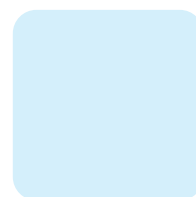
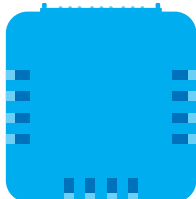


Artec® Links

アーテックリンクス

教員用

双方向性のあるコンテンツ



学習内容

- 双方向性のあるコンテンツ
- 情報通信ネットワークの構成と仕組み
- IoTブロック
- サーバの仕組み

達成目標

- 双方向性のあるコンテンツの理解
- 情報通信ネットワークの構成と仕組みの理解
- サーバの仕組みの理解
- Wi-Fi、サーバへの接続
- サーバと連携したプログラムの作成

活動内容	評価
導入 ①	
● 双方向性のあるコンテンツとは P.2	知識及び技能① 思考・判断・表現
● 情報通信ネットワークの仕組み P.3	知識及び技能① 思考・判断・表現
<ul style="list-style-type: none"> ・ LANとインターネット ・ 無線LANとWi-Fi ・ IoT 	
導入 ②	
● IoTブロック P.5	知識及び技能①
<ul style="list-style-type: none"> ・ IoTブロックの表示方法 ・ IoTブロックの使い方 ① 	
基本課題 ①	
● サーバの仕組み P.7	知識及び技能①
● タイムサーバに接続しよう P.8	知識及び技能①
<ul style="list-style-type: none"> ・ IoTブロックの使い方 ② 	
基本課題 ②	
● タイマー機能付き電波時計の作成 P.10	知識及び技能②
<ul style="list-style-type: none"> ・ アクティビティ図の作成 ・ プログラムの作成 ・ プログラムの実行 	

導入 ①

知識技能①



双方向性のあるコンテンツとは

一方向のコンテンツと双方向性のあるコンテンツ

文字、画像、音声、動画などを組み合わせて、意味のある情報としてあらわされたものを「コンテンツ」といいます。たとえば、デジタルのコンテンツにはWebサイトやSNSなどが、アナログのコンテンツには本や雑誌などがあります。コンテンツは情報を伝達する手段であり、私たちの生活に深く関わっています。

コンテンツの中でも、テレビ番組のようにメディアから視聴者へ向けて一方向に情報が発信されるコンテンツに対して、Webサイトのように操作（入力される情報）によって発信される情報が変化するようなものを「双方向性のあるコンテンツ」といいます。



思考・判断・表現



一方向のコンテンツと双方向性のあるコンテンツの例を考えよう

一方向のコンテンツ	双方向性のあるコンテンツ
例 テレビ	例 Web サイト
新聞	SNS
映画	ライブ配信サービス

知識技能①

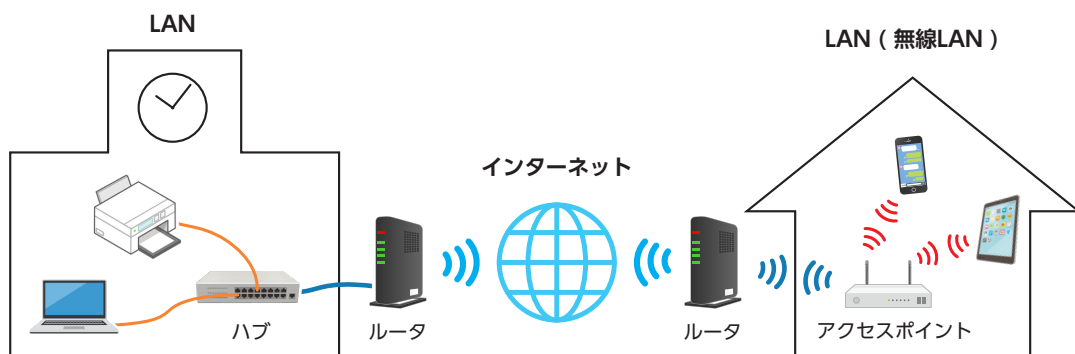


情報通信ネットワークの構成と仕組み

LANとインターネット

家や学校など限られた範囲内のコンピュータを接続したネットワークのことを「LAN（Local Area Network）」といいます。コンピュータなどの情報機器をLANに接続するためには、有線で接続する場合はハブという機器を、無線で接続する場合はアクセスポイントという機器を使います。このようなLANのうち、アクセスポイントを使って無線だけで接続しているネットワークのことを無線LANといいます。

また、LANなどの世界中のネットワークどうしをルーターでつなげた大規模なネットワークのことを「インターネット」といいます。



無線LANとWi-Fi

無線を使って接続している無線LANでは、通信に使用する電波の種類が異なっていると機器同士が接続できません。そのため「Wi-Fi」などの共通規格で定められた電波を使用して通信しています。共通の規格を用いることで、コンピュータ以外の機器も簡単に無線LANに接続することができます。

機器をWi-FiでLANに接続するときはアクセスポイント、またはアクセスポイントを内蔵した無線LANルーターを使用します。Wi-Fi接続では高速なデータ通信ができますが、アクセスポイントから離れると接続がとぎれてしまうといったデメリットがあります。



知識技能①

IoT

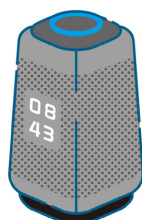
あらゆるモノを無線通信でインターネットに接続する技術を「IoT (Internet of Things) 」といいます。遠く離れたモノどうしがインターネットを通じてリアルタイムで情報をやりとりできるため、遠く離れたところから操作したり、測定したデータをもとに自動で制御したりすることが可能になりました。

例 宅配ドローン



ドローンの現在位置を取得して、
配達先まで飛行する

例 スマートスピーカ



認識した音声进行解析して、家電の操作や
インターネット検索など様々な動作をする

思考・判断・表現



IoTについて考えよう

身の回りにあるIoTを使った製品について調べ、どのようなことを可能にしているか下の表に書き入れましょう。

製品名	IoTによって可能になること
例 スマートウォッチ	例 血圧や心拍数を測定して、健康状態を記録する。
スマートロック	スマホから家の鍵の施錠・開錠ができる。
アプリ付冷蔵庫	冷蔵庫の中の食材を記録して、賞味期限が近付くとアプリから通知が送られる。
忘れもの防止タグ	スマホとタグがBluetoothで接続し、接続が切れると忘れられたと判断して音を鳴らす。


導入 ②

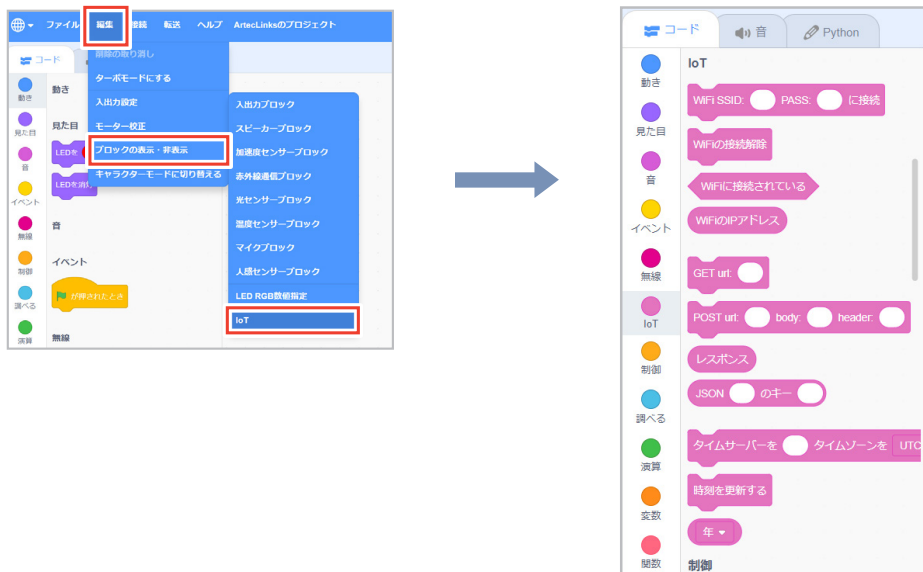
知識技能①

i IoTブロック

ソフトウェアを起動して、メインユニットをWi-Fiに接続しましょう。
メインユニットをWi-Fiに接続するときは、カテゴリ内の「IoTブロック」を使います。

IoTブロックの表示方法

ソフトウェアを起動したとき、IoTブロックは非表示になっています。画面左上のタブから「編集」を選択し、「ブロックの表示・非表示」から「IoTブロック」をクリックして  を表示させましょう。



IoTブロックの使い方 ①



Wi-Fiに接続するためのブロックです。左の空欄に接続するアクセスポイントのSSIDを、右の空欄にパスワードを入力します。



Wi-Fiの接続を解除するためのブロックです。

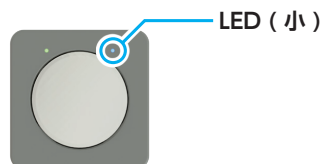


Wi-Fiに接続されているかどうかを判定するためのブロックです。

知識技能①

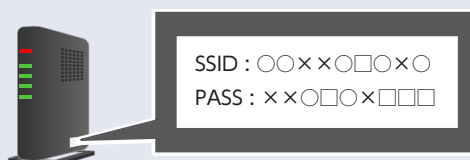
WiFi SSID: PASS: に接続 の左の空欄にアクセスポイントのSSIDを、右の空欄にパスワードを入力して、メインユニットをWi-Fiに接続しましょう。

Wi-Fiへ接続できていると、メインユニットのLED（小）が青色で光ります。



SSIDとは

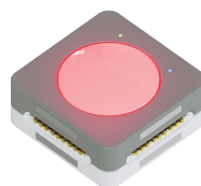
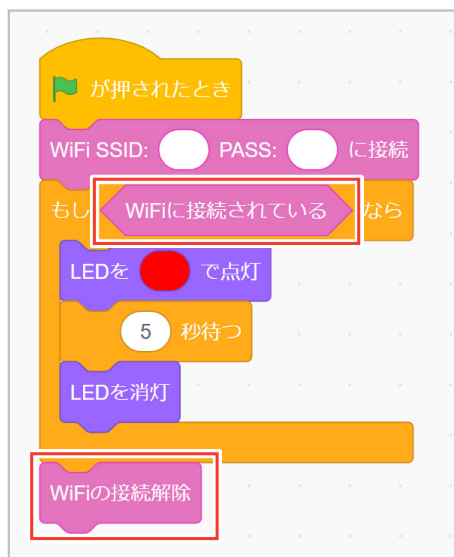
SSIDとはアクセスポイントに設定されている識別番号です。アクセスポイントと同じSSIDを設定した機器だけを認識して、Wi-Fiに接続させることができます。



※ 多くの機器が1つのアクセスポイントに同時接続していると、接続状況が不安定になることがあります。必要に応じて接続できているかどうかを確認したり、接続を解除したりしてください。

例 Wi-Fiへの接続を確認してからLEDを5秒間光らせたあと、Wi-Fiへの接続を解除するプログラム

※ Wi-FiのSSIDとPASSは、接続するアクセスポイントのものを入力してください。



基本課題 ①

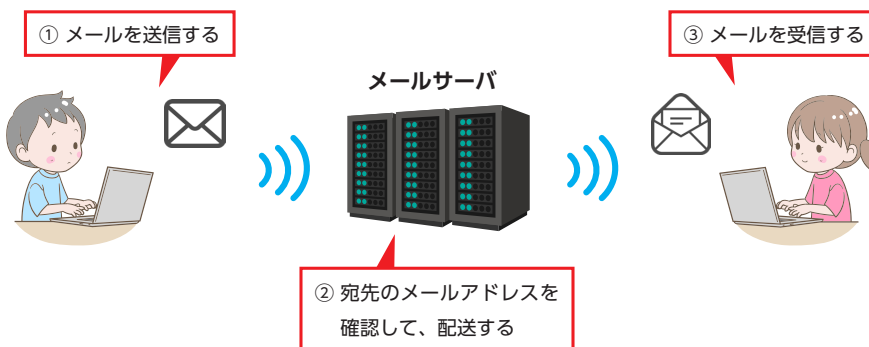
知識技能①

i サーバの仕組み

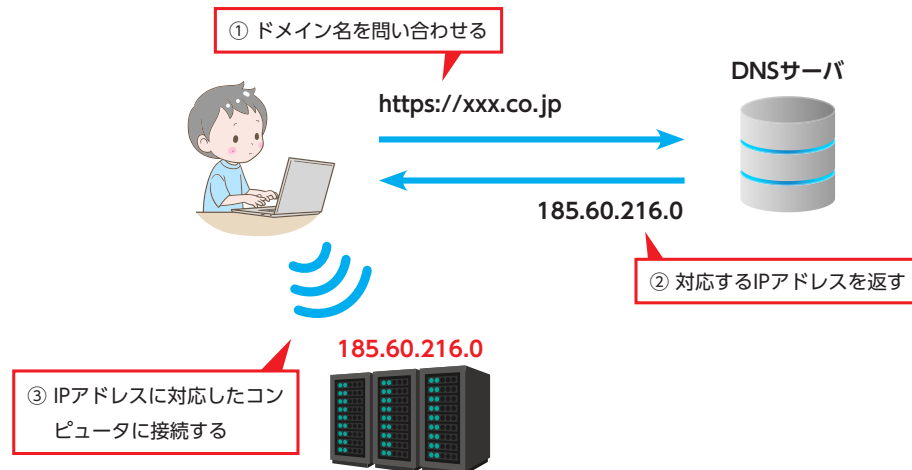
サーバ

サーバとは、ネットワークを通じてサービスを提供するコンピュータです。
たとえば、私たちがインターネットでWebページにアクセスしたとき、Webページに表示される画像や文章はWebページのデータを保管しているWebサーバから提供されています。
ネットワークには様々な種類のサーバが接続されており、メール通信サービスを提供するメールサーバや、データを保管するデータベースサーバなど、それぞれが特定の役割を担っています。

例 メールサーバを介したメールの送受信



サーバのようなネットワークに接続されている情報機器は、それぞれを識別するために「IPアドレス」と呼ばれる番号がつけられています。IPアドレスは人間にはわかりにくいので、私たちがコンピュータにアクセスするときは、IPアドレスに対応した「ドメイン名」や通信の方式を含んだURLを使います。ドメイン名は「DNSサーバ」というサーバでIPアドレスに変換されるため、私たちはIPアドレスを知らなくても目的のコンピュータにアクセスできます。



知識技能①

タイムサーバに接続しよう

Wi-Fiに接続しているメインユニットを使って、サーバに接続しましょう。
今回は全世界の時刻を保管しているタイムサーバに接続して、日本の現在時刻を取得します。

IoTブロックの使い方 ②

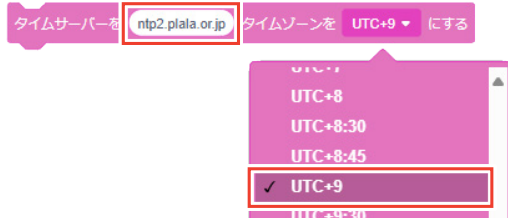
タイムサーバに接続するときは、「IoTブロック」のうち、以下のブロックを使用します。



タイムサーバとタイムゾーンを設定するためのブロックです。左の空欄にタイムサーバのURLを、右のプルダウンメニューからタイムゾーンを指定します。

→ タイムゾーンとは P.8

日本の現在時刻を取得したいときは、以下のように入力します。



時刻を更新する

タイムサーバから日時を取得するブロックです。ブロックを実行すると、
タイムサーバを タイムゾーンを UTC-12 にする で指定したタイムサーバから現在の日時を取得します。

※ タイムサーバには使用制限がかけられている場合があります。このブロックを短時間に何度も実行すると、タイムサーバに接続できなくなることがあるため注意してください。

年

このブロックには、現在の日時が保存されています。現在の日時は、時刻を更新する で取得した時点の日時を基準として、アーテックリンクス内のタイマーを使って計算しています。日時のデータは、年・月・日・時・分・秒の6つに分かれていて、プルダウンメニューから選択できます。現在の日時を確認したい場合は、以下のように変数に保存して使用します。



タイムサーバに接続できない場合は、次のようなエラーメッセージが表示されます。

app.artec-kk.co.jp の内容

```
Exception:
Traceback (most recent call last):
  File "/bin/testmode.py", line 24, in try_
    File "<string>", line 1, in <module>
    File "/bin/tmi_iot.py", line 61, in update_time
    File "/lib/al/hub/iot_block.py", line 126, in update_time
    File "ntptime.py", line 1, in time
OSError: [Errno 116] ETIMEDOUT
```

OK

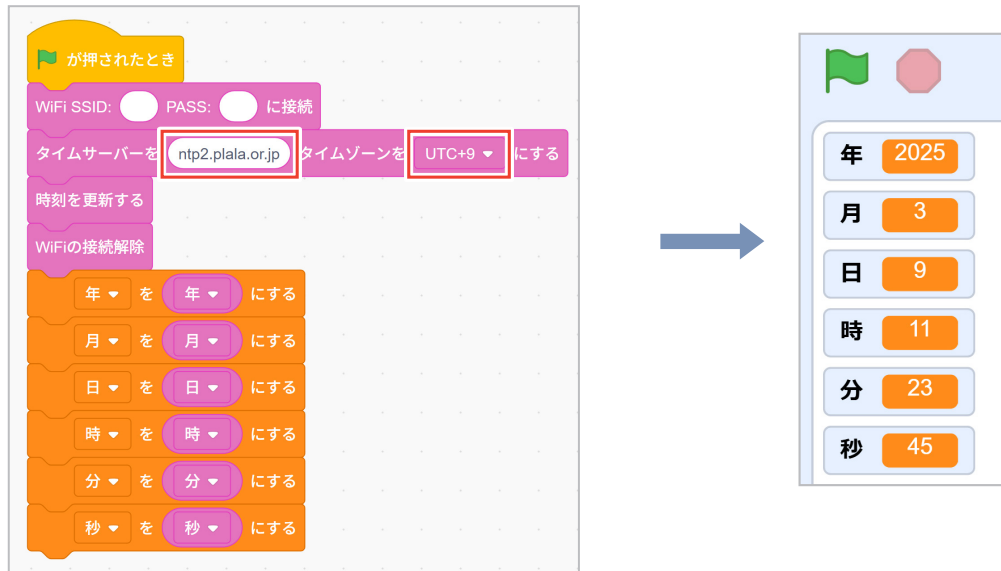
知識技能①

以下のプログラムを作成して、タイムサーバから日本の現在時刻を取得しましょう。

タイムサーバには「ntp2.plala.or.jp」を入力し、タイムゾーンはUTC+9に設定します。

※ Wi-FiのSSIDとPASSは、接続するアクセスポイントのものを入力してください。

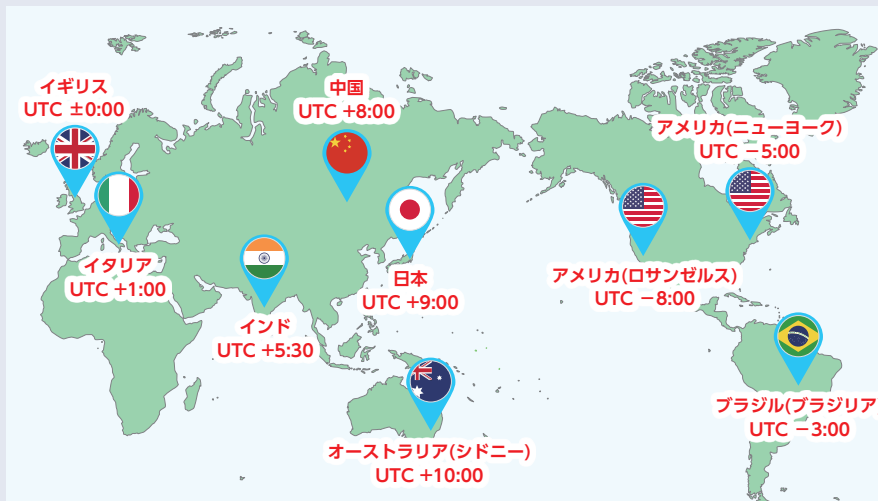
※ プログラム実行時にエラーが発生するときは、タイムサーバを「time.aws.com」や「ntp.nict.jp」など別のものに变更してください。



タイムゾーンとは

タイムゾーンとは、地球上で同じ標準時を採用している地域の集合のことです。地球の標準の時刻である協定世界時(UTC)は、セシウム原子時計を使って定められています。しかし、緯度が違えば時差が生じるため、各地域の標準時はUTCに緯度に応じた時間を足し引きして設定しています。

日本ではUTCに9時間を足した時刻が標準時になります。同じタイムゾーンの地域には韓国やインドネシア東部などがあります。



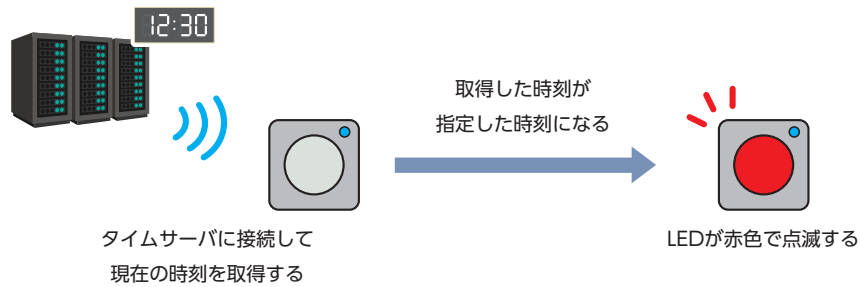
基本課題 ②

知識技能②

タイマー機能が付いた電波時計をつくろう

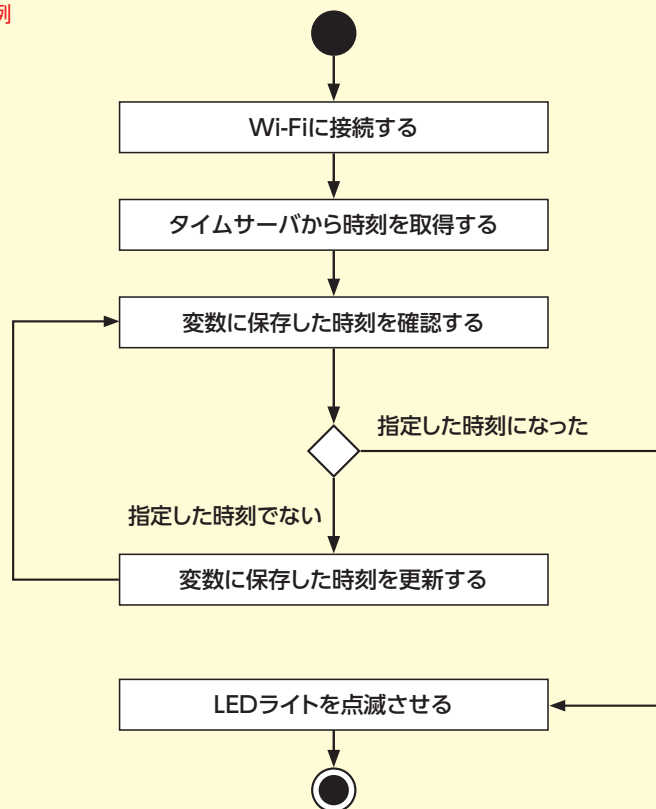
【課題】

タイムサーバから時刻を取得して、指定した時刻になったら赤色で点滅するタイマー機能付き電波時計を作成します。動作を確認できるように、指定する時刻は現在時刻に近い時刻を指定しましょう。



① プログラムの流れを考えて、アクティビティ図を作成しましょう。

アクティビティ図の例



知識技能②

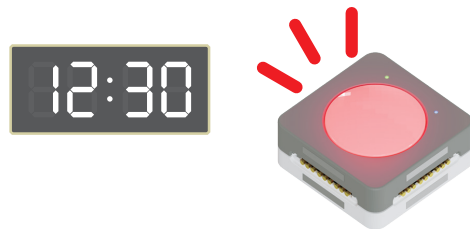
- ② P.9で作成したアクティビティ図を参考にして、プログラムを作成しましょう。
以下の使用ブロックを参考にしてください。

プログラ
ミング
実習

使用ブロック

	メインユニットのLEDを光らせるためのブロックです。
	メインユニットのLEDを消灯させるためのブロックです。
	が押されたときに、このブロックの下に組まれているプログラムが実行されます。 このようなブロックをハットブロックといい、ブロックの下側のみにブロックをつなげることができます。
	反復処理を行うためのブロックです。条件を満たすまで、ブロックに挟まれているプログラムを実行します。
	右の条件と左の条件の両方を満たすときだけ、全体の条件を満たしたことになるブロックです。
	右の値と左の値が「等しい」かどうか判定するためのブロックです。

- ③ プログラムが完成したら、プログラムを実行しましょう。
決まった時刻になったらLEDが赤色で点滅することを確認してください。



- ④ 達成できた項目にチェックをいれましょう。

- ☐ 双方向性のあるコンテンツについて理解できた。
- ☐ 情報通信ネットワークの仕組みを理解できた。
- ☐ メインユニットをWi-Fi、サーバに接続することができた。
- ☐ サーバから取得したデータを活用したプログラムを作成することができた。

プログラム例

が押されたとき

WiFi SSID: PASS: に接続

タイムサーバーを タイムゾーンを にする

時刻を更新する

WiFiの接続解除



5 回繰り返す

LEDを ☐ で点灯

1 秒待つ

LEDを消灯

1 秒待つ

はじめにWi-Fiとサーバに接続します。
タイムサーバとタイムゾーンは基本課題
①と同じものを使用してください。

指定した時刻になるまでタイムサーバから時刻を更新し続けます。設定する時刻の条件は演算ブロックを追加することで、細かく設定することができます。

時刻を保存するための変数は、設定した時刻の条件に合わせて変更・追加してください。

学習内容

- IoTブロック
- サーバから取得したレスポンスを読み取る
- これまでに学習した内容の応用、演習

達成目標

- サーバから取得したレスポンスを読み取り、必要な値だけを取り出す
- 自ら課題を見つけ、解決するための手段を講じて、実行する
- 情報通信ネットワークの仕組みについて理解する

活動内容	評価
基本課題 ①	
<ul style="list-style-type: none"> ● 気象情報サーバに接続しよう P.14 <ul style="list-style-type: none"> ・IoTブロックの使い方 ③ ・レスポンスを取得する ・レスポンスから指定した値を取り出す ① ● 熱中症アラートをつくろう P.19 <ul style="list-style-type: none"> ・アクティビティ図の作成 ・プログラムの作成 ・プログラムの実行 	<ul style="list-style-type: none"> 知識及び技能① 知識及び技能②
基本課題 ②	
<ul style="list-style-type: none"> ● 1週間の天気を予報しよう P.22 <ul style="list-style-type: none"> ・レスポンスから指定した値を取り出す ② ・天気予報システムの作成 	<ul style="list-style-type: none"> 知識及び技能①
発展課題	
<ul style="list-style-type: none"> ● サーバと連携したプログラムを考えよう P.27 <ul style="list-style-type: none"> ・アイデアを考える ・アクティビティ図、フローチャートの作成 ・プログラムの作成、実行 ・発表 	<ul style="list-style-type: none"> 知識及び技能② 思考・判断・表現 学向力・人間性
まとめ	
<ul style="list-style-type: none"> ● 情報通信ネットワークを復習しよう P.34 	<ul style="list-style-type: none"> 知識及び技能①

基本課題 ①

知識技能①

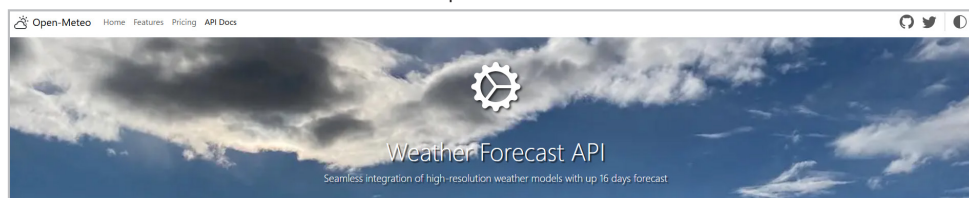
気象情報サーバに接続しよう

今回は「Open-Meteo」という気象情報を提供するWebサービスを使用します。Open-Meteoはユーザが設定した地域や期間に応じて、様々な気象情報をサーバから提供しています。

例として、現在の東京の気温を取得してみましょう。

- ① ブラウザで「<https://open-meteo.com/en/docs/>」を検索して、Webページにアクセスしましょう。

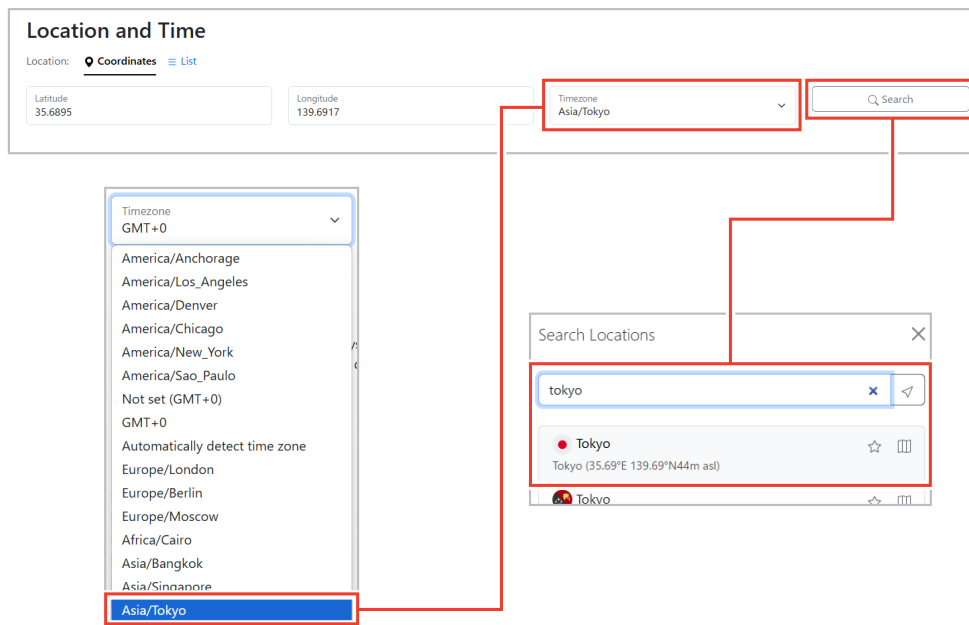
< Open-Meteo >



- ② 「Location and Time」で、タイムゾーンと地域、取得する期間を指定します。

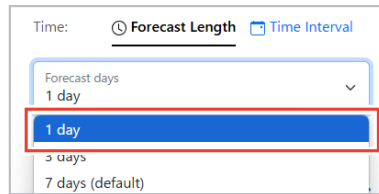
「Timezone」から「Asia / Tokyo」を選択し、「Search」から「Tokyo」を検索します。

※ 地域は「Latitude（緯度）」と「Longitude（経度）」を直接入力して指定することもできます。



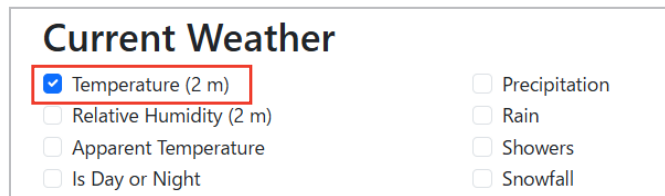
知識技能①

- ③ 「Time」を「7days」から「1day」に変えて、気象情報を取得する期間を1日（今日）に変更しましょう。



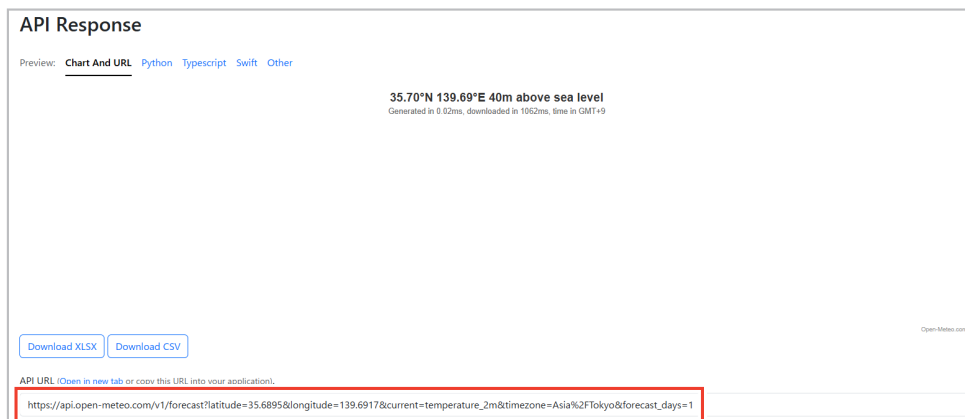
The screenshot shows a 'Time' dropdown menu with two tabs: 'Forecast Length' (active) and 'Time Interval'. Under 'Forecast Length', there are four options: '1 day', '3 days', '7 days (default)', and '7 days'. The '1 day' option is highlighted with a red box.

- ④ 「Current Weather」の「Temperature (2m)」にだけチェックをつけましょう。
他のデータは必要ないので、すべてのチェックを外しましょう。



The screenshot shows the 'Current Weather' section with a list of checkboxes. The 'Temperature (2 m)' checkbox is checked and highlighted with a red box. Other checkboxes include 'Relative Humidity (2 m)', 'Apparent Temperature', 'Is Day or Night', 'Precipitation', 'Rain', 'Showers', and 'Snowfall'.

- ⑤ ページの一番下にある「API Response」にURLが作成されています。
このURLにアクセスすると、指定した気象情報のデータを受け取ることができます。



The screenshot shows the 'API Response' section with a preview of the API response. The URL is displayed at the bottom and highlighted with a red box: `https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917¤t=temperature_2m&timezone=Asia%2FTokyo&forecast_days=1`. The preview also shows the location '35.70°N 139.69°E 40m above sea level' and the generation time 'Generated in 0.02ms, downloaded in 1062ms, time in GMT+9'.

IoTブロックの使い方 ③

タイムサーバ以外のサーバからデータを取得するときは、以下のブロックを使います。

- GET url:** このブロックは、指定したサーバからデータを取得するためのブロックです。空欄にサーバのURLを入力します。
- レスポンス** このブロックには、サーバから取得したデータ（レスポンス）が保存されています。レスポンスの内容は変数に保存して使います。
- JSON のキー** このブロックは、JSON形式のデータから指定した値を取り出すためのブロックです。左の空欄にJSON形式のデータを入力し、右側の空欄に指定した値に対応するキーを入力して、指定した値だけを取り出すことができます。
※ ブロックの使い方やJSONについては、以降のページで詳しく紹介します。

レスポンスの取得

- ① **GET url:** を実行して、サーバからレスポンスを取得します。
空欄には、12ページで取得したURLをコピー&ペーストしましょう。

GET url: `https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917¤t`

12ページで取得したURL

- ② 変数「Response」を作成して、取得したレスポンスを変数に保存しましょう。

Response を **レスポンス** にする



Response `{"latitude":35.7,"longitude":139.6875,"generationtime_ms":0.0159740447998,"time":"iso8601","interval":"seconds","tempe`

- ※ 変数に保存したレスポンスが見切れてしまうときは、URLをブラウザの検索タブにコピー&ペーストして「Enter」キーを押すと、レスポンスの内容を確認できます。

URL `https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917¤t=temperature_2m&timezone=Asia%2FTokyo&forecast_days=1`

```

1 {
2   "latitude": 35.7,
3   "longitude": 139.6875,
4   "generationtime_ms": 0.02300739288330078,
5   "utc_offset_seconds": 32400,
6   "timezone": "Asia/Tokyo",
7   "timezone_abbreviation": "JST",
8   "elevation": 40,
9   "current_units": {
10    "time": "iso8601",
11    "interval": "seconds",
12    "temperature_2m": "C"
13  },
14  "current": {
15    "time": "2024-12-17T09:15",
16    "interval": 900,
17    "temperature_2m": 6.6
18  }
19 }

```

レスポンス

知識技能①

サーバから取得したレスポンスは「JSON」という形式で書かれています。JSONでは、{ } の中にデータの種類を示す「キー」とデータの名前である「値」が「:」を挟んで対応しています。「キー」と「値」の対応は「,」ごとに区切られています。

例 JSON形式で書いた犬の情報

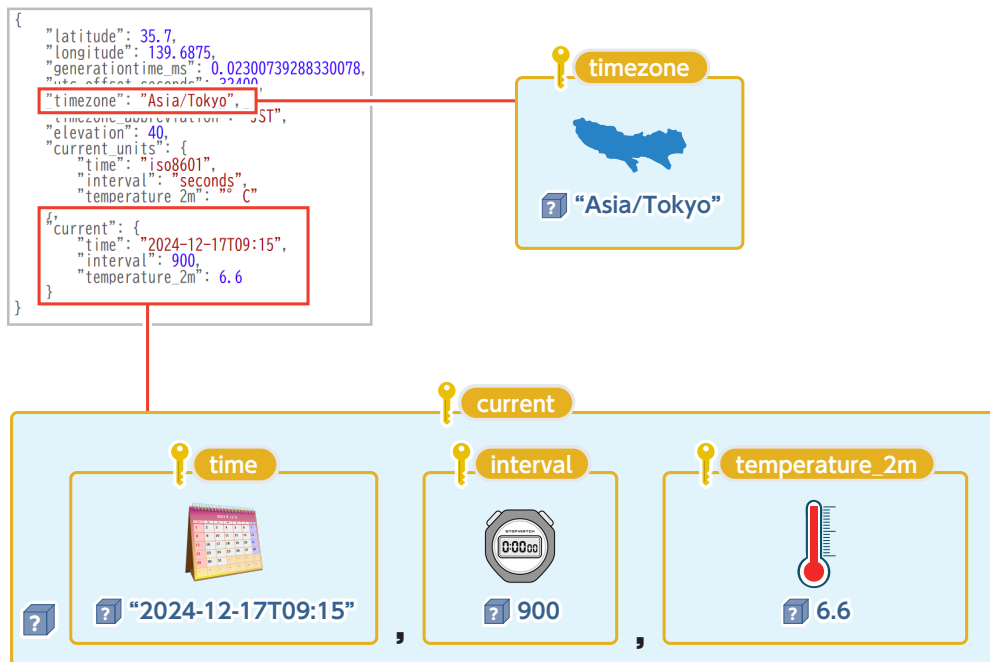


キー 値

```
{
  名前 : マロン ,
  年齢 : 2歳 ,
  性別 : ♂ ,
  犬種 : トイプードル
}
```

今回取得したレスポンスには、タイムゾーンを示す「timezone」や、現在の気象情報を示す「current」などのキーがあります。

今回欲しい気温の値は、「current」の値の中にある、「temperature_2m」に対応しています。



JSONとは

JSONとは「JavaScript Object Notation」の略で、JavaScriptと呼ばれるプログラミング言語の書き方を利用してデータを書いています。JSONはデータの構造が簡単で容量が少ないため、JavaScriptで書かれていないプログラムでもデータの受け渡しに使われています。

知識技能①

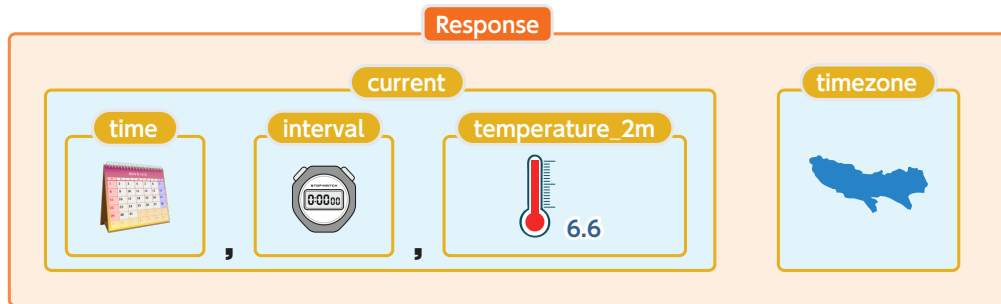
レスポンスから指定した値を取り出す ①

レスポンスを保存した変数「Response」から気温の値だけを取り出して、変数「temp」に保存しましょう。
指定した値だけを取り出すときは、左の空欄にJSON形式のデータを、右の空欄に対応するキーを入力します。

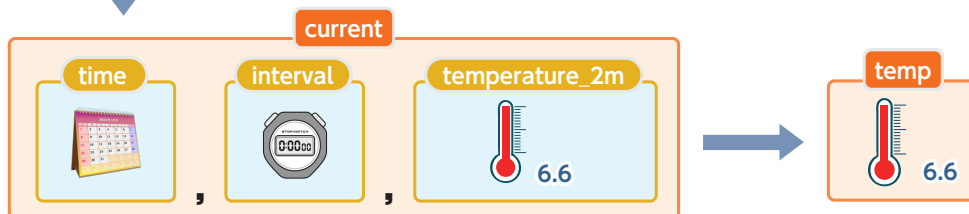


今回取り出す気温の値に対応するキーは "temperature_2m" ですが、"temperature_2m" は "current" の値に含まれています。

このような場合は2回に分けて、大きい値から順番に取り出しましょう。



- ① 変数「Response」から "current" の値を取り出して、変数「current」に保存する



- ② 変数「current」から "temperature_2m" の値を取り出して、変数「temp」に保存する



①と②を順番に実行することで、変数「temp」に現在の東京の気温を保存できます。

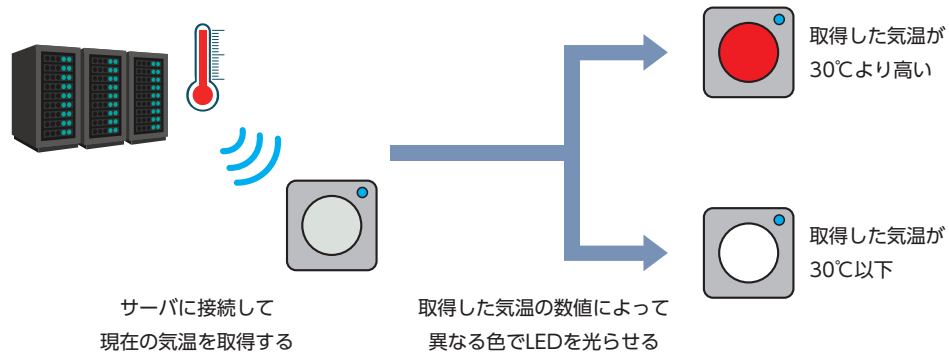


知識技能②

熱中症アラートをつくろう

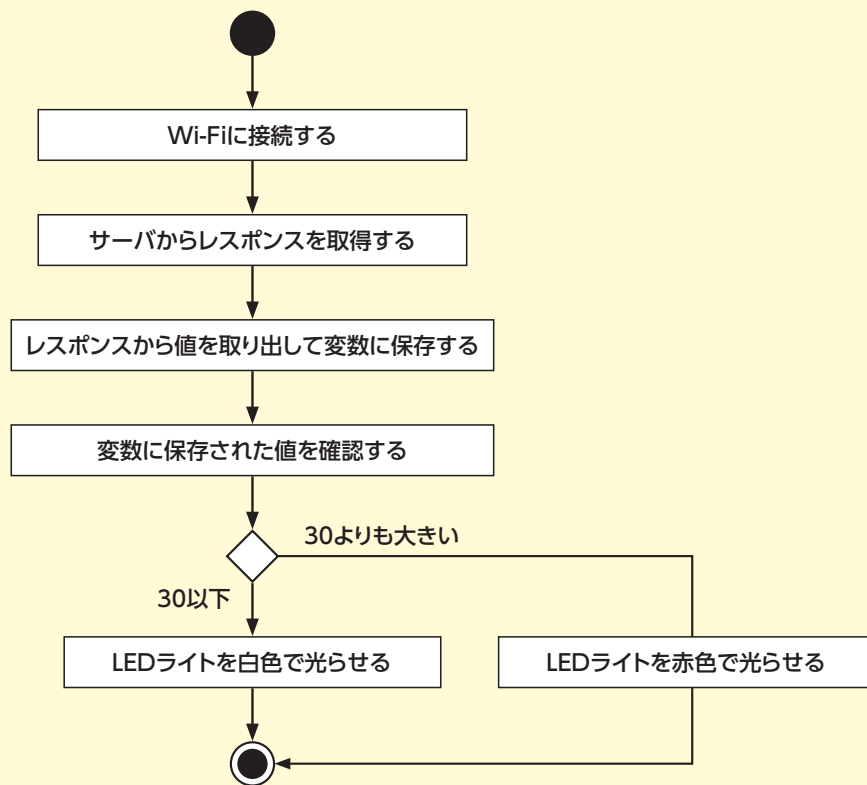
【課題】

取得した気象情報を使って、気温が高くなっていることをLEDで知らせるプログラムを作成します。
気温が30℃より高い場合は赤色でLEDを光らせ、それ以外の場合は白色でLEDを光らせてください。



① プログラムの流れを考えて、アクティビティ図を作成しましょう。

アクティビティ図の例



- ② P.16で作成したアクティビティ図を参考にして、プログラムを作成しましょう。……以下の使用ブロックを参考にしてください。

使用ブロック



メインユニットのLEDを光らせるためのブロックです。



メインユニットのLEDを消灯させるためのブロックです。



[] が押されたときに、このブロックの下に組まれているプログラムが実行されます。このようなブロックをハットブロックといい、ブロックの下側のみにブロックをつなげることができます。



次の処理の実行を遅らせるためのブロックです。指定した秒数だけ、プログラムをストップします。



条件分岐を行うためのブロックです。条件が満たされた場合は、ブロックの上側に挟まれているプログラムを実行します。満たされていない場合は、下側に挟まれているプログラムを実行します。



左の値が右の値より「大きい」かどうかを判定するためのブロックです。

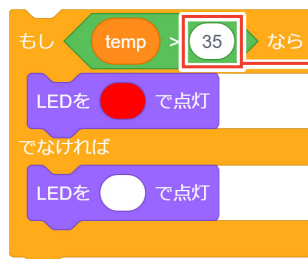
- ③ プログラムが完成したら、プログラムを実行しましょう。

取得した気温が30℃より高いときは赤色で、30℃以下のときは白色でLEDが光ることを確認してください。



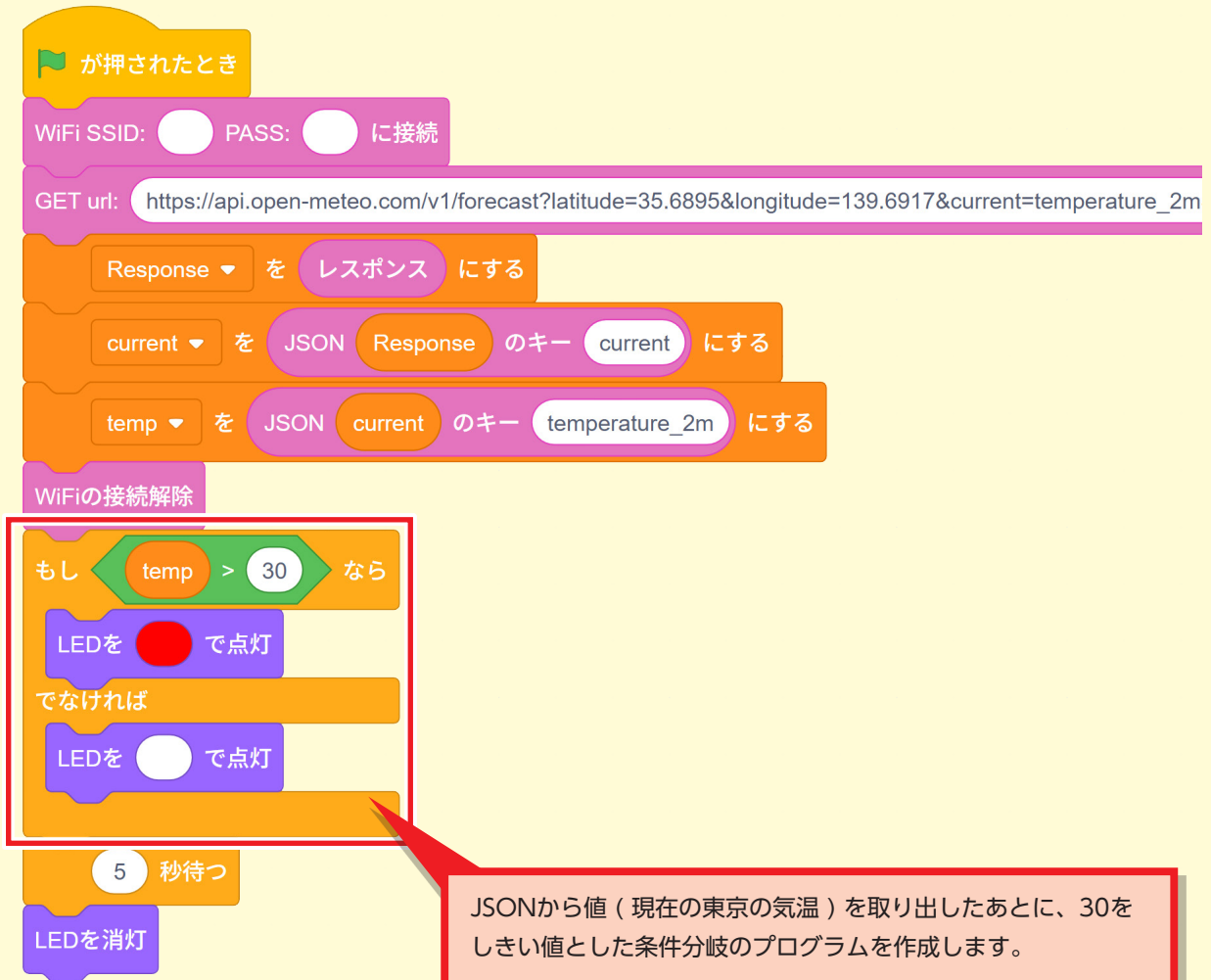
※ しきい値を取得した気温より±1℃の数値に変更して、他の色で光ることも確認しましょう。

例 取得した気温が34℃の場合（30℃より高い場合）



取得した気温より高い数値を入力して、LEDが白色で光ることも確認する

プログラム例



基本課題 ②

知識技能①

1週間の天気を予報しよう

Open-Meteoでは取得気象情報を自由に設定できます。
例として一週間の天気を取得して、LEDでお知らせする天気予報システムを作成しましょう。



- ① 気象情報を取得する期間を1週間に変更します。
「Time」を「1day」から「7days」に変えましょう。

Time:
Forecast Length
Time Interval

Forecast days
7 days (default)

Past days
0 (default)

- ② 取得する気象情報を1日ごとの天気（ウェザーコード）に変更します。
「Daily Weather Variables」の「Weather code」にチェックをいれて、1日ごとの天気を取得しましょう。
他のデータは必要ないので、すべてのチェックを外しましょう。







Daily Weather Variables

- ☒ Weather code
- ☐ Maximum Temperature (2 m)
- ☐ Minimum Temperature (2 m)

ウェザーコードとは

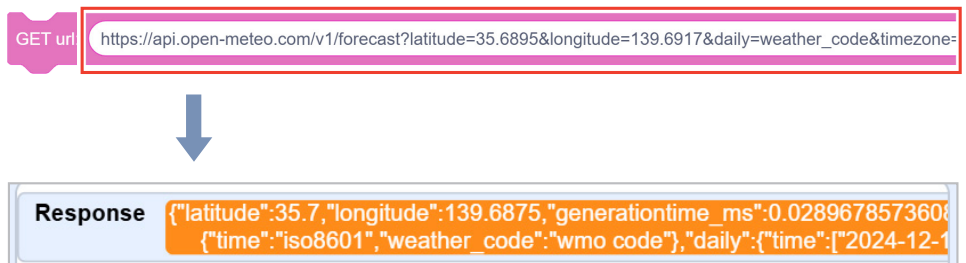
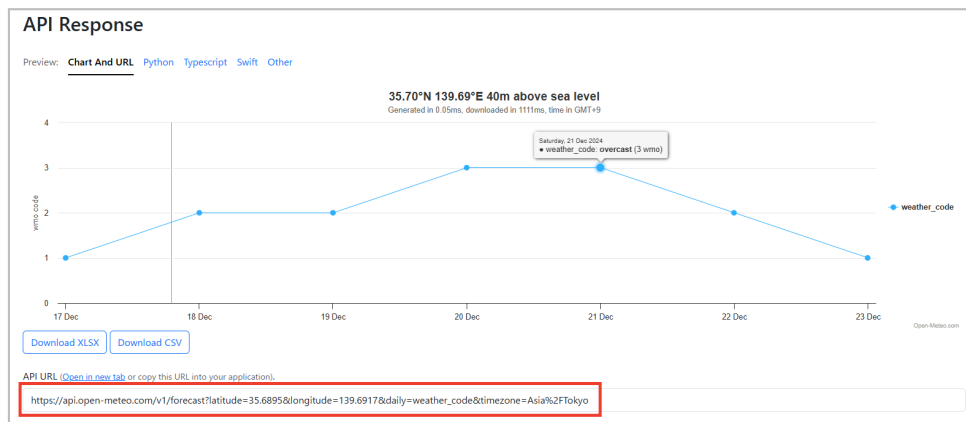
Open-Meteoでは、各地の天気を0~99の番号で表した「ウェザーコード」を提供しています。ウェザーコードは世界気象機関が定めた世界共通の番号で、言語や文化の異なる地域にいても、正確な天気を伝えることができます。

<ウェザーコードの例>

ウェザーコード	0	1,2	3	63	73	95
天気	 快晴	 晴れ	 くもり	 雨	 雪	 雷雨

知識技能①

- ③ 取得する情報を変更すると、ページの一番下にある「API Response」のURLが更新されています。
このURLをコピー&ペーストして、レスポンスを取得しましょう。



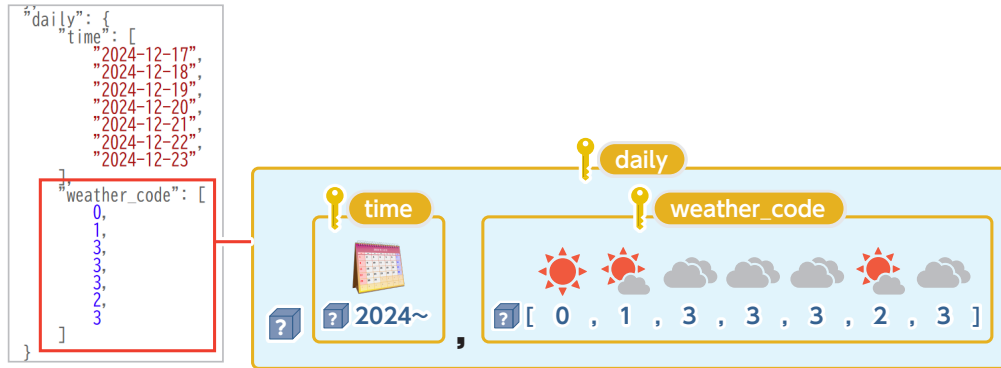
- 例 2024年12月17日から1週間の天気(ブラウザで表示した場合)

```
1 {
2   "latitude": 35.7,
3   "longitude": 139.6875,
4   "generationtime_ms": 0.04303455352783203,
5   "utc_offset_seconds": 32400,
6   "timezone": "Asia/Tokyo",
7   "timezone_abbreviation": "JST",
8   "elevation": 40,
9   "daily_units": {
10    "time": "iso8601",
11    "weather_code": "wmo code"
12  },
13  "daily": {
14    "time": [
15      "2024-12-17",
16      "2024-12-18",
17      "2024-12-19",
18      "2024-12-20",
19      "2024-12-21",
20      "2024-12-22",
21      "2024-12-23"
22    ],
23    "weather_code": [
24      0,
25      1,
26      3,
27      3,
28      3,
29      2,
30      3
31    ]
32  }
33 }
```

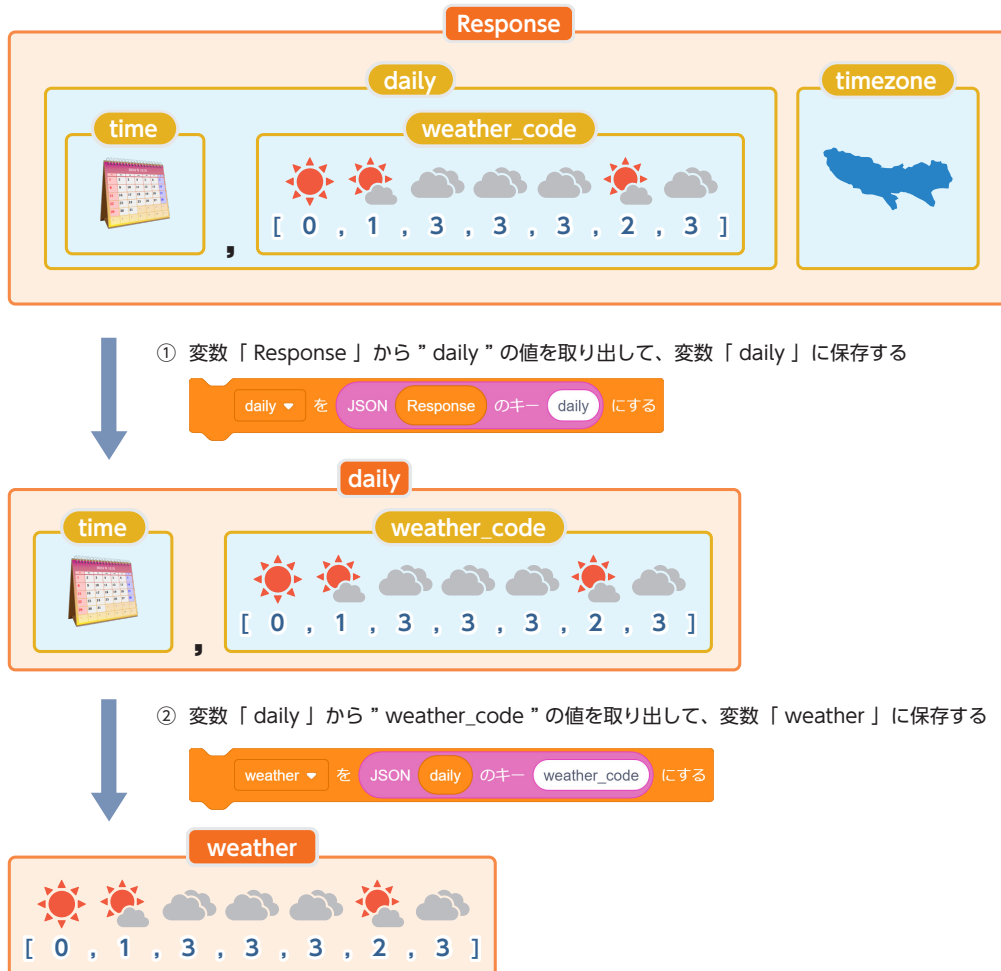

知識技能①

レスポンスから指定した値を取り出す ②

今回取得したレスポンスでは、1週間のウェザーコードは "weather_code" に対応していて、"weather_code" は "daily" の中に書かれています。



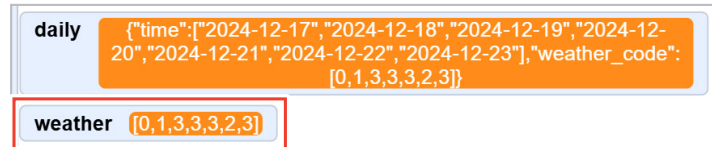
以下の順番で値を取り出して、"weather_code" の値を変数「weather」に保存しましょう。



天気予報システムの作成

変数「weather」には、1週間のウェザーコードが1つの値として保存されています。

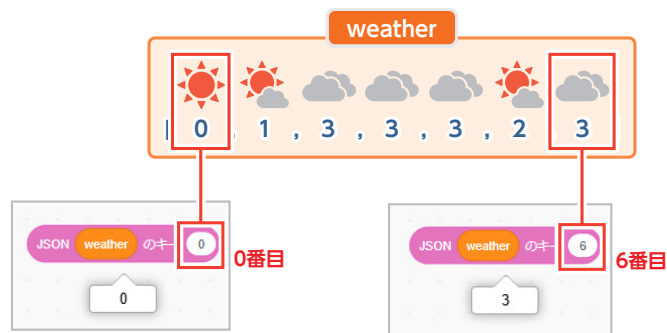
そのため、プログラムでウェザーコードを1日ずつ使うことはできず、数値として扱うこともできません。



ウェザーコードを1日ずつに分けて使う場合は、値を1つずつ取り出す必要があります。

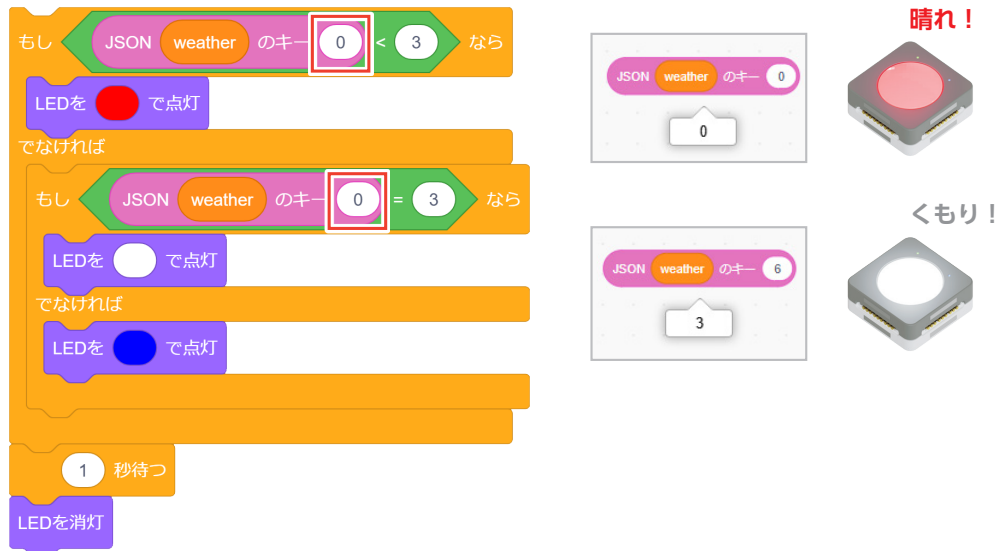
取得したウェザーコードはJSON形式になっているため、キーの代わりに順番を入力すると、順番に対応したウェザーコードの値を取り出すことができます。

※ JSONのキーは、リストと同じように「0」から順番に数えます。



ウェザーコードが0～2の場合は晴れ、3の場合はくもり、それ以外の場合を雨として、それぞれ異なる色でLEDを光らせるプログラムを実行すると、天気を予報するシステムを作成できます。

以下のプログラムで、キーに「0」～「6」の値を入力して日にちを指定すると、その日にちの天気に対応した色でLEDが点灯します。



知識技能①

今回は1週間（7日）のウェザーコードを取得しているため、キーを1ずつ変えて7回繰り返すと、1週間の天気をお知らせすることができます。

キーの数値を指定する変数「i」を作成して、天気を知らせた後に「i」を1ずつ変えるプログラムを、7回繰り返しましょう。

プログラムのはじめで、変数「i」を「0」に戻す

キーに変数「i」を挿入する

LEDを消灯した後に、変数「i」を「1」ずつ変える

例 プログラムの完成例（Wi-FiのSSIDとPASSは、接続するアクセスポイントのものを入力してください。）

発展課題

思考・判断・表現

学向力・人間性

サーバと連携したプログラムを考えよう

サーバから取得できる情報を活用して、私たちの生活に役立つプログラムを作成します。
取得する気象情報を変更したり、タイムサーバとも連携して情報を組み合わせたりするなど、どのような工夫ができるか考えましょう。



- ① サーバから取得する情報を活用してどのようなことができるか、自由にプログラムのアイデアを考えましょう。
また、そのプログラムを使うとどのような課題を解決することができるか、下の表に書き入れましょう。

アイデア ①	決まった時刻になったら、天気予報システムを実行する。
解決できる課題	忙しい朝でも、忘れずに天気予報を確認できる。
アイデア ②	帰宅する時刻の降水確率をLEDで表示する。
解決できる課題	出かける前に、傘を持っていく必要があるかどうか分かる。
アイデア ③	気温と湿度から不快指数を計算して、LEDで表示する。
解決できる課題	温度計だけではわからない暑さを、目に見える形で表すことができる。

知識技能②

- ② ①で考えたアイデアの中から1つ選び、プログラムの概要を文章やイラストを使ってまとめましょう。……また、プログラムを作成するうえで必要なデータや接続するサーバをまとめ、フローチャートやアクティビティ図を書いて、プログラムを作成しましょう。

プログラ
ミング
実習

プログラムの概要	
必要なデータ	
接続するサーバ	
フローチャートや アクティビティ図	

③ ②で作成したプログラムをみんなに発表しましょう。

【プログラム名】

【プログラムの概要】

【どのような課題を解決するか】

【工夫】



【今後解決したい改善点】

プログラム例① 決まった時刻になったら、天気予報システムを実行する。

が押されたとき

WiFi SSID: PASS: に接続

時 を 0 にする

タイムサーバーを ntp2.plala.or.jp タイムゾーンを UTC+9 にする

時刻を更新する

WiFiの接続解除

時 = 7 まで繰り返す

時 を 時 にする

60 秒待つ

WiFi SSID: PASS: に接続

GET url: https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917&daily=weather_code&timezone=

Response を レスポンス にする

daily を JSON Response のキー daily にする

weather を JSON daily のキー weather_code にする

i を 0 にする

WiFiの接続解除

7 回繰り返す

もし JSON weather のキー i < 3 なら

LEDを 赤 で点灯

でなければ

もし JSON weather のキー i = 3 なら

LEDを 白 で点灯

でなければ

LEDを 青 で点灯

1 秒待つ

LEDを消灯

i を 1 ずつ変える

はじめにタイムサーバに接続して時刻を更新し、指定した時刻（例では7時）になるまで待ちます。

天気予報のプログラム、URLは基本課題②と同じものを使います。

プログラム例② 帰宅する時刻の降水確率をLEDで表示する。

「Hourly Weather Variables」の「Precipitation Probability」にだけチェックをいれて、URLを更新してください。

【更新されたURL】

【https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917&hourly=precipitation_probability&timezone=Asia%2FTokyo&forecast_days=1】

Hourly Weather Variables

- ☐ Temperature (2 m)
- ☐ Relative Humidity (2 m)
- ☐ Dewpoint (2 m)
- ☐ Apparent Temperature
- ☒ Precipitation Probability
- ☐ Precipitation (rain + showers + snow)

が押されたとき

WiFi SSID: PASS: に接続

GET url: https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917&hourly=precipitation_

Response を レスポンス にする

hourly を JSON Response のキー hourly にする

precipitation を JSON hourly のキー precipitation_probability にする

WiFiの接続解除

もし JSON precipitation のキー 16 > 50 なら

LEDを  で点灯

でなければ

もし JSON precipitation のキー 16 > 25 なら

LEDを  で点灯

でなければ

LEDを  で点灯

1 秒待つ

LEDを消灯

1時間ごとの降水確率を取得します。
値に対応するキーは "hourly" の中の
"precipitation_probability" です。

取得した降水確率によって、条件分岐するプログラムを作成します。
例では、

- ・ 50%を超えている → 青（傘が必要）
- ・ 50%以下だが、25%を超えている → 水色（折り畳み傘が必要）
- ・ 25%以下 → 白色（傘は必要ない）

としています。

※ 表示させる時間帯は、キーの数値に時刻を入力して指定します。

例では16時の降水確率を表示させています。

プログラム例③ 気温と湿度から不快指数を計算して、LED で表示する。

「Current Weather」の「Temperature (2m)」と「Relative Humidity (2m)」にだけチェックをいれて、URLを更新してください。

【更新されたURL】

`https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917¤t=temperature_2m,relative_humidity_2m&timezone=Asia%2FTokyo&forecast_days=1`

Current Weather

- ☒ Temperature (2 m)
- ☒ Relative Humidity (2 m)
- ☐ Apparent Temperature
- ☐ Is Day or Night

が押されたとき

WiFi SSID: PASS: に接続

GET url: `https://api.open-meteo.com/v1/forecast?latitude=35.6895&longitude=139.6917¤t=temperature_2m,relative_humidity_2m`

Response を レスポンス にする

current を JSON Response のキー current にする

temp を JSON current のキー temperature_2m にする

humi を JSON current のキー relative_humidity_2m にする

現在の気温と湿度を取得します。
値に対応するキーは "current" の中の "temperature_2m" と "relative_humidity_2m" です。

WiFiの接続解除

DiscomfortIndex を $0.81 * \text{temp} + 0.01 * \text{humi} * (0.99 * \text{temp} - 14.3) + 46.3$ にする

不快指数を算出します。計算式は以下の通りです。
 $0.81 \times \text{気温} + 0.01 \times \text{湿度} \times (0.99 \times \text{湿度} - 14.3) + 46.3$
※ 演算ブロックを入れる順番に注意してください。

もし DiscomfortIndex > 85 なら

LEDを ● で点灯

でなければ

もし DiscomfortIndex > 75 なら

LEDを ● で点灯

でなければ

もし DiscomfortIndex > 60 なら

LEDを ● で点灯

でなければ

LEDを ● で点灯

算出した不快指数によって条件分岐するプログラムを作成します。
例では、

- ・ 85を超えている → 青 (非常に暑い)
- ・ 85以下だが、75を超えている → 黄色 (暑い)
- ・ 75以下、60以上 → 緑 (快適)
- ・ 60未満 → 水色 (寒い)

としています。

1 秒待つ

LEDを消灯

プログラム例 ②

```
hourly": {
  "time": [
    "2024-12-20T00:00",
    "2024-12-20T01:00",
    "2024-12-20T02:00",
    "2024-12-20T03:00",
    "2024-12-20T04:00",
    "2024-12-20T05:00",
    "2024-12-20T06:00",
    "2024-12-20T07:00",
    "2024-12-20T08:00",
    "2024-12-20T09:00",
    "2024-12-20T10:00",
    "2024-12-20T11:00",
    "2024-12-20T12:00",
    "2024-12-20T13:00",
    "2024-12-20T14:00",
    "2024-12-20T15:00",
    "2024-12-20T16:00",
    "2024-12-20T17:00",
    "2024-12-20T18:00",
    "2024-12-20T19:00",
    "2024-12-20T20:00",
    "2024-12-20T21:00",
    "2024-12-20T22:00",
    "2024-12-20T23:00"
  ]
}
```

[illegible]

プログラム例 ③

```
"current": {
  "time": "2024-12-20T16:00",
  "interval": 900,
  "temperature_2m": 10.6,
  "relative_humidity_2m": 59
}
```

まとめ

知識技能①

情報通信ネットワークを復習しよう

- ① 文字、画像、音声、動画などを組み合わせて、意味のある情報としてあらわされたものを何といいますか？

➡ **コンテンツ**

- ② LANなどの世界中のネットワークどうしをルータでつなげた大規模なネットワークのことを何といいますか？

➡ **インターネット**

- ③ あらゆるモノを無線通信で②に接続する技術を何といいますか？

➡ **IoT (Internet of Things)**

- ④ ネットワークに接続されている情報機器を識別するための番号のことを何といいますか？

➡ **IPアドレス**

- ⑤ ドメイン名を④に変換するサーバを何といいますか？

➡ **DNSサーバ**



**双方向性のあるコンテンツ
教員用**

テキストに関するお問い合わせ

株式会社 **アーテック** お客様相談窓口



◀Webからのお問い合わせはこちら
<https://www.artec-kk.co.jp/contact/>

お電話でのお問い合わせはこちら
TEL 072-990-5656