



Time to quake

Nos buts :

Notre projet a pour but de répertorier les séismes dans le monde en temps réel et depuis 2002

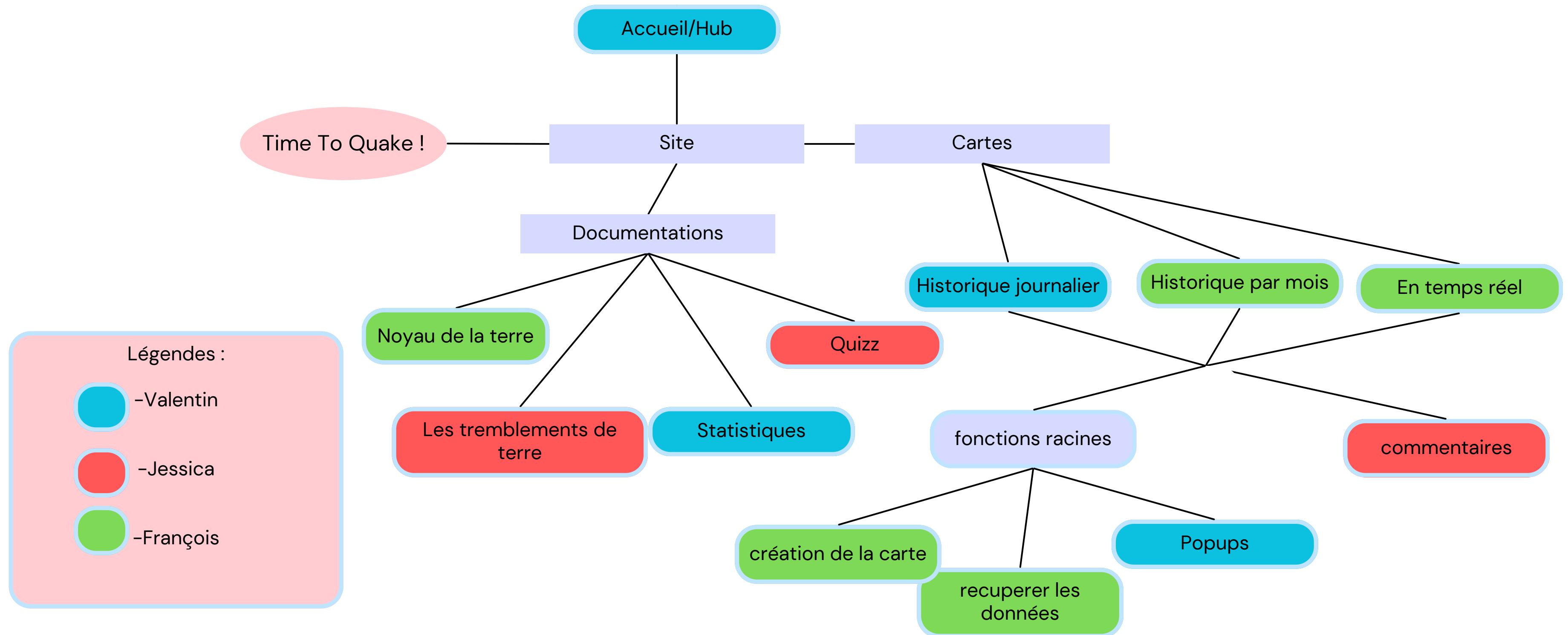
Motivation (pourquoi ce projet ?) : Informer les gens, pour qu'ils voient que la terre bouge constamment. De plus ce projet nous amuse.

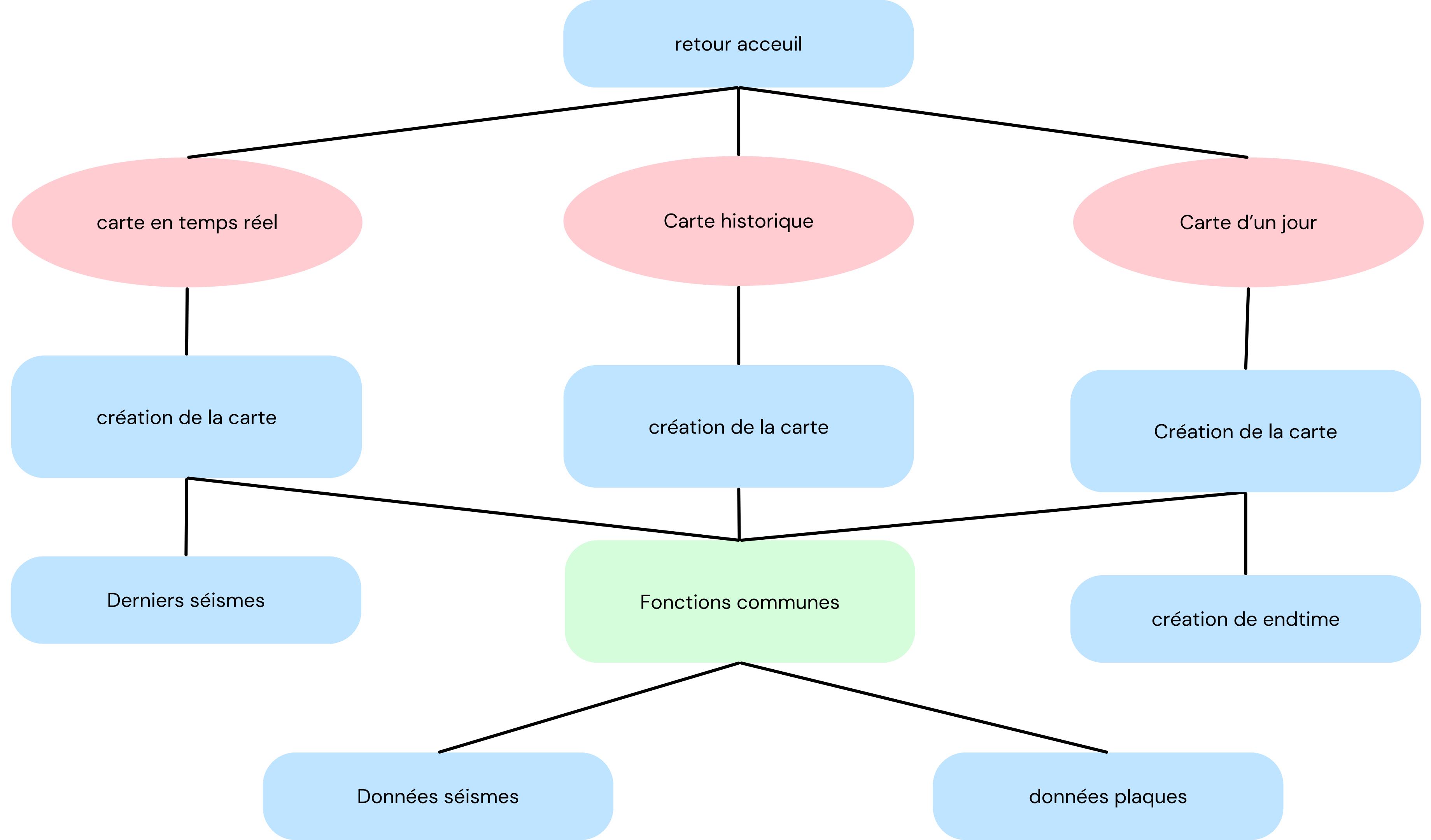
LES ÉTAPES DU PROJET :

- L'idée est née de l'envie de connaître mieux les séismes
- La première étape a consisté à trouver les données des séismes
- Ensuite à traiter les données et les ajouter sur une carte folium
- Enfin, nous avons fait de la documentation pour approfondir le sujet

Difficultés : trouver les données,
utiliser un nouveau module folium et ses plugins

Organisation du travail au sein du groupe :





A detailed satellite image of Earth, centered on the Northern Hemisphere and the Middle East. The image shows the blue oceans, white clouds, and various landmasses in shades of brown, green, and blue. The Middle East is clearly visible, along with parts of Europe, Africa, and Asia. The text 'Fonctions communes:' is overlaid on the left side of the globe.

Fonctions communes :

Donnée_plaques :

```
# -*- coding: utf-8 -*-
import requests

def donnees_plaques():

    url = "https://raw.githubusercontent.com/fraxen/tectonicplates/master/GeoJSON/PB2002_boundaries.json"#URL à interroger
    reponse = requests.get(url)#on utilise le module requests pour l'url et on récupère les données de plaques tectoniques

    if reponse.status_code == 200:

        data = reponse.json() #on utilise le fichier json
        return data
```

Donnée_séisme :

Version 1:

```
# -*- coding: utf-8 -*-
import requests

def donnees_seismes(starttime,endtime,minmagnitude):
    url = 'https://earthquake.usgs.gov/fdsnws/event/1/query'#URL à interroger

    parametres = {
        'format': 'geojson', #on demande les données en format geojson
        'starttime': starttime ,#on specifie la date et l'heure de début pour la recherche des séismes.
        'endtime':endtime,#on specifie la date et l'heure de la fin pour la recherche des séismes.
        'minmagnitude': minmagnitude, # on définit la magnitude minimale a integrer dans les resultat
        'orderby': 'time' #on trie les séismes par ordre chronologique
    }

    reponse = requests.get(url, params=parametres)

    if reponse.status_code == 200:

        data = reponse.json()#on convertit la réponse en format JSON
        return data
```

Donnée_séisme :

Version 2 :

```
def donnees_seisme():
    url = "https://www.seismicportal.eu/fdsnws/event/1/query?format=json&limit=500"

    reponse = requests.get(url)#recupere les données des seismes

    if reponse.status_code == 200:

        data = reponse.json()#on transforme la réponse en fichier json
        return data
```

Retour accueil :

```
def retour_accueil(): #fonction pour afficher un bouton sur la carte permettant de retourner dans l'accueil
    html_button = '''<button onclick="window.location.href='https://tremblement-du-monde.com/'"
        style="position: absolute; top: 10px; left: 45%; z-index: 1000; font-size: 16px; padding: 10px 12px;">
        Retour à l'Accueil</button>''
    carte.get_root().html.add_child(folium.Element(html_button))
```

Cercle :

```
def cercle(mag):
    if mag <= 2:
        return 5
    elif mag > 2 and mag <= 4:
        return 10
    elif mag > 4 and mag <= 6:
        return 20
    else:
        return 30
```

Cercle HTML :

```
template = """
{% macro html(this, kwargs) %}

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Carte temps réel</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

</head>
<body>

<div id='maplegend' class='maplegend'
  style='position: absolute; z-index:9999; border:2px solid grey; background-color:rgba(255, 255, 255, 0.8);
  border-radius:6px; padding: 10px; font-size:14px; right: 20px; bottom: 20px;'>

<div class='legend-title'><H3>Légende :</H3></div>
<div class='legend-scale'>
  <ul class='legend-labels'>
    <table id = "tab">
      <tr>
        <th><p>Magnitude < 2 </p></th>
        <th><div id="moncercle1"></div></th>
      </tr>
      <tr>
        <th><p>Magnitude entre 2 et 4 </p></th>
        <th><div id="moncercle2"></div></th>
      </tr>
      <tr>
        <th><p>Magnitude entre 4 et 6 </p></th>
        <th><div id="moncercle3"></div></th>
      </tr>
      <tr>
        <th><p>Magnitude > 6 </p></th>
        <th><div id="moncercle4"></div></th>
      </tr>
    </ul>
  </div>
</div>

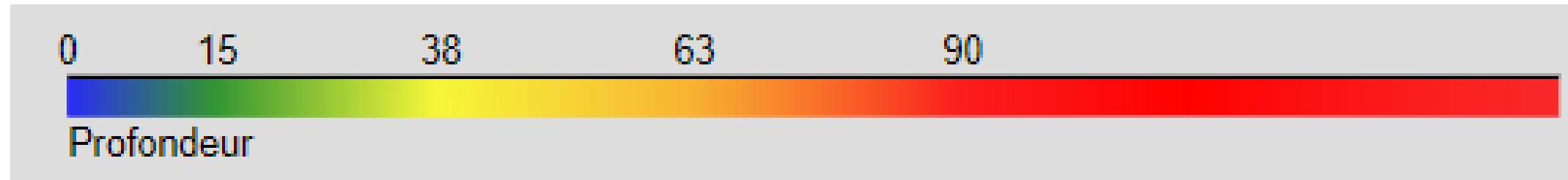
</body>
</html>
```

Cercle CSS:

```
<style type='text/css'>
.maplegend .legend-title {
  text-align: left;
  margin-bottom: 5px;
  font-weight: bold;
  font-size: 90%;
}
.maplegend .legend-scale ul {
  margin: 0;
  margin-bottom: 5px;
  padding: 0;
  float: left;
  list-style: none;
}
.maplegend .legend-source {
  font-size: 80%;
  color: #777;
  clear: both;
}
.maplegend a {
  color: #777;
}
#moncercle1{
  background:#d80e18;
  border-radius:50%;
  width:10px;
  height:10px;
  border:2px solid #930396;
}
#moncercle2{
  background:#d80e18;
  border-radius:50%;
  width:20px;
  height:20px;
  border:2px solid #930396;
}
#moncercle3{
  background:#d80e18;
  border-radius:50%;
  width:40px;
  height:40px;
  border:2px solid #930396;
}
#moncercle4{
  background:#d80e18;
  border-radius:50%;
  width:60px;
  height:60px;
  border:2px solid #930396;
}
</style>
{{% endmacro %}}""
```

Système d'échelle de profondeur

```
import branca.colormap as cm#branca permet d'avoir une echelle de couleurs
couleurs = cm.LinearColormap(["blue", "green", "yellow", "orange", "red"], index=[0, 0.1, 0.25, 0.42, 0.6])#
couleurs.caption = "Profondeur"
carte.add_child(couleurs.scale(0, 150))#ça ajoute à la carte une échelle colorée allant de 0 à 150 corr
```



A detailed satellite image of Earth, centered on the Northern Hemisphere and the Middle East. The image shows the blue oceans, white clouds, and various landmasses in shades of brown, green, and blue. The text 'Fonctions spécifiques:' is overlaid on the left side of the globe.

Fonctions spécifiques :

Créé carte

Version 1 : En temps réel

```
def cree_carte(seismes,plaques_tectoniques):
    min_lon, max_lon = -200, 200#on établie les bords de la cartes pour que l'utilisateurs n'en voie pas plusieurs
    min_lat, max_lat = -100, 100

    #initialisation de la carte en utilisant des tuiles d'imagerie satellite, et limitant la zone visible aux continents.
    carte = folium.Map(max_bounds=True,location=[0, 0], zoom_start=2, tiles="Esri WorldImagery",
                        min_lon=min_lon, max_lon=max_lon,min_lat=min_lat,max_lat=max_lat,min_zoom=2)

    couleurs = cm.LinearColormap(["blue","green", "yellow","orange","red"], index=[0, 0.1, 0.25,0.42,0.6])#on associe des valeurs au couleur grace à cm.LinearColormap
    couleurs.caption = "Profondeur"
    carte.add_child(couleurs.scale(0, 150))#ça ajoute à la carte une échelle colorée allant de 0 à 150 correspondant à la profondeur

    plaques = folium.FeatureGroup(name='Plaques Tectoniques', show = False)#on crée un groupe d'entité pour les plaques tectonique
    folium.GeoJson(plaques_tectoniques).add_to(plaques)# On met les données geographique des plaques dans la carte
    carte.add_child(plaques)#on ajoute les plaques tectoniques à notre carte

    for seisme in seismes["features"]:
        magnitude = seisme["properties"]["mag"]#on récupère la magnitude
        coordonnee = seisme["geometry"]["coordinates"]#on obtient les données geographique
        location = (coordonnee[1], coordonnee[0])#on prend les coordonnées longitude latitude
        profondeur = -coordonnee[2]#on inverse les coordonnées verticales
        date = seisme["properties"]["time"]

        date, temps = date.split('T')#on divise la chaine de caractère en 2 patie : l'heure et la date
        temps = temps[:5] + " UTC"#on supprime les 4 derniers carctères et on les remplace par UTC

        popup_text = f'''Date : {date}<br> Heure : {temps} <br> Magnitude : {magnitude} <br> Profondeur : {profondeur} km'''# popup avec:le temps,la magnitudeet la profondeur
        iframe = folium.IFrame(popup_text,width=200,height=80)#on definit la largeur et la hauteur de la fenetre popup
        popup = folium.Popup(iframe,max_width=300,max_height=100)#on cree le popup sur la carte puis on définit la longueur max et la largeur max

        folium.CircleMarker(#on cree un marqueur en forme de cercle sur la carte
                            location=location,#on spécifie la position du séisme
                            radius= taille_cercle.cercle(magnitude),#le rayon du cercle est proportionnelle à la magnitude du séisme fois 2
                            color=couleurs(profondeur/100),#La couleur du cercle est déterminée par la profondeur du séisme
                            fill_color= couleurs(profondeur/100),#la couleur de remplissage est basse sur la couleur de profondeur

                            popup=popup,#le popup affichera les données du séisme
                            ).add_to(carte)

    carte.add_child(folium.LayerControl())#gestion des calques

    return carte
```

Créé carte

Version 1 : En temps réel

```
def derniers_seismes(seismes):
    html = f'''  

        <H1 style="position: absolute; top: 80%; left: 3%; z-index: 1000; font-size: 20px; padding: 10px 12px; color: red;"> Les 5 derniers séismes : </H1>'''  

    for i in range(5):
        seisme = seismes["features"][i]
        magnitude = seisme["properties"]["mag"]
        date = seisme["properties"]["time"]
        date, temps = date.split('T')#on divise la chaîne de caractère en 2 parties : l'heure et la date
        temps = temps[:5] + " UTC"#on supprime les 4 derniers caractères et on les remplace par UTC
        location = seisme["properties"]["flynn_region"]
        html += f'''<H5 style="position: absolute;  

            top: {i*2+84}%;  

            left: 4%; z-index: 1000;  

            font-size: 16px;  

            padding: 10px 12px;  

            ">• {location}, {date}, {temps}, magnitude : {magnitude}</H5>'''  

    carte.get_root().html.add_child(folium.Element(html))
```

```
seismes = donnees_seisme()
plaques = donnees.donnees_plaques()
carte = cree_carte(seismes, plaques)

retour_accueil()
derniers_seismes(seismes)

heure_actuelle = str(datetime.now())

html = f'''  

    <H1 style="position: absolute; top: 2%; left: 5%; font-family: Arial; z-index: 1000;  

    font-size: 20px; padding: 10px 12px; color: red; font-weight: bold;">  

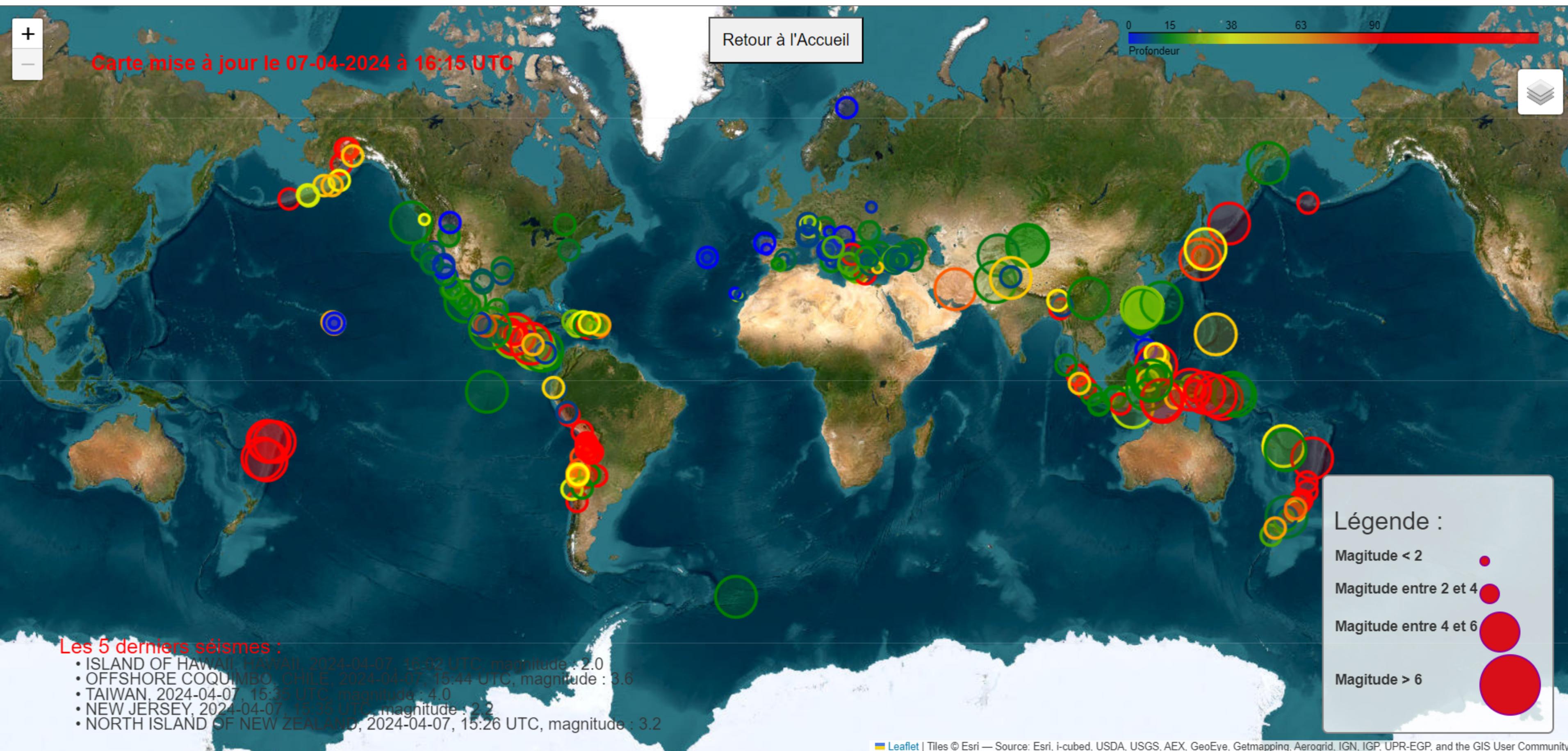
    Carte mis à jour le {heure_actuelle[8:10]}{heure_actuelle[4:8]}{heure_actuelle[:4]} à {heure_actuelle[10:16]} UTC </H1>'''  

carte.get_root().html.add_child(folium.Element(html))

carte.save("carte_temps_reel.html")#la carte est enregistrée comme carte_temps_reel.html
```

Créé carte

Version 1 : En temps réel, Exemple d'exécution



Créé carte

Version 2: Historique

```
# -*- coding: utf-8 -*-

import datetime#permet de manipuler les dates et les heures
import folium#Folium est un puissant outil car il rend possible la conception de cartes interactives.
import folium.plugins# plugins de folium
import branca.colormap as cm#branca permet d'avoir une echelle de couleurs
import donnees#permet de recuperer les données

def cree_carte(plaques_tectoniques):
    min_lon, max_lon = -200, 200#on établie les bords de la cartes
    min_lat, max_lat = -100, 100

    #initialisation de la carte en utilisant des tuiles d'imagerie satellite, et limitant la zone visible aux continents.
    carte = folium.Map(max_bounds=True,location=[0, 0], zoom_start=2, tiles="Esri WorldImagery",
                        min_lon=min_lon, max_lon=max_lon,min_lat=min_lat,max_lat=max_lat,min_zoom=2)

    plaques = folium.FeatureGroup(name='Plaques Tectoniques')#on crée une entité pour les plaques tectonique
    folium.GeoJson(plaques_tectoniques).add_to(plaques)# On met les données geographique des plaques dans la carte
    carte.add_child(plaques)#on ajoute les plaques tectoniques à notre carte.

    points = []

    couleurs = cm.LinearColormap(["blue","green", "yellow","orange","red"], index=[0, 0.1, 0.25,0.42,0.6])#on associe des valeurs au couleur grace à cm.LinearColormap
    couleurs.caption = "Profondeur"
    carte.add_child(couleurs.scale(0, 150))#ça ajoute à la carte une échelle colorée allant de 0 à 150 pour le niv de profondeur

    starttime = '2000-01-01'#date du début du tri
    endtime = '2024-04-01'#date de la fin du tri

    seismes = donnees.donnees_seismes(starttime,endtime,6)

    for seisme in seismes["features"]:
        magnitude = seisme["properties"]["mag"]#on récupère la magnitude
        coordonnee = seisme["geometry"]["coordinates"]#on obtient les données geographique
        location = [coordonnee[0], coordonnee[1]]#on crée un tableau location avec les coordonnées de longitude et de latitude
        profondeur = coordonnee[2]#on obtient la profondeur du séisme
        date_seisme = seisme["properties"]["time"] / 1000.0
        date = datetime.datetime.utcfromtimestamp(date_seisme).strftime('%Y-%m-%d %H:%M:%S UTC')#On convertis la date du séisme (en millisecondes) en une date lisible
        date = date[:10]# on la limite pour que ça affiche uniquement l'année, le mois et le jour.
        popup_text = f'''<p>Date : {date} <br> Magnitude : {magnitude} <br> Profondeur : {profondeur} km</p>'''#on crée le texte pour la fenetre popup
        color = couleurs(profondeur/100)#On détermine la couleur du marqueur en fonction de la profondeur du séisme.
        points.append({"time": date, "popup": popup_text, "coordinates": location,'color': color})#on ajoute le seisme traité à l'ensemble
```

Créé carte

Version 2: Historique

```
features = [#on range les séismes
{
    "type": "Feature",
    "geometry": {#les coordonnées géographique du séisme latitude et longitude
        "type": "Point",
        "coordinates": point["coordinates"],
    },
    "properties": {#données de séismes
        "time": point["time"],#date et heure du séisme
        "popup": point["popup"],#le contenu de la fenetre popup
        "style": {#dico vide pour le style
            "color": ""},
        "icon": 'circle',#le marqueur en forme de cercle
        'iconstyle': {#carctéristiques du marqueur comme la couleur de remplissage, l'opacité, le rayon, etc.
            'fillColor': point["color"],
            'fillOpacity': 0.8,
            'stroke': 'true',
            'radius': magnitude*2
        },
    },
},
}
for point in points
]

folium.plugins.TimestampedGeoJson(#affiche les séismes sur la carte en fonction de leur date.
{"type": "FeatureCollection", "features": features},
period="P1M",#La fréquence des s de données (ici, 1 mois).
add_last_point=True,#le dernier poin est inclus dans l'animation
auto_play=False,#l'animation ne démarre pas automatiquement
loop=False,#l'animation ne se répète pas en boucle
max_speed=1,#la vitess max 1fps
loop_button=True,#bouton pour activer/desactiver la boucle quand on veut
date_options="YYYY/MM/DD",#c'est le format de comment on affiche la date
time_slider_drag_update=True,#ça permet aux utilisateurs de faire glisser le curseur de temps pour explorer les séismes
duration="P2M",#durée totale de l'animation ici 2 mois
).add_to(carte)

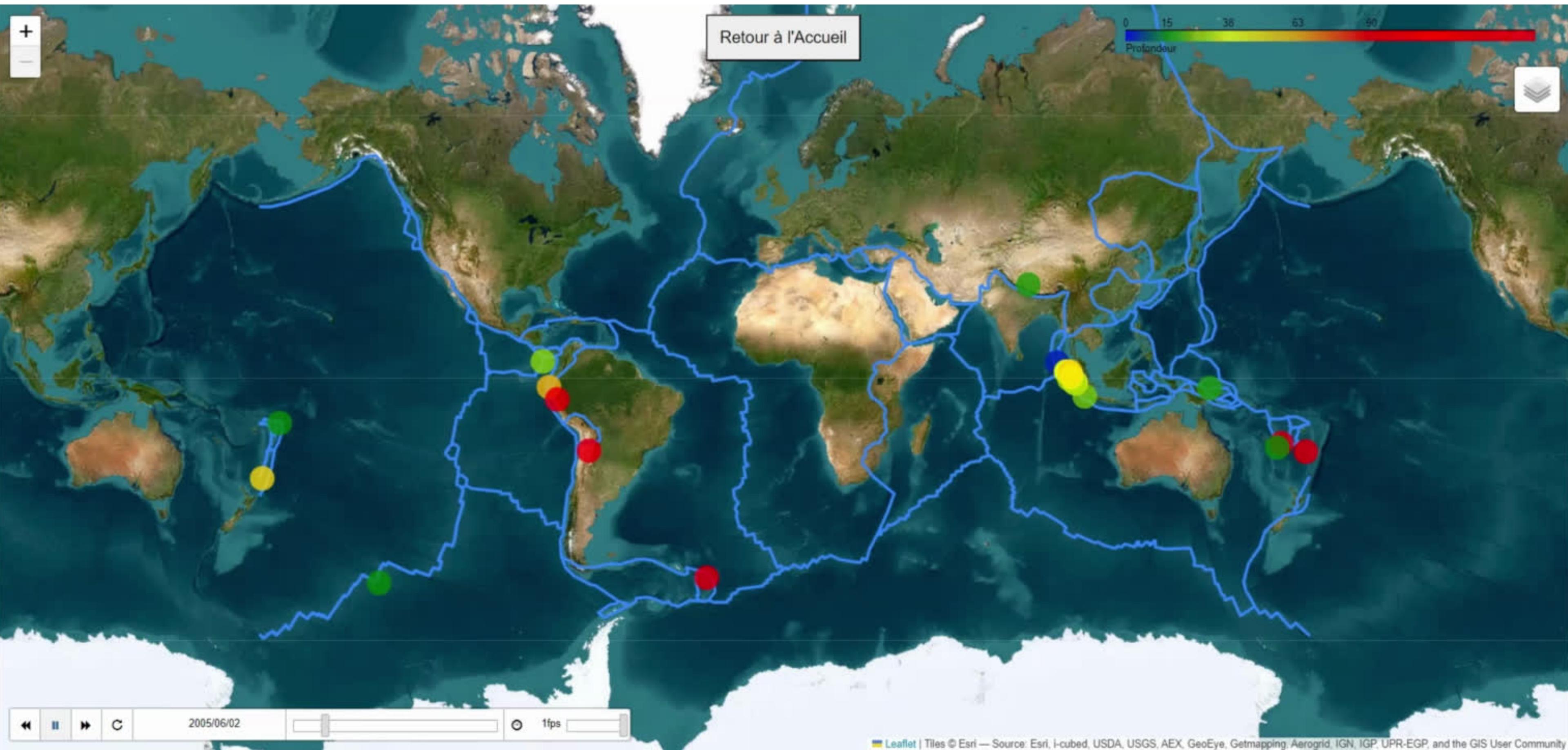
carte.add_child(folium.LayerControl())#on ajoute un contrôle de calques à la carte pour permettre aux utilisateurs de basculer entre les couches (séismes, plaques tectoniques, etc.)

return carte,features
plaques = donnees.donnees_plaques()

carte,features = cree_carte(plaques)
retour_accueil()
carte.save("carte.html")
```

Créé carte

Version 2: historique, exemple d'exécution



Créé carte

Version 3: Par jour

Création de endtime

```
def end(startime):
    endtime = startime

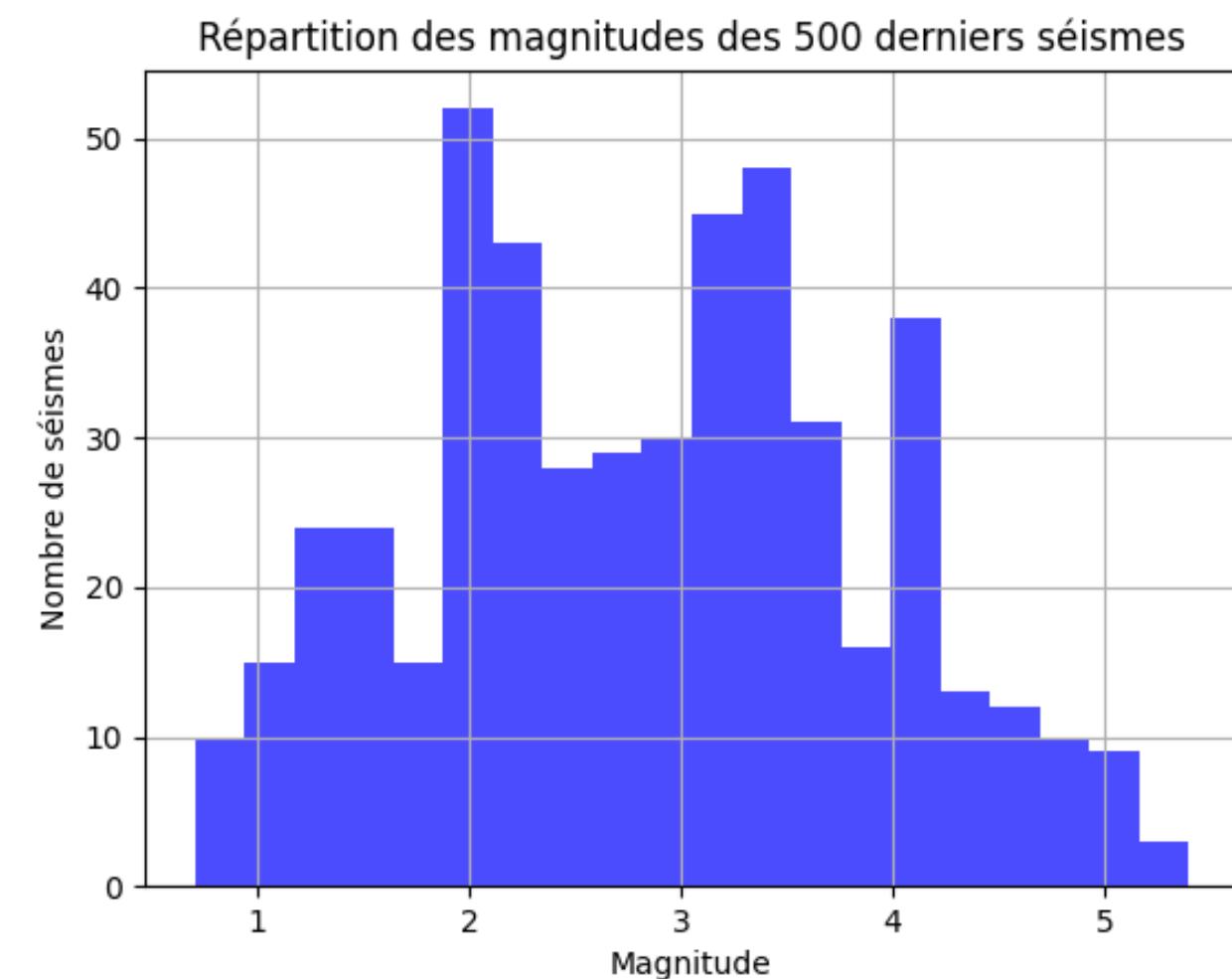
    if endtime[5:7] == "12" and endtime[-2::] == "31":
        year = str(int(endtime[0:4]) + 1)
        endtime = year + "-01-01"
    elif endtime[5:7] in ["01","03","05","07","08","10"] and endtime[-2::] == "31":
        if int(endtime[5:7]) < 9:
            mois = "0" + str(int(endtime[5:7]) + 1)
            endtime = endtime[0:5] + mois + "-01"
        else:
            mois = str(int(endtime[5:7]) + 1)
            endtime = endtime[0:5] + mois + "-01"
    elif endtime[5:7] in ["04","06","09","11"] and endtime[-2::] == "30":
        if int(endtime[5:7]) < 9:
            mois = "0" + str(int(endtime[5:7]) + 1)
            endtime = endtime[0:5] + mois + "-01"
        else:
            mois = str(int(endtime[5:7]) + 1)
            endtime = endtime[0:5] + mois + "-01"
    elif endtime[5:7] == "02" and endtime[-2::] == "29" and endtime[0:5] in ["2000","2004","2008","2012","2016","2020","2024","2028"]:
        mois = "0" + str(int(endtime[5:7]) + 1)
        endtime = endtime[0:5] + mois + "-01"
    elif endtime[5:7] == "02" and endtime[-2::] == "28":
        mois = "0" + str(int(endtime[5:7]) + 1)
        endtime = endtime[0:5] + mois + "-01"
    else:
        if int(endtime[-2::]) < 9 :
            jour = "0" + str(int(endtime[-2::]) + 1)
            endtime = endtime[0:-2]+ jour
        else:
            jour = str(int(endtime[-2::]) + 1)
            endtime = endtime[0:-2]+ jour
    print(endtime)
```

A detailed satellite image of Earth, centered on the continent of Europe. The image shows the blue oceans, white clouds, and various landmasses in shades of brown, green, and blue. The text "Statistiques:" is overlaid on the left side of the globe.

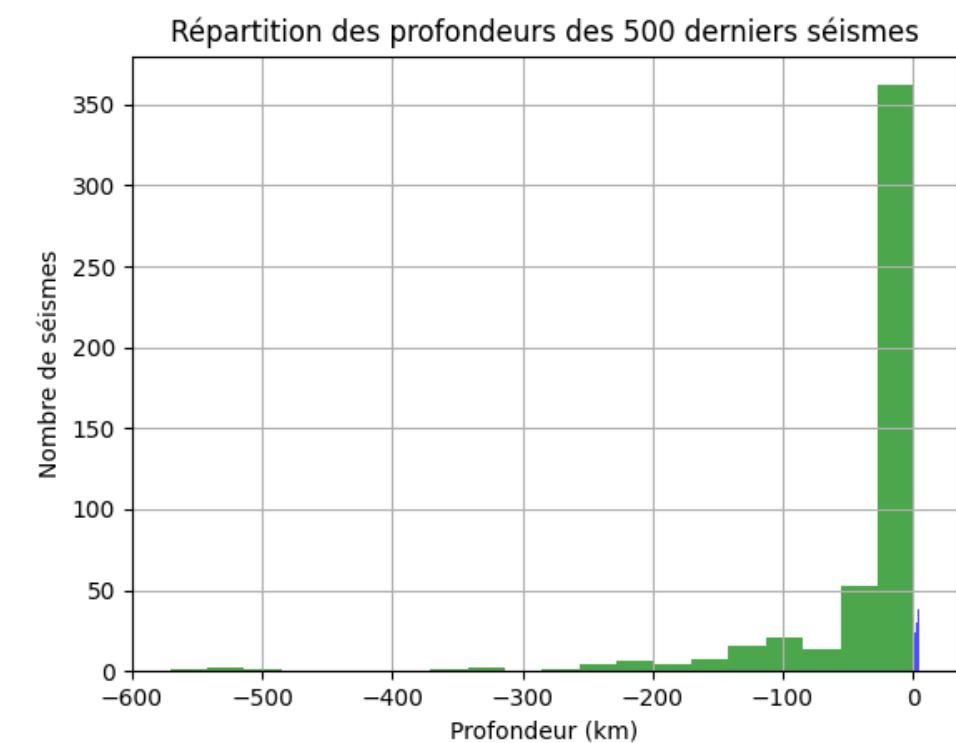
Statistiques:

Statistiques en temps réel:

```
def graphique_seisme_magnitude(data):
    magnitudes = [float(seisme['properties']['mag']) for seisme in data['features']]
    plt.hist(magnitudes, bins=20, color='blue', alpha=0.7)
    plt.xlabel('Magnitude')
    plt.ylabel('Nombre de séismes')
    plt.title('Répartition des magnitudes des 500 derniers séismes')
    plt.grid(True)
    plt.savefig('magnitudes.png')
    plt.show()
```



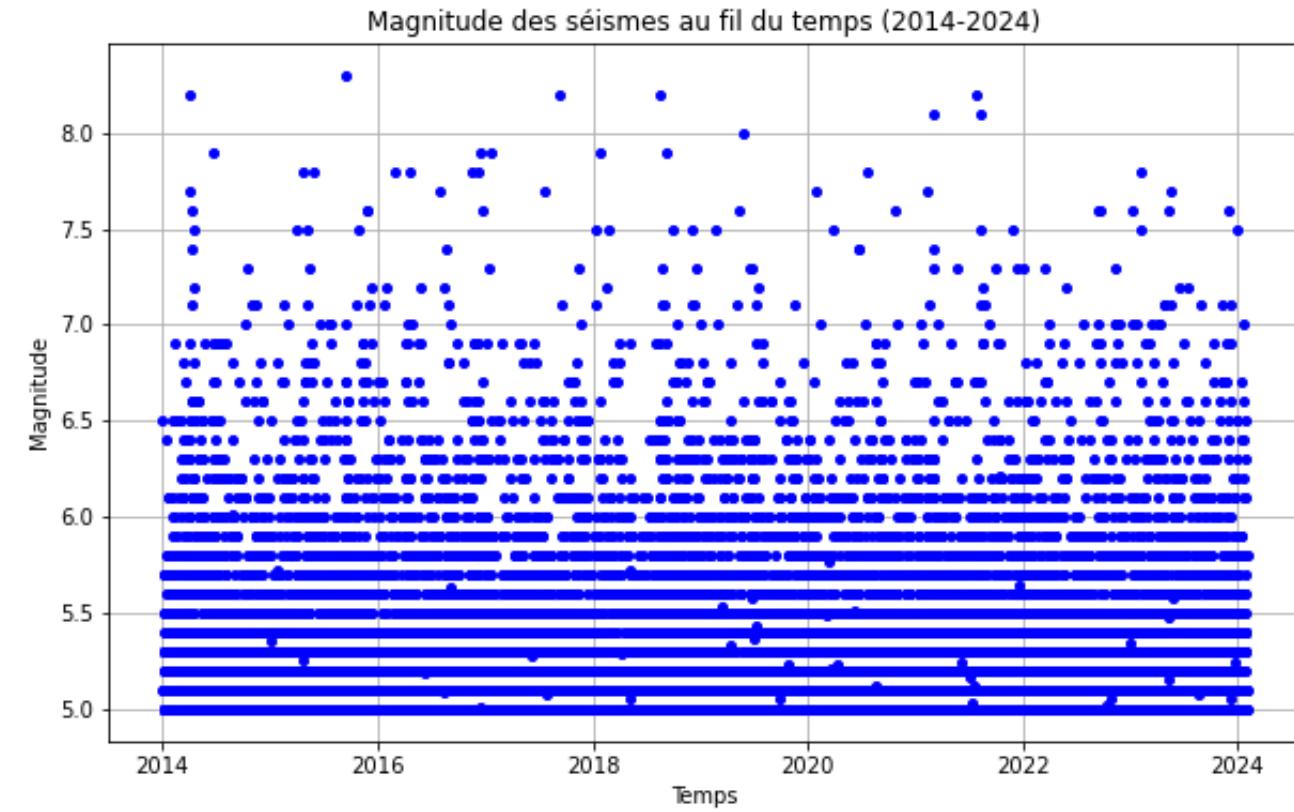
```
def graphique_seisme_profondeur(data):
    profondeurs = [float(seisme['geometry']['coordinates'][2]) for seisme in data['features']]
    plt.hist(profondeurs, bins=20, color='green', alpha=0.7)
    plt.xlabel('Profondeur (km)')
    plt.ylabel('Nombre de séismes')
    plt.title('Répartition des profondeurs des 500 derniers séismes')
    plt.grid(True)
    plt.savefig('profondeurs.png')
    plt.show()
```



Statistiques en Général:

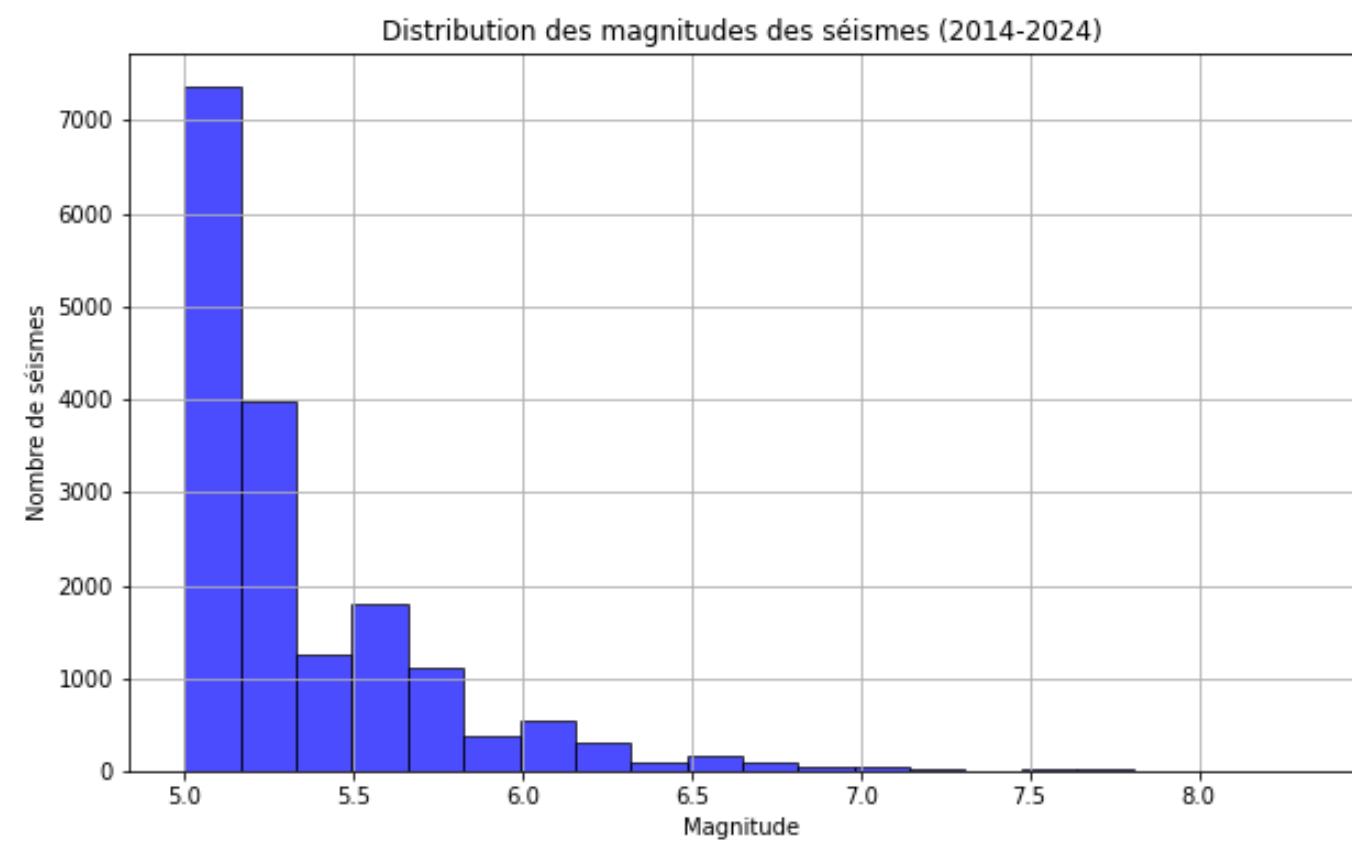
```
def plot_seismes_magtemps(data):
    magnitudes = []
    temps = []
    for feature in data['features']:
        magnitude = feature['properties']['mag']
        time = feature['properties']['time']
        temps.append(datetime.fromtimestamp(time / 1000)) # Convertir le temps en format lisible
        magnitudes.append(magnitude)

    plt.figure(figsize=(10, 6))
    plt.plot(temps, magnitudes, 'o', color='blue', markersize=4)
    plt.xlabel('Temps')
    plt.ylabel('Magnitude')
    plt.title('Magnitude des séismes au fil du temps (2014-2024)')
    plt.grid(True)
    plt.savefig('histoire.png')
    plt.show()
```



```
def plot_seismes_histogramme(data):
    magnitudes = []
    for feature in data['features']:
        magnitude = feature['properties']['mag']
        magnitudes.append(magnitude)

    plt.figure(figsize=(10, 6))
    plt.hist(magnitudes, bins=20, color='blue', edgecolor='black', alpha=0.7)
    plt.xlabel('Magnitude')
    plt.ylabel('Nombre de séismes')
    plt.title('Distribution des magnitudes des séismes (2014-2024)')
    plt.grid(True)
    plt.savefig('histogramme.png')
    plt.show()
```



Statistiques en Général:



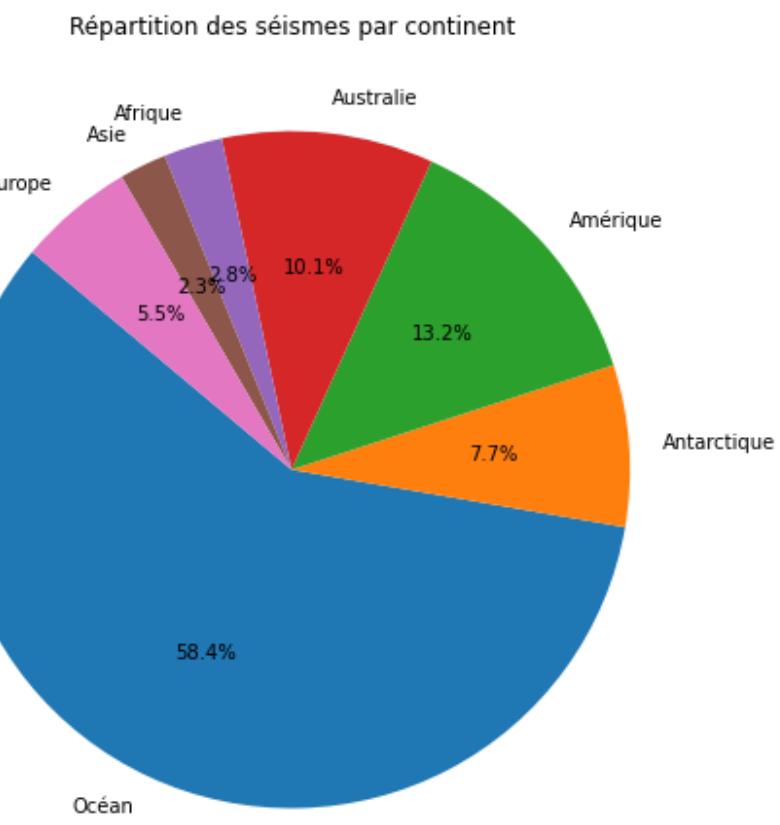
Statistiques en Général:

```
def plot_seismes_par_continent(data):
    continents_counts = {}
    for feature in data['features']:
        geometry = feature['geometry']
        if 'coordinates' in geometry:
            coordinates = geometry['coordinates']

            lon, lat = coordinates[:2] # Latitude et longitude
            continent = f_continente(lat, lon)
            if continent in continents_counts:
                continents_counts[continent] += 1
            else:
                continents_counts[continent] = 1

    # Création du diagramme en camembert
    plt.figure(figsize=(8, 8))
    plt.pie(continents_counts.values(), labels=continents_counts.keys(), autopct='%.1f%%', startangle=140)
    plt.title('Répartition des séismes par continent')
    plt.savefig('représentation.png')
    plt.show()

def f_continente(lat, lon):
    if -180 <= lon <= 180 and -90 <= lat <= 90: # Vérifier si les coordonnées sont valides
        if 50 <= lat <= 70 and -180 <= lon <= 180: # Europe
            return 'Europe'
        elif -25 <= lat <= 50 and -100 <= lon <= 40: # Amérique
            return 'Amérique'
        elif -40 <= lat <= -25 and -100 <= lon <= -30: # Afrique
            return 'Afrique'
        elif -60 <= lat <= -40 and -70 <= lon <= 180: # Antarctique
            return 'Antarctique'
        elif -10 <= lat <= 20 and 70 <= lon <= 100: # Asie
            return 'Asie'
        elif -40 <= lat <= -10 and 100 <= lon <= 180: # Australie
            return 'Australie'
        else:
            return 'Océan'
    else:
        return 'Inconnu'
```



A detailed satellite image of Earth, centered on the continent of Europe. The image shows the blue oceans, white clouds, and various landmasses in shades of brown, green, and blue. The curvature of the Earth is visible against a dark background of space with numerous small stars.

Le site

Accueil

```
<?php
// Vérifie si la variable est bien envoyée
if(isset($_POST['variable'])) {
    // Récupère la variable envoyée par JavaScript
    $a = $_POST['variable'];
    exec("python carte_jour.py $a", $variable);
    // Termine l'exécution du script PHP
    exit();
}

?>

<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <meta name="description" content="Time to Quake ! est le site qui répertoire tout les tremblements du monde en temps réel et historique. Le site présente de la documentation et un quizz !"/>
    <meta name="google-site-verification" content="HZIIzMEuHZb--5cwvp6iQH_PAUP8t1Eg5WL0TNvbfFc" />
    <title>Accueil</title>
    <link href="style_hub.css" rel="stylesheet" type="text/css" />
    <link rel="shortcut icon" href="images/icon.png"/>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
```

Accueil

```
<body>
    <H1>Time To Quake !</H1>
    <H2>Cartes :</H2>
    <a href="carte-en-temps-réel.html">Carte en Temps Réel</a><br>
    <a href="carte-historique.html">Carte Historique</a><br>
    <a onclick="envoyerVariable()">Carte par jour</a>
    <H2>Documentation :</H2>
    <a href="noyau-terre.html">Noyau de la Terre</a><br>
    <a href="Les-tremblements-de-terre.html">Les Outils Technologiques</a><br>
    <a href="Les-tremblements-de-terre-2.html">Les Tremblements de Terre</a>
    <H2>Autres :</H2>
    <a href="quizz.html">Quizz </a><br>
    <a href="statistiques.html">Statistiques</a>
    <script>
        function envoyerVariable() {
            // Définit la variable JavaScript
            var maVariable = prompt("Rentrez une date sous cette forme", "2008-07-26");
            // Envoie la variable à PHP via AJAX
            $.ajax({
                url: '<?php echo $_SERVER['PHP_SELF']; ?>', // URL du script actuel
                type: 'POST',
                data: { variable: maVariable },
                success: function(response) {
                    // Affiche la réponse de PHP dans la console du navigateur
                    console.log('Réponse de PHP : ' + response);
                }
            });
        }
        
    </script>
</body>
</html>
```

Documentation sur les tremblement de terre

```
<!DOCTYPE html>
<html>
  <head>
    <title>Les tremblements de Terre</title>
    <meta charset="utf-8">
    <meta name="keywords" content="mon site,html,la doc">
    <meta name="description" content="mon site informe les gens sur les tremblements de terre">
    <link rel="shortcut icon" href="images/icone.png"/>
    <link rel="stylesheet" type="text/css" href='style.css'>
  </head>
  <body>
    <h1 class="title"><a href="index.html">Time To Quake !</a></h1>

    <div class="dropdown">
      <button class="dropbtn">Les Cartes</button>
      <div class="dropdown-content">
        <a href="carte-en-temps-réel.html">Carte en Temps Réel</a>
        <a href="carte-historique.html">Carte Historique</a>
      </div>
    </div>

    <div class="dropdown">
      <button class="dropbtn">Documentation</button>
      <div class="dropdown-content">
        <a href="noyau-terre.html">Noyau de la Terre</a>
        <a href="Les-tremblements-de-terre.html">Les Outils Technologiques</a>
        <a href="Les-tremblements-de-terre-2.html">Les Tremblements de Terre</a>
      </div>
    </div>

    <div class="dropdown">
      <button class="dropbtn">Autres</button>
      <div class="dropdown-content">
        <a href="quizz.html">Quizz</a>
        <a href="statistiques.html">Statistiques</a>
      </div>
    </div>

    <h1>Les tremblements de terre</h1><br>

    <h2>Qu'est ce que c'est un tremblement de Terre ?</h2>
    <p>Un tremblement de terre est une secousse du sol causée par le déplacement de roches sous la surface de la terre. Ces mouvements peuvent être dus à l'activité volcanique ou magmatique de la terre. Les tremblements de terre peuvent être annoncés ou soudains, et varier en intensité et en durée. Les sismologues utilisent des sismographes pour enregistrer les secousses sismiques. Ces appareils détectent et mesurent les ondes générées par les séismes.</p>
  </body>

```

Documentation sur les tremblement de terre

```
<br>
</p>
<p>

<h2>Qu'est ce que c'est une plaque tectonique</h2>
<p>Les plaques tectoniques sont d'énormes morceaux de la croûte terrestre qui se déplacent lentement et interagissent les unes avec les autres. Leur mouvement est responsable de phénomènes tels que les tremblements de terre, les volcans et la formation de chaînes de montagnes

<h2>Le mouvement des plaques tectoniques</h2>
<p>
Les mouvements relatifs entre les grandes plaques lithosphériques sont de 3 types principaux :

1)de l'étirement, là où se crée les plaques océaniques au niveau des rifts et des grandes dorsales médio-océaniques

2)du raccourcissement, dans les zones de chevauchement de plaques: on parle de subduction lorsqu'une plaque océanique passe sous une autre plaque (par exemple la ceinture du Pacifique au Chili, Alaska, Japon) ou de collision lorsque deux plaques continentales sont impliquées (par exemple les chaînes himalayenne ou alpine)

3)du coulissage latéral (ou décrochement), comme au niveau des failles transformantes ou des célèbres failles de San Andreas (Californie, Etats-Unis) ou nord-anatolienne (Turquie).

<h2>Où ce passe les séismes</h2>
<p>Environ 80 % des séismes dans le monde se produisent le long de la ceinture de feu de l'océan Pacifique. Cette région est ainsi nommée en raison de l'activité volcanique importante qui s'y déroule. La plupart des séismes surviennent dans des zones de failles où les plaques tectoniques (d'énormes dalles de pierre formant la croûte terrestre) entrent en collision ou glissent les unes sur les autres. Ces mouvements sont souvent imperceptibles à la surface, mais une énorme tension peut s'accumuler entre les plaques. Lorsqu'elle est libérée brusquement, elle génère des ondes sismiques qui se propagent à travers la roche jusqu'à la surface de la Terre. D'autres tremblements de terre peuvent survenir loin des zones de failles lorsque les plaques sont étirées ou compressées.

<h2>Les dégâts causés par les séismes</h2>
<p>La plupart des séismes sont trop faibles pour être ressentis, mais les tremblements de terre importants peuvent entraîner des dégâts matériels considérables. Les séismes peuvent provoquer des effondrements de bâtiments, des glissements de terrain et des tsunamis, entraînant des pertes humaines et des destructions. Nombre de séismes : de 500 000 à 1 million de tremblements de terre par an, dont 100 000 ressentis et 1 000 capables de causer des dégâts car peu de séismes majeurs atteignent des régions habitées .

</body>
</html>
```

Activer Windows
Accédez aux paramètres pour activer Windows.

Documentation sur les outils technologiques

```
<!DOCTYPE html>
<html>

    <head>
        <title>Les Outils Technologiques</title>
        <meta charset="utf-8">
        <meta name="description" content="mon site informe les gens sur les tremblements de terre">
        <link rel="shortcut icon" href="images/icon.png"/>
        <link rel="stylesheet" type="text/css" href='style.css'>
    </head>
    <body>
        <h1 class="title"><a href="index.html">Time To Quake !</a></h1>

        <div class="dropdown">
            <button class="dropbtn">Les Cartes</button>
            <div class="dropdown-content">
                <a href="carte-en-temps-reel.html">Carte en Temps Réel</a>
                <a href="carte-historique.html">Carte Historique</a>
            </div>
        </div>

        <div class="dropdown">
            <button class="dropbtn">Documentation</button>
            <div class="dropdown-content">
                <a href="noyau-terre.html">Noyau de la Terre</a>
                <a href="Les-tremblements-de-terre.html">Les Outils Technologiques</a>
                <a href="Les-tremblements-de-terre-2.html">Les Tremblements de Terre</a>
            </div>
        </div>

        <div class="dropdown">
            <button class="dropbtn">Autres</button>
            <div class="dropdown-content">
                <a href="quizz.html">Quizz</a>
                <a href="statistiques.html">Statistiques</a>
            </div>
        </div>

        <h1>Les Outils Technologiques</h1>

        <h2>Qu'est ce que c'est un sismographe et à quoi sert-il?</h2>
        <p>
            Un sismographe, ou sismomètre, est un instrument utilisé pour détecter et enregistrer les tremblements de terre. Généralement, il est constitué d'une masse fixée à une base fixe. Lors d'un tremblement de terre, la base bouge et la masse reste immobile. Les sismographes sont des instruments utilisés pour enregistrer le mouvement du sol. Un sismographe est solidement fixé à la surface de la terre de sorte que lorsque la terre tremble, l'ensemble de l'unité tremble avec elle, SAUF la masse sur le ressort, qui a une inertie
            <img src= "images/seisme8.jpg">
        </p>
    </body>
</html>
```

Documentation sur les outils technologiques

<h2>Comment un sismographe enregistre-t-il les ondes sismiques ?</h2>

<p>

Les sismographes sont conçus de manière à ce que de légères vibrations terrestres déplacent les instruments ; la masse suspendue (M) a cependant tendance à rester au repos, et son stylet enregistreur enregistre cette différence de mouvement. Ce mouvement - le signal d'une onde sismique - peut ensuite être enregistré sur un tambour rotatif.

<h2> Qu'est-ce que l'échelle de Richter</h2>

<p>

L'échelle de magnitude de Richter, également connue sous le nom d'échelle de magnitude locale (M), attribue un nombre pour quantifier la quantité d'énergie sismique libérée par un tremblement de terre. Micro tremblements de terre, non ressentis. Généralement non ressenti, mais enregistré. L'échelle de Richter est une unité de mesure de la force d'un séisme. Elle a été établie en 1935 par le géologue américain Charles Francis Richter.

C'est une échelle dite logarithmique : un séisme atteignant 5 sur l'échelle de Richter est en fait 10 fois plus puissant qu'un séisme de magnitude 4. En principe, cette échelle n'a pas de limite supérieure. Mais les géologues estiment qu'étant donné la résistance maximale des roches terrestres, aucun séisme supérieur à 10 ne pourra jamais être enregistré. La limite inférieure de l'échelle de Richter est déterminée par la sensibilité des sismographes. Les plus sensibles sont capables de détecter la chute d'une brique depuis une hauteur de 1 mètre, événement qui produit une magnitude de -2. Concrètement, il faut qu'un séisme atteigne une magnitude de 3 sur l'échelle de Richter pour qu'il ait une chance d'être ressenti par des humains. À partir de 4, on le ressent nettement, mais il n'y a généralement pas de dégâts causés aux habitations. À partir de 5, les premiers dégâts peuvent apparaître.

<h2>Qu'est-ce qu'un séisme de classe 9 ?</h2>

<p>

Un tremblement de terre est la secousse de la surface de la Terre due à la libération soudaine d'énergie dans la croûte terrestre, en conséquence, des ondes sismiques (également appelées ondes S) sont créées. Les activités sismiques dans une zone déterminent le type et l'intensité du tremblement de terre.

</p>

<h2>Quelle est la différence entre le sismographe et l'échelle de Richter ?</h2>

<p>

Un sismographe est un instrument utilisé pour détecter et enregistrer les tremblements de terre. Il est également connu sous le nom de sismomètre. Il enregistre le mouvement du sol lors d'un tremblement de terre. L'échelle de Richter est une valeur numérique utilisée pour mesurer la puissance ou la magnitude des tremblements de terre.

</p>

</body>
</html>

Documentation sur le noyau de la terre

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <meta name="description" content="Time to Quake ! est le site qui répertorie tout les tremblements du monde en temps réel et historique. Le site présente de la documentation et un quizz !"/>
    <meta name="google-site-verification" content="HZIIzMEuHZb--5cwvp6iQH_PAUP8t1Eg5WL0TNvbffC" />
    <title>Noyau de la Terre</title>
    <link rel="shortcut icon" href="images/icon.png"/>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

<h1 class="title"><a href="index.html">Time To Quake !</a></h1>

<div class="dropdown">
    <button class="dropbtn">Les Cartes</button>
    <div class="dropdown-content">
        <a href="carte-en-temps-réel.html">Carte en Temps Réel</a>
        <a href="carte-historique.html">Carte Historique</a>
    </div>
</div>

<div class="dropdown">
    <button class="dropbtn">Documentation</button>
    <div class="dropdown-content">
        <a href="noyau-terre.html">Noyau de la Terre</a>
        <a href="Les-tremblements-de-terre.html">Les Outils Technologiques</a>
        <a href="Les-tremblements-de-terre-2.html">Les Tremblements de Terre</a>
    </div>
</div>

<div class="dropdown">
    <button class="dropbtn">Autres</button>
    <div class="dropdown-content">
        <a href="quizz.html">Quizz</a>
        <a href="statistiques.html">Statistiques</a>
    </div>
</div>

<h1>Découverte du noyau de la Terre</h1>
<p> Notre connaissance de la structure interne de notre planète est venue par étapes. Grâce à Newton et à sa loi de la gravitation universelle, on peut estimer la masse de la Terre à partir de la force gravitationnelle que la Terre exerce sur les corps à sa surface. Ceci nous renseigne sur le fait que la densité de la partie interne de la Terre est beaucoup plus élevée que la densité que nous observons à la surface.</p>

<h2> Que peuvent nous apprendre les séismes? </h2>
<p> Les séismes génèrent plusieurs types d'ondes, dont des ondes de pression (ondes P) et des ondes de cisaillement (ondes S). Les ondes de compression se propagent aussi bien dans les milieux solides que dans les milieux liquides.
```

Documentation sur le noyau de la terre

```
<h3>La découverte du noyau interne de la Terre</h3>
<p>C'était le modèle connu de la Terre, et pourtant Inge Lehmann remarqua qu'on avait détecté des ondes sismiques dans la zone d'ombre... Le modèle n'était donc pas bon. Regardons donc le modèle proposé par Inge Lehmann pour expliquer les observations.</p>
<p>Dans ce modèle, elle a divisé le noyau en deux parties : un noyau externe comme ci-dessus, et un noyau interne dans lequel les ondes se propagent plus rapidement que dans le noyau externe, et dont le rayon est d'environ 2/9 de celui de la Terre. Pour connaître la trajectoire des ondes, il faut donc utiliser l'équation de la loi de réfraction de la lumière: <br/>
<img src= images/equation.svg> <br/>
</p>
<p>Mais que se passe-t-il si <math xmlns="http://www.w3.org/1998/Math/MathML">
<mi>sin</mi>
<mo data-mjx-texclass="NONE">#x2061;</mo>
<msub>
  <mi>#x3B8;</mi>
  <mn>1</mn>
</msub>
</math> est assez proche de 1, c'est-à-dire si l'onde arrive presque tangentiellement sur le noyau interne? Le côté droit de l'équation peut dépasser 1 ! Il ne peut donc être égal à sin <math xmlns="http://www.w3.org/1998/Math/MathML">
<msub>
  <mi>#x3B8;</mi>
  <mn>2</mn>
</msub>
<mo>!</mo>
</math>
Cela signifie que l'onde ne peut pénétrer dans le noyau interne.. Que se passe- t-il alors? L'onde est réfléchie sur le noyau interne en suivant la loi de la réflexion, c'est à dire avec l'angle d'incidence égal à l'angle de réflexion</p>
<img src= images/rayons.png>
<p>Voici donc comment se propagent les ondes sismiques dans le modèle d'Inge Lehmann. Les rayons noirs sont réfractés dans le noyau extérieur et réfléchis sur le noyau intérieur </p>
<p>Nous pouvons donc avoir la structure entière de la Terre : </p>
<img src= images/structure.png>

</body>
</html>
```

Quizz html

```
<!DOCTYPE html>
<html>
<head>
    <title>Quizz</title>
    <meta charset="utf-8">
    <meta name="keywords" content="mon site,html,la doc">
    <meta name="description" content="mon site informe les gens sur les tremblements de terre">
    <link rel="shortcut icon" href="https://media.sciencephoto.com/image/c0077777/800wm/C0077777-Earthquake_Fissure.jpg"/>
    <link rel="stylesheet" type="text/css" href='style-quizz.css'>
</head>
<body onload="NextQuestion(0)">
    <main>

        <div class="modal-container" id="score-modal">

            <div class="modal-content-container">

                <h1>Félicitation le quiz est complété.</h1>

                <div class="grade-details">
                    <p>Tentatives : 10</p>
                    <p>Mauvaise réponse : <span id="wrong-answers"></span></p>
                    <p>Bonne réponse : <span id="right-answers"></span></p>
                    <p>pourcentage de réussite : <span id="grade-percentage"></span>%</p>
                    <p><span id="remarks"></span></p>
                </div>

                <div class="modal-button-container">
                    <button onclick="closeScoreModal()">Nouveau QCM</button>
                </div>

            </div>
        </div>

        <div class="game-quiz-container">

            <div class="game-details-container">
                <h1>Score : <span id="player-score"></span> / 10</h1>
                <h1> Question : <span id="question-number"></span> / 10</h1>
            </div>

            <div class="game-question-container">
                <h1 id="display-question"></h1>
            </div>

            <div class="game-options-container">
                <div class="modal-container" id="option-modal">

```

Activer Windows

Accédez aux paramètres pour activer Window

Quizz

```
<div class="modal-content-container">
    <h1>Choisissez une réponse</h1>

    <div class="modal-button-container">
        <button onclick="closeOptionModal()">Commencer le QCM</button>
    </div>

</div>

<span>
    <input type="radio" id="option-one" name="option" class="radio" value="optionA" />
    <label for="option-one" class="option" id="option-one-label"></label>
</span>

<span>
    <input type="radio" id="option-two" name="option" class="radio" value="optionB" />
    <label for="option-two" class="option" id="option-two-label"></label>
</span>

<span>
    <input type="radio" id="option-three" name="option" class="radio" value="optionC" />
    <label for="option-three" class="option" id="option-three-label"></label>
</span>

<span>
    <input type="radio" id="option-four" name="option" class="radio" value="optionD" />
    <label for="option-four" class="option" id="option-four-label"></label>
</span>

</div>

<div class="next-button-container">
    <button onclick="handleNextQuestion()">Suivant</button>
</div>

</div>
</main>
<script src="quizz.js"></script>
</body>
</html>
```

Activer Windows
Accédez aux paramètres pour activer

Quizz js

```
let shuffledQuestions = []

function handleQuestions() {
    while (shuffledQuestions.length <= 9) {
        const random = questions[Math.floor(Math.random() * questions.length)]
        if (!shuffledQuestions.includes(random)) {
            shuffledQuestions.push(random)
        }
    }
}

let questionNumber = 1
let playerScore = 0
let wrongAttempt = 0
let indexNumber = 0

function NextQuestion(index) {
    handleQuestions()
    const currentQuestion = shuffledQuestions[index]
    document.getElementById("question-number").innerHTML = questionNumber
    document.getElementById("player-score").innerHTML = playerScore
    document.getElementById("display-question").innerHTML = currentQuestion.question;
    document.getElementById("option-one-label").innerHTML = currentQuestion.optionA;
    document.getElementById("option-two-label").innerHTML = currentQuestion.optionB;
    document.getElementById("option-three-label").innerHTML = currentQuestion.optionC;
    document.getElementById("option-four-label").innerHTML = currentQuestion.optionD;
}

function checkForAnswer() {
    const currentQuestion = shuffledQuestions[indexNumber]
    const currentQuestionAnswer = currentQuestion.correctOption
    const options = document.getElementsByName("option");
    let correctOption = null

    options.forEach((option) => {
```

```
        correctOption = option.labels[0].id
    })
}

if (options[0].checked === false && options[1].checked === false && options[2].checked === false && options[3].checked == false) {
    document.getElementById('option-modal').style.display = "flex"
}

options.forEach((option) => {
    if (option.checked === true && option.value === currentQuestionAnswer) {
        document.getElementById(correctOption).style.backgroundColor = "green"
        playerScore++
        indexNumber++

        setTimeout(() => {
            questionNumber++
        }, 1000)
    }

    else if (option.checked && option.value !== currentQuestionAnswer) {
        const wrongLabelId = option.labels[0].id
        document.getElementById(wrongLabelId).style.backgroundColor = "red"
        document.getElementById(correctOption).style.backgroundColor = "green"
        wrongAttempt++
        indexNumber++

        setTimeout(() => {
            questionNumber++
        }, 1000)
    }
})
```

```
function handleNextQuestion() {
    checkForAnswer()
    unCheckRadioButtons()

    setTimeout(() => {
        if (indexNumber <= 9) {
            NextQuestion(indexNumber)
        }
    }, 1000)
}
```

Quizz js

```
        }
    else {
        handleEndGame()
    }
resetOptionBackground()
}, 1000);
}

function resetOptionBackground() {
    const options = document.getElementsByName("option");
    options.forEach((option) => {
        document.getElementById(option.labels[0].id).style.backgroundColor = ""
    })
}

function unCheckRadioButtons() {
    const options = document.getElementsByName("option");
    for (let i = 0; i < options.length; i++) {
        options[i].checked = false;
    }
}

function handleEndGame() {
    let remark = null
    let remarkColor = null

    if (playerScore <= 3) {
        remark = "Score bas, relie la doc."
        remarkColor = "red"
    }
    else if (playerScore >= 4 && playerScore < 7) {
        remark = "Niveau intermédiaire, Tu peux faire mieux."
        remarkColor = "orange"
    }
    else if (playerScore >= 7) {
        remark = "Excellent, continue ton bon travail."
        remarkColor = "green"
    }
    const playerGrade = (playerScore / 10) * 100

    document.getElementById('remarks').innerHTML = remark
    document.getElementById('remarks').style.color = remarkColor
    document.getElementById('grade-percentage').innerHTML = playerGrade
}
```

```
document.getElementById('wrong-answers').innerHTML = wrongAttempt
document.getElementById('right-answers').innerHTML = playerScore
document.getElementById('score-modal').style.display = "flex"

}

function closeScoreModal() {
    questionNumber = 1
    playerScore = 0
    wrongAttempt = 0
    indexNumber = 0
    shuffledQuestions = []
    NextQuestion(indexNumber)
    document.getElementById('score-modal').style.display = "none"
}

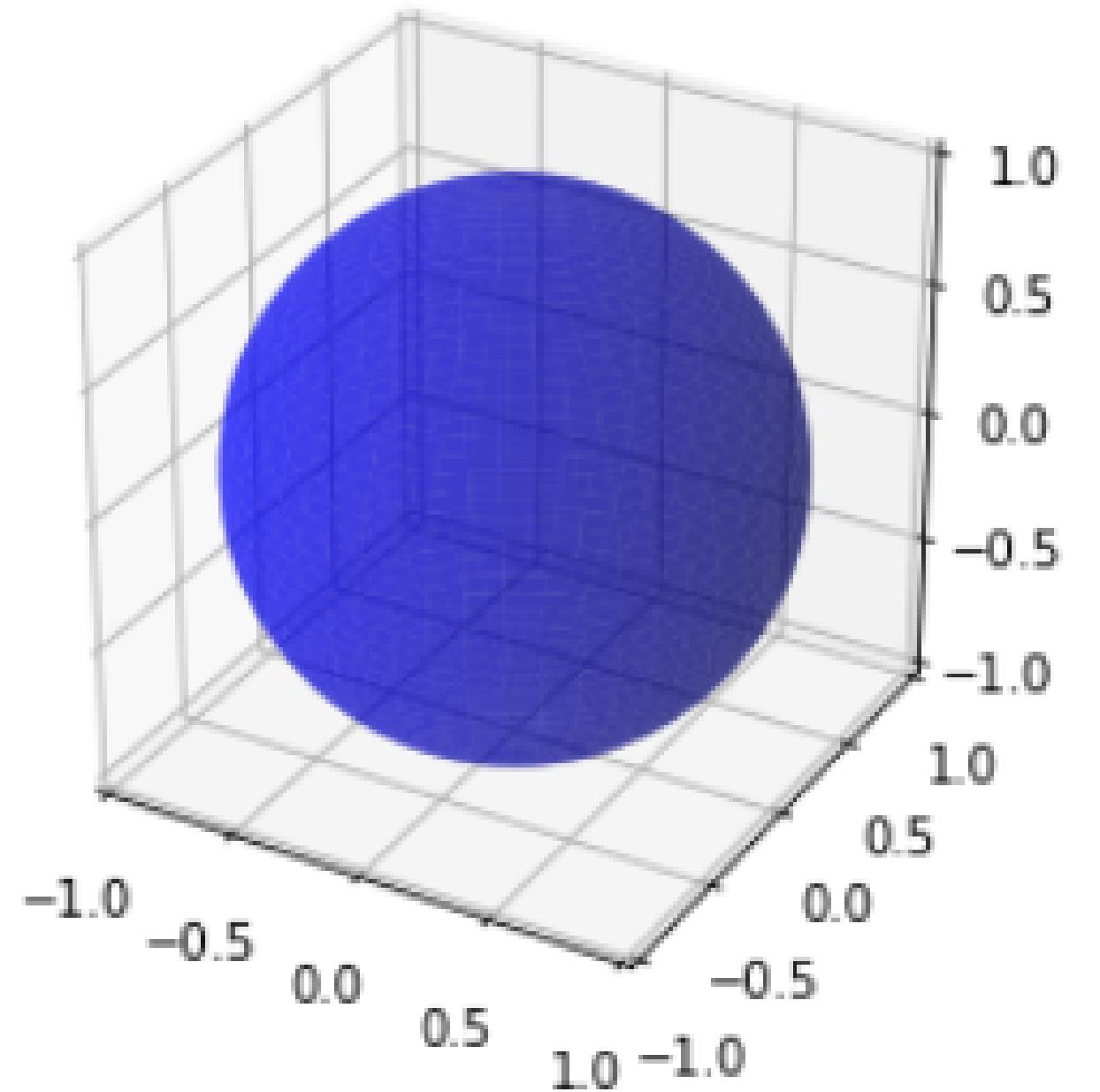
function closeOptionModal() {
    document.getElementById('option-modal').style.display = "none"
}
```



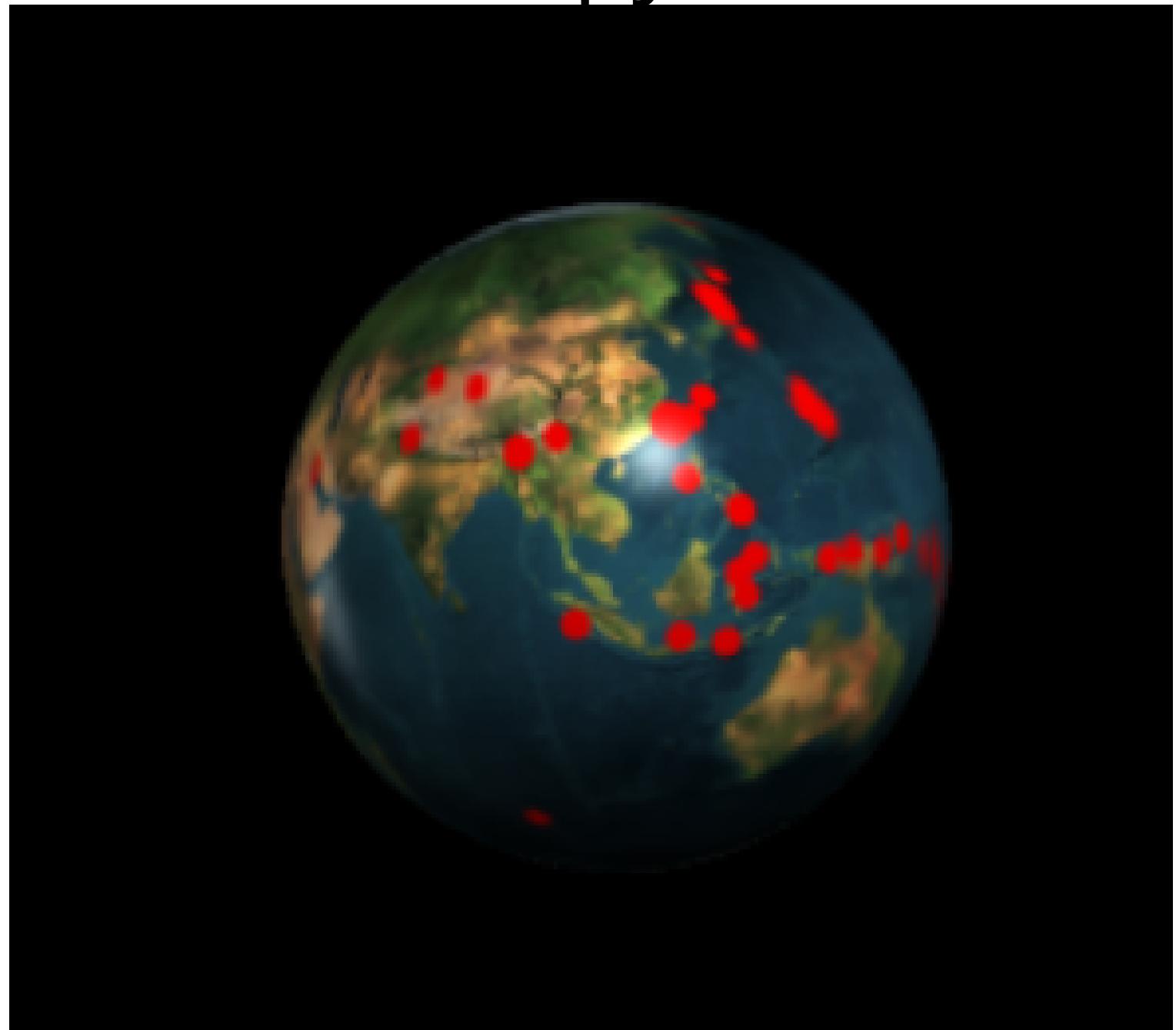
Idée de prolongement

La Terre en 3D

Avec numpy et matplotlib



Avec vpython





Merci