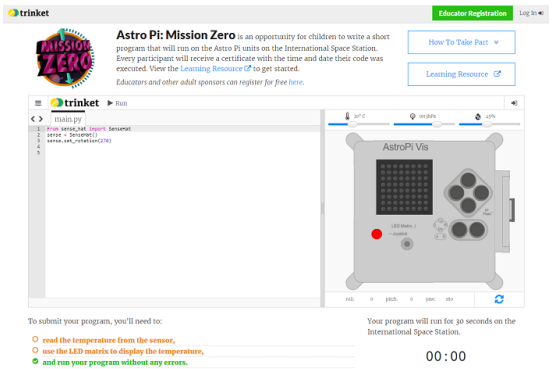
```
humid = sense.humidity
if humid >= 40:
    sense.set_pixels(wet)
else:
    sense.set_pixels(dry)
```

Utilise le curseur d'humidité pour définir une humidité sur l'émulateur. Exécute ton programme et vérifie que l'image que tu as choisie pour cette humidité est correctement affichée.

Modifie ton code pour que ton programme affiche l'humidité pour les astronautes de la manière que tu as choisie.

Étape 8 Soumettre ton programme

Ton code doit respecter quelques règles pour que tu puisses le soumettre et le faire exécuter dans la station spatiale internationale. Si ton code les respectent, les règles en bas de l'émulateur s'allument en vert lorsque tu exécutes le programme.



1. Mesure l'humidité.
2. Allume les LED
3. Vérifie que ton code s'exécute jusqu'au bout sans erreur. Tu ne dois pas inclure de boucles `while True` dans ton code car cela fera exécuter le code en continu et l'empêchera de se terminer.

Assure-toi également que les critères suivants sont satisfaits :

1. Assure-toi que ton message pour les astronautes ne dure pas plus de 30 secondes, car c'est la durée d'exécution de ton code dans l'ISS
2. Évite d'utiliser des méthodes nécessitant une saisie de donnée
3. Pour les importations n'utilisent que des données des modules `sense_hat`, `time`, et `random`
4. Assure-toi de ne pas inclure de juron

Une fois que toutes les règles sont passées au vert, tu es prêt à faire ta soumission.

Saisis le code de ta classe dans la case en bas - ton enseignant ou ton mentor te dira quel est ton code.

Les **Notes pour les enseignants et les mentors** figurent dans l'étape **Introduction** (https://projects.raspberrypi.org/fr-FR/projects/astro-pi-mission-zero/1).

Le nom de ton enseignant va s'afficher. Si c'est le bon nom, clique sur le bouton vert **Continue to form** (Continuer vers le formulaire).

Donne le nom de ton équipe et les noms des membres de l'équipe. Ces noms seront imprimés sur le certificat une fois que ton code est exécuté dans l'espace, alors assure-toi de les épeler correctement !

Appuie sur le bouton **Submit** (Soumettre) pour donner ton code. Ton enseignant ou ton mentor recevra un e-mail de confirmation pour ton inscription.

Si tu le souhaites, tu peux partager le lien vers ton code sur les réseaux sociaux pour dire aux gens que le code que tu as écrit sera exécuté dans l'espace !

Étape 9 Défi : d'autres projets Astro Pi

Maintenant que tu as terminé ta mission, pourquoi ne pas essayer d'autres projets en utilisant les autres capteurs de l'Astro Pi ?

Si tu te sens confiant, tu pourrais prendre part à **Mission Space Lab** (https://astro-pi.org/missions/space-lab/) ! Forme une équipe de deux à six personnes pour travailler ensemble comme de vrais scientifiques de l'espace et concevoir ta propre expérience. Reçois gratuitement du matériel informatique pour ta mission et écris le code Python pour mener à bien ton expérience. Si tu atteins le statut de vol, ton code sera envoyé vers la station spatiale internationale et fonctionnera sur Astro Pi pendant trois heures (deux orbites). Toutes les données que ton code collecte dans l'espace - images ou fichiers des données du capteur - seront téléchargées et te seront renvoyées pour analyse.

Ou tu peux aussi essayer l'un des autres projets Astro Pi :

- En apprendre **plus sur le Sense HAT** (https://projects.raspberrypi.org/fr-FR/projects/getting-started-with-the-sense-hat) et les autres choses qu'il peut faire
- Créer de jolies **scintillements aléatoires** (https://projects.raspberrypi.org/fr-FR/projects/sense-hat-random-sparkles) sur l'écran LED du Sense HAT
- Créer le jeu **Flappy Astronaute** (https://projects.raspberrypi.org/fr-FR/projects/flappy-astronaut)
- Défie tes amis avec le jeu **Labyrinthe de marbre** (https://projects.raspberrypi.org/fr-FR/projects/sense-hat-marble-maze)
- Recrée le jeu classique **Pong** (https://projects.raspberrypi.org/fr-FR/projects/sense-hat-pong)

Ce projet a été traduit par des bénévoles:

Jonathan Vannieuwkerke

Michel Arnols

Grâce aux bénévoles, nous pouvons donner aux gens du monde entier la chance d'apprendre dans leur propre langue. Vous pouvez nous aider à atteindre plus de personnes en vous portant volontaire pour la traduction - plus d'informations sur **rpf.io/translate** (https://rpf.io/translate).
