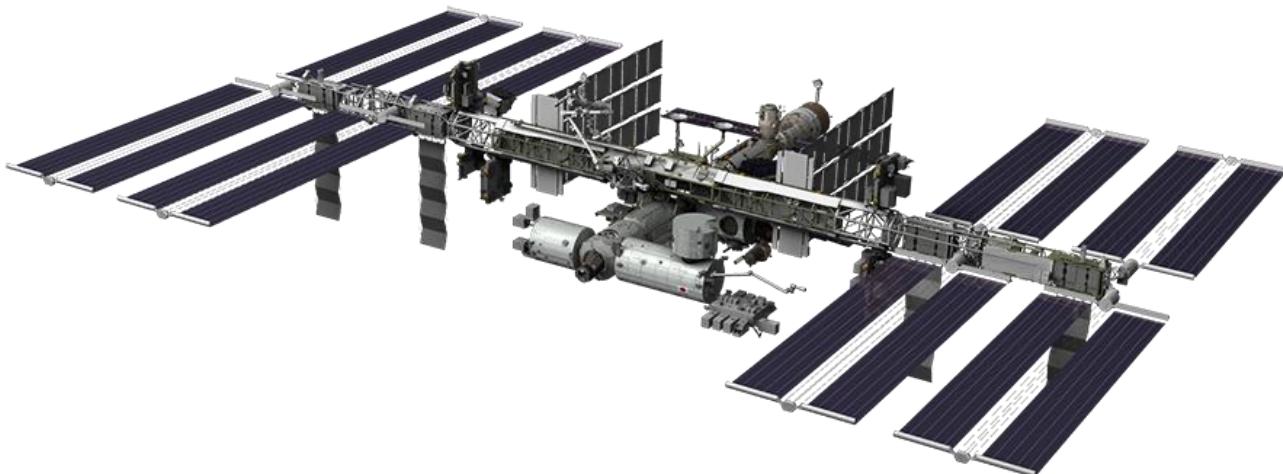




# Où est l'ISS ?

Après s'être interrogé sur qui est dans la station ISS, vous allez maintenant déterminer où se situe la station ?  
Suite de l'activité « Qui est dans la station ISS ? » (Thème 4 - Activité 2).



Vous réaliserez le programme à l'aide du logiciel « Pyzo ».

La Station Spatiale Internationale orbite autour de la Terre. Elle fait le tour de la Terre environ toutes les 90 minutes. L'ISS se déplace à une vitesse moyenne de 7,66 km par seconde. C'est du rapide !

Utilisons un autre webservice afin de connaître la position de la Station Spatiale Internationale.

- Tout d'abord il faut que tu ouvres l'URL pour le webservice dans un autre onglet dans ton navigateur :  
<http://api.open-notify.org/iss-now.json> (CTRL + CLIC)

La page affichée devrait ressembler cela :

Au format JSON :

```
iss_position:  
    latitude: "8.54938193505081"  
    longitude: "73.16560793639105"  
timestamp: 1461931913  
message: "success"
```

En données brutes :

```
{"iss_position": {"latitude": "3.7211", "longitude": "59.7313", "timestamp": 1616838076}, "message": "success"}
```



## LOCALISATION, CARTOGRAPHIE ET MOBILITÉ

Le résultat contient les coordonnées du point sur la Terre au-dessus duquel l'ISS se trouve actuellement.

- *La longitude est la position Est-Ouest et elle passe entre -180 et 180.  
0 est le premier méridien qui traverse Greenwich à Londres au Royaume-Uni.*
- *La latitude est la position Nord-Sud et elle passe entre 90 et -90.  
0 est l'Équateur terrestre.*

Réalise un nouveau programme Python à l'aide Pyzo (Autres raccourcis/SNT/Pyzo\_SNT). Sauvegarde ton programme dans le dossier « [Documents/SNT/THEME\\_5](#) » avec un nom de la forme « [ISS-MonNomDeFamille.py](#) ».

En début de programme, tu vas insérer les bibliothèques nécessaires pour la suite du programme

```
1 import urllib.request  
2 import json  
3 import time  
4 from turtle import *
```

Maintenant il faut appeler le webservice à travers Python ; ajoute le code suivant à la fin de ton script pour récupérer la position actuelle de l'ISS :

```
6 url = "http://api.open-notify.org/iss-now.json"  
7 reponse = urllib.request.urlopen(url)  
8 resultat = json.loads(reponse.read())  
9  
10 print(resultat)
```

Si tu exécutes ton script, tu devrais obtenir cette réponse dans le « Shell » :

```
{'iss_position': {'latitude': '2.2136', 'longitude': '-127.8171'}, 'timestamp': 1  
652963389, 'message': 'success'}
```

Les valeurs seront certainement différentes car le temps passe et l'ISS se déplace 😊

La valeur renvoyée par le webservice est un dictionnaire qui comporte entre les { et } des données sous la forme ‘clé’ : ‘valeur’.

Pour la suite, la ligne : `print(resultat)` n'est plus utile ; tu peux insérer un # au début de la ligne afin de la transformer en commentaire.

Maintenant, tu vas créer des variables pour stocker la latitude et la longitude, puis les afficher :

```
12 Position_ISS = resultat['iss_position']  
13 latitude_ISS = Position_ISS['latitude']  
14 longitude_ISS = Position_ISS['longitude']  
15 print(f"ISS latitude : {latitude_ISS}")  
16 print(f"ISS longitude : {longitude_ISS}")
```

Après exécution du script, voici ce que tu devrais obtenir dans le « Shell » :

```
ISS latitude : 50.9406  
ISS longitude : -177.1406 avec des valeurs différentes, bien sûr !
```

**Il est temps de réaliser une petite sauvegarde ton programme 😊**



## LOCALISATION, CARTOGRAPHIE ET MOBILITÉ

Ça serait plus parlant d'afficher la position de l'ISS sur une carte. Pour cela, tu vas utiliser la bibliothèque graphique ‘turtle’ et un fichier image « [Earthmap1000x500.gif](#) » comme fond d'écran. Fais bien attention à ce que cette image soit enregistrée dans le même dossier que ton programme : « [Documents/SNT/THEME\\_5](#) » ! Il faut la télécharger depuis le site.

La taille de l'écran doit être adaptée à la taille d'image qui est 1000 pixels par 500 pixels grâce à la ligne :

```
screen.setup(1000, 500)
```

Tu veux pouvoir envoyer la ‘turtle’ à une latitude et longitude donnée. Afin de faciliter cette tâche, tu vas paramétrer l'écran afin qu'il corresponde aux coordonnées utilisées :

```
18 screen = Screen()
19 screen.setup(1000,500)
20 screen.setworldcoordinates(-180,-85,180,85)
21 screen.bgpic ("Earthmap1000x500.gif")
22
23 # a laisser pour fermer la fenêtre
24 exitonclick()
```

À saisir à la fin du programme pour que la carte se ferme lorsque que l'on clique dessus.

On crée ainsi un rapport de proportionnalité entre la longitude  $-180^{\circ}$  +  $180^{\circ}$  et les bords gauche et droite de la carte ainsi qu'entre la latitude  $-90^{\circ}$  et  $+90^{\circ}$  et les bords bas et haut de la carte.

Cependant il reste deux problèmes :

1. La carte coupe une partie des pôles Nord et Sud  $-85^{\circ}$  et  $+85^{\circ}$
2. En prenant une échelle de  $1^{\circ}$ , on obtient une trop petite précision on alors divise l'échelle par 10.

Donc tu vas modifier la ligne 20 :

```
18 screen = Screen()
19 screen.setup(1000,500)
20 screen.setworldcoordinates(-1800,-850,1800,850)
21 screen.bgpic ("Earthmap1000x500.gif")
22
23
24 # a laisser pour fermer la fenêtre
25 exitonclick()
```

Maintenant les coordonnées correspondent à la latitude et longitude que tu as reçues du webservice.

Voici le code complet :

```
1 import urllib.request
2 import json
3 import time
4 from turtle import *
5
6 url = "http://api.open-notify.org/iss-now.json"
7 reponse = urllib.request.urlopen(url)
8 resultat = json.loads(reponse.read())
9
10 #print(resultat)
11
12 Position_ISS = resultat['iss_position']
13 latitude_ISS = Position_ISS['latitude']
14 longitude_ISS = Position_ISS['longitude']
15 print(f"ISS latitude : {latitude_ISS}")
16 print(f"ISS longitude : {longitude_ISS}")
17
18 screen = Screen()
19 screen.setup(1000,500)
20 screen.setworldcoordinates(-1800,-850,1800,850)
21 screen.bgpic ("Earthmap1000x500.gif")
22
23
24 # a laisser pour fermer la fenêtre
25 exitonclick()
```



À saisir à la fin du programme pour que la carte se ferme lorsque que l'on clique dessus.

Enregistre ton programme puis exécute ton script !

La carte doit, maintenant, être correctement centrée sur 0, 0.



## LOCALISATION, CARTOGRAPHIE ET MOBILITÉ

Créons une ‘turtle’ pour l’ISS.

```
18 screen = Screen()
19 screen.setup(1000,500)
20 screen.setworldcoordinates(-1800,-850,1800,850)
21 screen.bgpic ("Earthmap1000x500.gif")
22
23 screen.register_shape("iss.gif")
24 ISS =Turtle()
25 ISS.shape("iss.gif")
26 ISS.setheading(0)
27
28 # a laisser pour fermer la fenêtre
29 exitonclick()
```



Ton projet inclut les fichiers ‘iss.gif’, ‘iss2.gif’ et ‘iss3.gif’, essayes les deux autres images afin de choisir celle que tu préfères. Ces images doivent être téléchargées et enregistrées dans le dossier « [THEME\\_5](#) ». L’ISS est placée au centre de la carte, maintenant il faut la placer au bon endroit sur la carte :

```
23 screen.register_shape("iss2.gif")
24 ISS =Turtle()
25 ISS.shape("iss2.gif")
26 ISS.setheading(0)
27
28 ISS.penup()
29 lon_int = int(float(longitude_ISS)*10)
30 lat_int = int(float(latitude_ISS)*10)
31 print(f"Longitude entier : {lon_int} latitude entier : {lat_int}")
32 ISS.goto(lon_int,lat_int )
33
34 # a laisser pour fermer la fenêtre
35 exitonclick()
```

À noter que la latitude, d’habitude, est donnée en premier, mais tu as besoin de la longitude d’abord car on utilise un système de coordonnées (x,y) pour placer la ‘turtle’.

Teste ton programme en l’exécutant. L’ISS devrait bouger jusqu’à sa position au-dessus de la Terre.

Attends quelques minutes et exécute de nouveau ton programme pour voir jusqu’où l’ISS s’est déplacée.

