

音楽を交換しよう

ネットワーク機能を使って、教室の中で友だちと音楽データを交換するプログラムを作ってみよう。

音楽を演奏する

`chmusic` で扱った音楽演奏を思い出してみよう。ドリトルでは、「ドレミー」のようにメロディを書いて音楽を演奏できる。次のプログラムでは、メロディを入力する「曲」というフィールドオブジェクトを作り、**リターンキー**を押したときにメロディオブジェクトにフィールドに書かれた旋律を追加して演奏する。



```
曲 = フィールド!作る 600 45 大きさ - 250 100 位置。  
曲:動作 = 「メロディ!作る (自分!読む)追加 演奏」。
```

音楽を送信する

入力した音楽をサーバーに書き込めるようにする。次のプログラムでは、「送信」ボタンを押したときに、曲名とメロディをサーバーに書き込む。



```
サーバ"localhost"接続。  
曲 = フィールド!作る 600 45 大きさ - 250 100 位置。  
曲:動作 = 「メロディ!作る (自分!読む)追加 演奏」。  
曲名 = フィールド!作る 130 45 大きさ - 380 100 位置。  
送信ボタン = ボタン!"送信" 作る - 350 50 位置。  
送信ボタン:動作 = 「サーバー!(曲名!読む)(曲!読む)書く」。
```

実行すると、サーバーにメロディの文字列が書き込まれる。

音楽を受信できるようにする

サーバーから音楽を受信して演奏できるようにする。次のプログラムでは、曲名を入力して「受信」ボタンを押したときに、サーバーからその曲のメロディをダウンロードして「曲」フィールドに表示する。曲フィールドでリターンキーを押すことで、ダウンロードした曲が演奏される。このように、ひとつのフィールドを送信と受信のために共通に使うこともできる。



```
サーバ"localhost"接続。  
曲 = フィールド!作る 600 45 大きさ - 250 100 位置。  
曲:動作 = 「メロディ!作る (自分!読む)追加 演奏」。  
曲名 = フィールド!作る 130 45 大きさ - 380 100 位置。  
送信ボタン = ボタン!"送信" 作る - 380 50 位置。  
送信ボタン:動作 = 「サーバー!(曲名!読む)(曲!読む)書く」。  
受信ボタン = ボタン!"受信" 作る - 230 50 位置。
```

受信ボタン：動作 = 「曲！（サーバー！（曲名！読む）読む）書く」。

ネットワークで接続された複数台のコンピュータで動作を確認する場合には、次のようにする。

- 複数台のコンピュータでドリトルを起動する。2台以上であれば何台でも構わない。
- それぞれのドリトルに上のプログラムを入力する。
- どれか1台のドリトルでサーバーを起動する。他のドリトルでは、サーバーを「終了」ボタンで閉じておくとよい。
- プログラムの「localhost」の部分にサーバーを実行しているコンピュータ名を記述する。コンピュータ名が分からない場合は、サーバーのIPアドレスをsec[ipaddress]の手順で調べて記述する。
- 曲を入力し、分かりやすい名前を付けてサーバーに書き込む。
- ドリトルの編集画面から「サーバー」ボタンを押してサーバー上のオブジェクトを確認する。自分の書き込んだ曲や、他の人が書き込んだ曲を一覧できる。確認したら「了解」ボタンでダイアログを閉じる。
- プログラムの実行画面で、他の人の曲名を入力してから「受信」ボタンを押して受信し、表示された曲のフィールドでリターンキーを押して、ダウンロードした曲を演奏する。これは音楽や映像をインターネットのサーバーに書き込み、それをいろいろな人が視聴することに相当する。
- なお、受信時に曲が見つからないときは、メロディの欄に「値が存在しない」という意味の未定義オブジェクトである「[undef]」が表示される。

長い曲を交換（ダウンロードとストリーミング）

携帯電話やパソコンでは、インターネットからダウンロードした音楽を聴くことができる。また、ダウンロードせずにストリーミングとして聴いていくこともできる。ここでは、ドリトルで作った音楽を、ダウンロードとストリーミングという2種類の方法で聴くためのプログラムを作成する。授業で学習する場合には、先生など1人が送信プログラムを実行し、他の生徒は受信プログラムを実行する。

送信プログラム

次のプログラムは、音楽をサーバーに送信するプログラムである。送信する曲は、小節ごとに6個に分けて「曲」という配列に格納している。「送信」ボタンを押すと、最初に曲の行数（6行）と送信間隔（4秒）をサーバーに送信した後、曲を一定間隔で1行ずつサーバーに送信する。実行は次の節の受信プログラムと合わせて行う。

```
サーバ"localhost"接続。
曲 = 配列！作る。
曲！"どどそそらそー"書く。
曲！"ふぁふぁみみれれどー"書く。
曲！"そそふぁふぁみみれー"書く。
曲！"そそふぁふぁみみれー"書く。
曲！"どどそそらそー"書く。
曲！"ふぁふぁみみれれどー"書く。
送信間隔 = 4。
```

```
送信中 = ラベル！作る 300 45 大きさ。
送信ボタン = ボタン！"送信" 作る。
送信ボタン：動作 = 「
    行数 = 曲！要素数？。
    送信中！（行数）書く。
    サーバー！"配信行数"（行数）書く。
    サーバー！"配信間隔"（送信間隔）書く。
    タイマー！作る（送信間隔）間隔（行数）回数「 | 番号 |
        メッセージ = 曲！（番号）読む。
```

```

    送信中！（メッセージ）書く。
    サーバー！"配信"（メッセージ）書く。
  」実行。

```

```

[]

```

受信プログラム

次のプログラムは、音楽をサーバーから受信するプログラムである。「ダウンロード」ボタンまたは「ストリーミング」ボタンを押すと、最初に曲の行数と受信間隔を受信した後、曲を一定間隔で1行ずつ受信する。ダウンロードボタンを押したときは、曲全体を受信してから演奏する。ストリーミングボタンを押したときは、曲の一部を受信しながら随時演奏していく。

実行する際は2つのドリトルの画面で、送信プログラムと受信プログラムを実行する。そして、送信プログラムの「送信」ボタンを押した直後に受信プログラムの「ダウンロード」または「ストリーミング」のボタンを押す。

```

サーバ[]"localhost"[]接続。

```

```

ダウンロード=ボタン！"ダウンロード"作る 230 45 大きさ -300 50 位置。

```

```

ストリーミング=ボタン！"ストリーミング"作る 230 45 大きさ -300 0 位置。

```

```

曲リスト=リスト！作る 200 400 大きさ 0 200 位置。

```

```

ダウンロード：動作 = 「

```

```

  曲 = ""。

```

```

  曲リスト！クリア。

```

```

  行数 = サーバー！"配信行数" 読む。

```

```

  受信間隔 = サーバー！"配信間隔" 読む。

```

```

  ラベル！（行数）作る -50 200 位置。

```

```

  時計 = タイマー！作る（受信間隔）間隔（行数）回数「 | 番号 |

```

```

    メッセージ = サーバー！"配信" 読む。

```

```

    曲リスト！（メッセージ）書く。

```

```

    曲 = 曲！（メッセージ）連結。

```

```

  」実行。

```

```

  時計！「メロディ！作る（曲）追加 演奏」最後に実行。

```

```

[]

```

```

ストリーミング：動作 = 「

```

```

  曲リスト！クリア。

```

```

  行数 = サーバー！"配信行数" 読む。

```

```

  受信間隔 = サーバー！"配信間隔" 読む。

```

```

  ラベル！（行数）作る -50 200 位置。

```

```

  時計 = タイマー！作る（受信間隔）間隔（行数）回数「 | 番号 |

```

```

    メッセージ = サーバー！"配信" 読む。

```

```

    曲リスト！（メッセージ）書く。

```

```

    メロディ！作る（メッセージ）追加 演奏。

```

```

  」実行。

```

```

[]

```

From:

<https://dolittle.eplang.jp/> - プログラミング言語「ドリトル」

Permanent link:

https://dolittle.eplang.jp/ch_nw_music



Last update: 2018/02/09 11:05

