Creating a Python script for IoT devices to send real-time environmental data to a monitoring platform involves several steps. Below is a basic example of how you can achieve this using the MQTT protocol to send data to a cloud-based platform like AWS IoT Core. Keep in mind that specific implementations may vary based on your IoT device and platform choice.

1. *Setup Your IoT Device*:

   - Ensure your IoT device is set up and connected to the internet.

2. *Install Required Libraries*:

   - You might need to install libraries like paho-mqtt for MQTT communication.

   python

   pip install paho-mqtt

3. *Write the Python Script*:

python

```
import paho.mqtt.client as mqtt
import json
import time
import random

# Define your IoT device's parameters
device_id = "your_device_id"
topic = "environmental_data_topic"
mqtt_broker = "your_mqtt_broker_address"
mqtt_port = 1883
update_interval = 10  # Update interval in seconds

def on_connect(client, userdata, flags, rc):
    print(f"Connected with result code {rc}")
```

```
    client.subscribe(topic)

def on_publish(client, userdata, mid):
    print("Data published")

def send_environmental_data():
    client = mqtt.Client(client_id=device_id)
    client.on_connect = on_connect
    client.on_publish = on_publish

    # Connect to the MQTT broker
    client.connect(mqtt_broker, mqtt_port, 60)

    while True:
        # Simulate environmental data (replace with your real data source)
        temperature = random.uniform(20, 30)
        humidity = random.uniform(40, 60)
        data = {
            "temperature": temperature,
            "humidity": humidity,
        }

        # Publish the data to the MQTT topic
        client.publish(topic, json.dumps(data))
        time.sleep(update_interval)

if __name__ == "__main__":
    send_environmental_data()
```

4. *Replace the Simulated Data*:

- Replace the simulated data with actual sensor data readings from your environmental sensors.

5. *Set Up MQTT Broker*:

- Make sure you have an MQTT broker (e.g., Mosquitto) running and properly configured.

6. *Set Up the Monitoring Platform*:

- Configure your monitoring platform (e.g., AWS IoT Core) to receive and process the data sent to the MQTT topic.

7. *Run the Script*:

- Run the Python script on your IoT device to start sending real-time environmental data.

This is a basic example, and you might need to adapt it to your specific IoT device and monitoring platform. Additionally, ensure you handle error conditions and security considerations, such as encryption and authentication, as necessary for your application.