YouTubeにおけるランキング表の出力

1. プログラムの内容

調べたいキーワードを入力し、そのキーワードに関連したYouTube動画の情報を取得し、ランキング表を出力する。

- 「検索キーワード」に調べたいキーワードを入力する。
- 「検索タイプ」にランキングを作成したい項目を選択する。
  - 「キーワードの関連性」、「投稿日」、「再生回数」、「高評価数」のいずれかを選択する。
    - キーワードの関連性: キーワードに最も関連した動画を検索する。
    - 投稿日: キーワードに関連し、最近アップロードされた動画を検索する。
    - 再生回数: キーワードに関連し、再生回数が多い動画を検索する。
    - 高評価数: キーワードに関連し、高評価数が多い動画を検索する。
- 「件数」に、ランキングの表示件数を選択する。
- 検索を実行すると、キーワードに関連した動画情報を取得されるので、その動画情報をデータベースに登録したのち、ブラウザにランキング表を表示する。

2. プログラムの動作

YouTube Data API v3 を用いて、ユーザーが入力したキーワードに関連したYouTube動画の情報をデータベースに登録し、ブラウザに表を出力する。

- 検索パラメータの入力は YouTube_API_Search.php で処理する。
- YouTube_API_Result.php において、ユーザーが表示件数を指定した分だけYouTube Data API v3からデータを取得し、データベースに登録する。
- YouTube_API_Result.php において、APIから「サムネイルのファイルパス」「タイトル」「投稿日」「再生回数」「高評価数」「動画ID」を取得する。
  - サムネイルのファイルパス: ファイルパスをimgタグに記載することで、サムネイル画像が表示される。
- ランキング表を出力するため、YouTube_API_Result.php でYouTube_API データベースのtbl_video_infoテーブルにデータを登録する際、順位も一緒に登録をしている。

3. Webページのデザイン

図1:検索パラメータの入力画面

検索条件を決めるための選択画面。

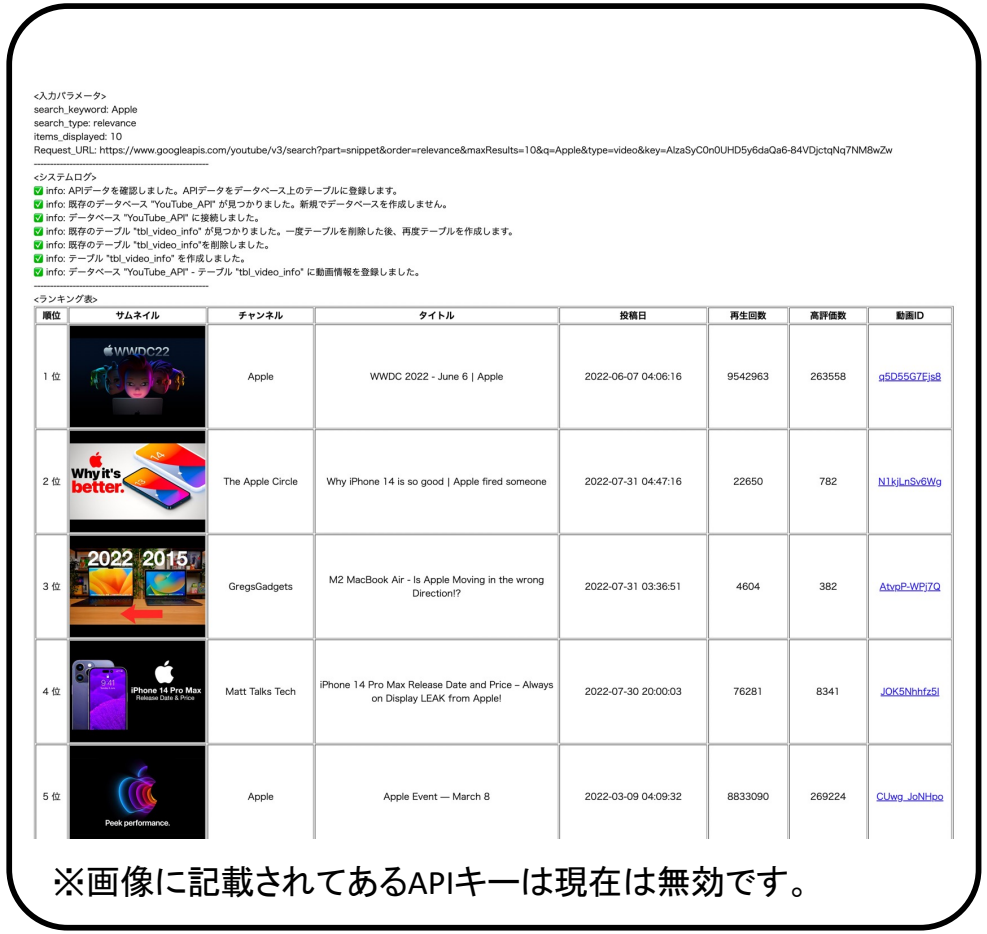
1. 検索キーワードには任意のキーワードを入力する。入力キーワードを元に検索を行う。

2. 検索タイプは「キーワードの関連性」「投稿日」「再生回数」「高評価数」を選択できる。

3. 件数表示は上位何件を表示させるか選択できる。
（10, 20, 30, 40, 50件のいずれかを選択可能）

・検索実行ボタンを押すと、上記で入力した情報を元に関連した動画の検索を行う。

3. Webページのデザイン



<入力パラメータ>
search_keyword: Apple
search_type: relevance
items_displayed: 10
Request_URL: https://www.googleapis.com/youtube/v3/search?part=snippet&order=relevance&maxResults=10&q=Apple&type=video&key=AIzaSyC0n0UHD5y6daQa6-84VDjctqNq7NM8wZw
---
<システムログ>
✅ info: APIデータを確認しました。APIデータをデータベース上のテーブルに登録します。
✅ info: 既存のデータベース "YouTube_API" が見つかりました。新規でデータベースを作成しません。
✅ info: データベース "YouTube_API" に接続しました。
✅ info: 既存のテーブル "tbl_video_info" が見つかりました。一度テーブルを削除した後、再度テーブルを作成します。
✅ info: 既存のテーブル "tbl_video_info"を削除しました。
✅ info: テーブル "tbl_video_info" を作成しました。
✅ info: データベース "YouTube_API" - テーブル "tbl_video_info" に動画情報を登録しました。

<ランキング表>

順位	サムネイル	チャンネル	タイトル	投稿日	再生回数	高評価数	動画ID
1 位		Apple	WWDC 2022 - June 6 | Apple	2022-06-07 04:06:16	9542963	263558	q5D55G7Ejs8
2 位		The Apple Circle	Why iPhone 14 is so good | Apple fired someone	2022-07-31 04:47:16	22650	782	N1kjLnSv6Wg
3 位		GregsGadgets	M2 MacBook Air - Is Apple Moving in the wrong Direction!?	2022-07-31 03:36:51	4604	382	AtvpP-WPj7Q
4 位		Matt Talks Tech	iPhone 14 Pro Max Release Date and Price – Always on Display LEAK from Apple!	2022-07-30 20:00:03	76281	8341	JOK5Nhhfz5I
5 位		Apple	Apple Event — March 8	2022-03-09 04:09:32	8833090	269224	CUwg_JoNHpo

※画像に記載されてあるAPIキーは現在は無効です。

図2:検索結果画面

図1で入力した情報を元に検索を実行すると、左のような画面が表示される。

画面上部から、「入力したパラメータ」「システムログ」「ランキング表」の順に表示される。

ランキング表については、図1で入力したデータを元に出力する。

表の左から、検索タイプで選択した順位、サムネイル、タイトル、投稿日、再生回数、高評価数、動画リンクを表示する。

4. プログラムの説明

4.1 プログラムの構成

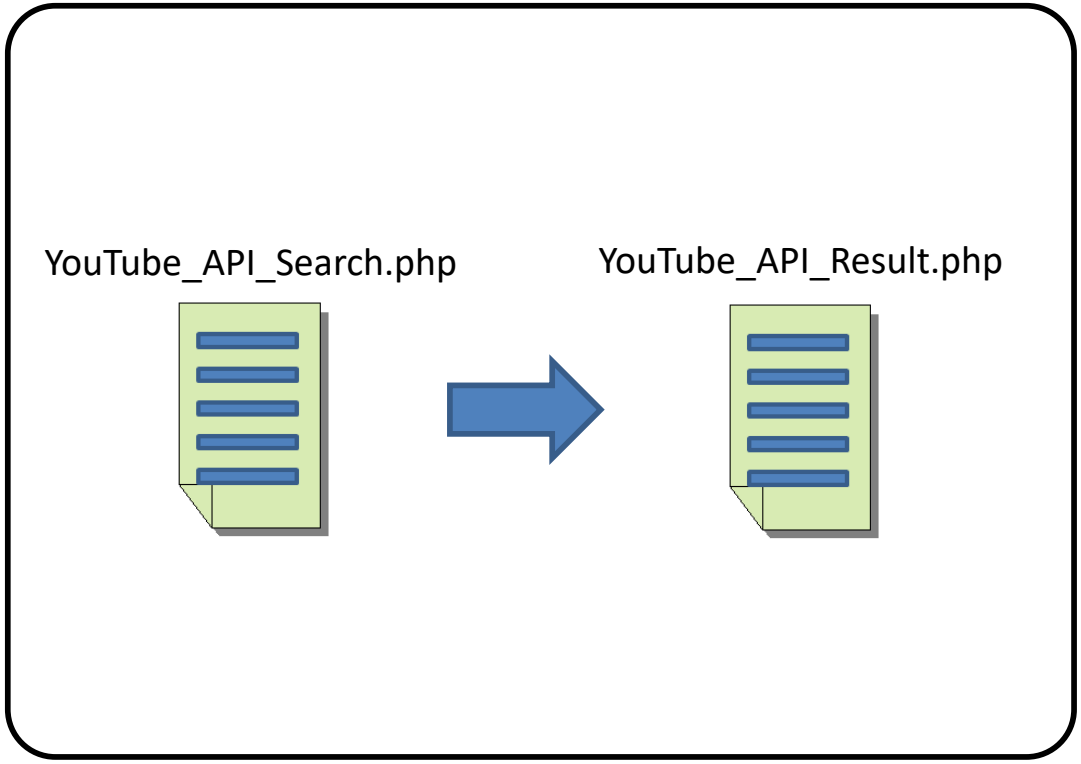


図1:PHPプログラムファイル間の関係

①YouTube_API_Search.php は検索パラメータを入力できるようにしたプログラムで、「検索実行」ボタンを押すと、YouTube_API_Result.phpへ遷移する。

②遷移した後、YouTube_API_Search.php で入力したパラメータに基づき、YouTube_API_Result.php で関連した動画情報を検索し、動画情報をデータベースに登録する。そしてブラウザに表を出力する。

4.2.1 YouTube_API_Search.php での関数の構成

- YouTube_API_Search.php では、2個の関数で構成されている

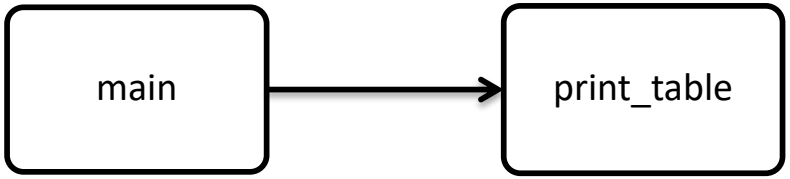


図2.1:YouTube_API_Search.php での関数の構成図

4.2.2 YouTube_API_Result.php での関数の構成

- YouTube_API_Result.php では、10個の関数で構成されている。

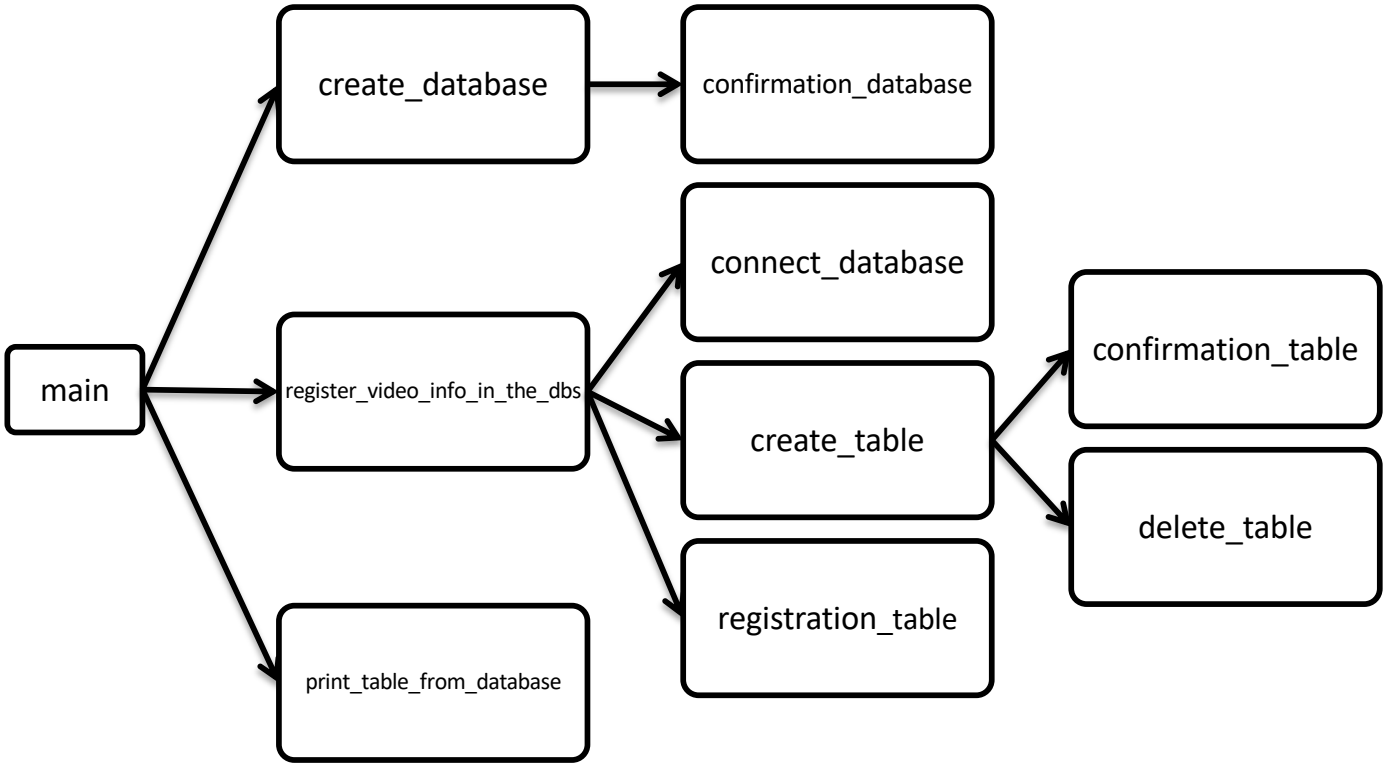


図2.2:YouTube_API_ Result.php での関数の構成図

4.3 各関数の説明

（1）YouTube_API_Search.phpのmain 関数: 入力パラメータに必要な配列の初期化を行う関数

```
// main 関数
function main()
{
// 検索タイプの連想配列
$search_type = array("relevance" => "キーワードの関連性", "date" => "投稿日", "viewCount" => "再生回数", "rating" => "高評価数");

// 表示件数の配列 (10, 20, 30, 40, 50)
for ($i = 0; $i < 5; $i++)
{
$items_displayed[$i] = ($i + 1) * 10;
}

// テーブルの中に入力フィールド等を表示する
print_table($search_type, $items_displayed);
}
```

(1)「検索タイプ」のプルダウンメニューの各項目名を配列に格納する。

(2)「表示件数」のプルダウンメニューの各項目名を配列に格納する。

4.3 各関数の説明

（2）YouTube_API_Search.phpのprint_table 関数: 入力パラメータを表示するための関数

```
// テーブルを表示する関数
function print_table($search_type, $items_displayed)
{
// CSS (Style Sheet)
print(
"<style> .header, .button_submit{ text-align: center;}
.cell_size table{margin: auto;}
.cell_size th{ width: 200px; height: 40px;}
.cell_size td{ width: 500px; height: 40px;}
.cell_size .select_1 select{ width: 270px;}
.cell_size .select_2 select{ width: 70px;}
.button_submit input{ width: 100px; height: 30px;}
</style>¥n");
print "<div class = ¥"header¥"> <h2> YouTube における特定キーワードのランキング表の出力 </h2> </div>¥n";

// パラメータ入力フィールド
print "<form action = ¥"http://localhost:8080/project/YouTube_API_Result.php¥" method = ¥"post¥">¥n";
print "<div class = ¥"cell_size¥">";
print "<table border = ¥"1¥" style = ¥"text-align: center;¥">¥n";

// 検索キーワード
print "<tr>¥n <th>検索キーワード</th> <td> <input type = ¥"text¥" name = ¥"search_keyword¥" size = ¥"60¥"
required> </td>¥n</tr>¥n";

// 検索タイプ
print "<tr>¥n <th>検索タイプ</th> <td> <div class = ¥"select_1¥"> <select size = ¥"1¥" name = ¥"search_type¥"> ";
foreach ($search_type as $key => $value)
{
print "<option value = {$key}>{$value} ({$key})</option> ";
}
print "</select> </div> </td>¥n</tr>¥n";

// 表示件数
print "<tr> <th>表示件数</th> <td> <div class = ¥"select_2¥"> <select size = ¥"1¥" name = ¥"items_displayed¥">";
foreach ($items_displayed as $value)
{
print "<option value = {$value}>{$value} 件</option> ";
}
print "</select> </div> </td>¥n</tr>¥n";

print "</table>¥n";
print "</div>¥n";

print "<br> <div class = ¥"button_submit¥"> <input type = ¥"submit¥" value = ¥"検索実行¥"> </div>¥n";

print "</form>¥n";
}
```

(1)CSSを記載し、表示を整える。

(2)各項目を表示するためのHTMLを記載する。（YouTube_API_Result.phpにパラメータ情報を送れるようにform action タグを記載している）

4.3 各関数の説明

(3) YouTube_API_Result.phpのmain 関数: 全ての処理をまとめた関数

```
// main 関数
function main($info)
{
// 文字列置換関数 URLチェック {str_replace("検索を行う文字列", "置き換えを行う文字列", "対象の文字列", "str_replace処理の回数")}
$conversion_string = str_replace(array(" ", "   "), "+", $info["search_keyword"]);

// 検索用URL 生成
$URL = $info["request_URL"] ."/search?part=snippet&order=".
$info["search_type"] ."&maxResults=". $info["items_displayed"] ."&q=".
$conversion_string ."&type=video&key=". $info["API_key"];
print "Request_URL: {$URL}<br>¥n";

print "------------------------------------------------------------ <br>¥n";
print "<システムログ> <br>¥n";
// ファイルの内容の読み込み
$json = file_get_contents($URL);

// 連想配列にする
$api_array = json_decode($json, true);

print “&#x2705; info: APIからのデータを確認しました。APIからのデータをデータベース上のテーブルに登録します。<br>¥n";

// 作成するデータベース名
$database_name = "YouTube_API";

// データベースを作成する関数
create_database($database_name);

// デバック用（APIから取得した配列を表示する）
// print_array($api_array);

// APIから動画情報を取得してデータベースに登録する
$dbs_info = register_video_info_in_the_dbs($info, $api_array, $database_name);

// データベースに登録してある動画情報を出力する
print_table_from_database($dbs_info);
}
```

(1) str_replace関数（文字列置換関数）を実行し、検索キーワードに空白があった場合に”+”に置き換えるようにする。

(2) 検索に必要なパラメータを検索用URLに連結していく。

(3) APIデータファイルの内容の読み込みをした後に、連想配列にする。

(4) データベースを作成する関数を実行する。

(5) 取得した連想配列から動画情報を取得してデータベースに登録する関数を実行する。

(6)データベースに登録してある動画情報を出力する関数を実行する。

4.3 各関数の説明

（4）YouTube_API_Result.phpのcreate_database 関数: データベースを作成する関数

```
// データベースを作成する関数
function create_database($database_name)
{
try
{
$dbs = 'mysql:host=localhost;';
$user = 'root';
$password = "";

// データベースに接続
$pdo = new PDO($dbs, $user, $password);

// データベースが既に存在するかどうかを確認する
if (confirmation_database($pdo, $database_name))
{
return 0; // データベースが存在するならば、関数を抜ける
}

// データベースを作成するクエリを書く
$query = "CREATE DATABASE {$database_name}";

$stmt = $pdo->prepare($query);
$stmt->execute();
}
catch (Exception $e)
{
print("&#x1f6ab; error: データベースを作成中にエラーが発生しました。
<br>¥n");
print($e->getMessage());
exit();
}
print "&#x2705; info: データベース ¥"". $database_name ."¥" を作成しました。
<br>¥n";
}
```

(1) MySQLに接続する。

(2) confirmation_database 関数を呼び出し、同一のデータベース名が既に存在するかどうかを確認する。返り値が true ならば、同一のデータベース名が存在しており、false ならば、同一のデータベース名が存在しない。

(3) (2)の返り値が true ならば、関数を抜ける。false ならば、データベースを作成する。

4.3 各関数の説明

(5) YouTube_API_Result.phpのconfirmation_database 関数: 特定のデータベースが存在するかを確認する関数

```
// 特定のデータベースが存在するかを確認する関数
function confirmation_database($pdo, $database_name)
{
// データベースを探索するクエリ
$query = "SHOW DATABASES LIKE '". $database_name ."'";

$stmt = $pdo -> prepare($query);
$stmt -> execute();

$info = $stmt -> fetch(PDO::FETCH_ASSOC);

foreach ($info as $value)
{
if ($value == $database_name)
{
print "&#x2705; info: 既存のデータベース ¥"". $database_name ."¥" が
見つかりました。新規でデータベースを作成しません。<br>¥n";
// データベースが存在するならば、true を返す
return true;
}
}
print "&#x2705; info: 既存のデータベース ¥"". $database_name ."¥" が
見つかりませんでした。新規でデータベースを作成します。<br>¥n";
// データベースが存在しなければ、false を返す
return false;
}
```

(1) 引数にしていたデータベース名から、データベースを探索する。

(2) 同一のデータベース名が存在したのならば、true を返す。さもなくば、false を返す。

4.3 各関数の説明

(6) YouTube_API_Result.phpのregister_video_info_in_the_dbs 関数: 動画情報を取得し、データベースに登録する関数

```
// 動画情報を取得し、データベースに登録する関数 (register video information in the database: 動画情報をデータベースに登録)
// パラメータ: 設定項目, APIデータ, 接続するデータベース名
function register_video_info_in_the_dbs($info, $api_array, $database_name)
{
// ■ 作成するテーブル情報 ----------------------------
// テーブルの名前
$table_name = "tbl_video_info";

// テーブルの定義 (英語(データベースの属性名) => (定義, 日本語(ブラウザ上の表示))
$table_info = array(
"ranking" => array("definition" => "INTEGER", "japanese" => "順位"),
"thumbnail_URL" => array("definition" => "VARCHAR(200)", "japanese" => "サムネイル"),
"channelTitle" => array("definition" => "VARCHAR(50)", "japanese" => "チャンネル"),
"title" => array("definition" => "VARCHAR(200)", "japanese" => "タイトル"),
"publishedAt" => array("definition" => "VARCHAR(30)", "japanese" => "投稿日"),
"view_Count" => array("definition" => "INTEGER", "japanese" => "再生回数"),
"like_Count" => array("definition" => "INTEGER", "japanese" => "高評価数"),
"video_ID" => array("definition" => "VARCHAR(100)", "japanese" => "動画ID"));
// ----------------------------------------------
try
{
// "YouTube_API" データベースに接続する（実引数: 接続したいデータベース名）
$pdo = connect_database($database_name);

// データベースにテーブルを作成する
create_table($pdo, $table_name, $table_info);

// APIからデータを取得
foreach ($api_array["items"] as $key => $array)
{
// print_array($array["snippet"]);
// 1. 動画投稿日
$publishedAt = new DateTime($array["snippet"]["publishedAt"]); // 世界標準時間 UTC
$publishedAt->setTimeZone(new DateTimeZone('Asia/Tokyo')); // 日本時間 JST
$data[$key]["publishedAt"] = (string) $publishedAt->format('Y-m-d H:i:s');

// 2. 動画ID
$data[$key]["video_ID"] = $array["id"]["videoId"];

// 3. 再生回数 & 高評価数
$video_info_URL = $info["request_URL"] ."/videos?part=statistics&id=". $data[$key]["video_ID"] ."&key=". $info["API_key"];
$json_video_info = file_get_contents($video_info_URL);
$array_video_info = json_decode($json_video_info, true);
$data[$key]["view_Count"] = $array_video_info["items"][0]["statistics"]["viewCount"];
if (isset($array_video_info["items"][0]["statistics"]["likeCount"]))
{
$data[$key]["like_Count"] = $array_video_info["items"][0]["statistics"]["likeCount"];
}
else
{
$data[$key]["like_Count"] = 0;
}

// 4. タイトル
$data[$key]["title"] = $array["snippet"]["title"];
// 5. サムネイルURL
$data[$key]["thumbnail_URL"] = $array["snippet"]["thumbnails"]["high"]["url"];

// 6. チャンネル名
$data[$key]["channelTitle"] = $array["snippet"]["channelTitle"];

// テーブルに動画情報を登録する
registration_table($table_info, $pdo, $data[$key], $key);
}
}
catch (Exception $e)
{
print("&#x1f6ab; error: APIからのデータを取得する際にエラーが発生しました。<br>¥n");
print($e->getMessage());
die();
}
print "&#x2705; info: データベース ¥"". $database_name ."¥" - テーブル ¥"". $table_name ."¥" に動画情報を登録しました。<br>¥n";
return array("table_name" => $table_name, "table_info" => $table_info, "pdo" => $pdo);
}
```

(1) 作成するテーブル情報を記載する。

(2) connect_database 関数から、“YouTube_API” データベースに接続する。

(3) create_table 関数から、データベースにテーブルを作成する。

(4) APIからデータを取得する。

(5) registration_table 関数から、テーブルに動画情報を登録する。

4.3 各関数の説明

（7）YouTube_API_Result.phpのconnect_database 関数: データベースに接続する関数

```
// データベースに接続する関数
function connect_database($database_name)
{
try
{
$dbs = 'mysql:dbname='.$database_name.';host=localhost';
$user = 'root';
$password = "";

// データベースに接続
$pdo = new PDO($dbs, $user, $password);
}
catch (Exception $e)
{
print "&#x1f6ab; error: データベース ¥"".
$database_name ."¥" に接続できませんでした。<br>¥n";
print($e->getMessage());
die();
}
print "&#x2705; info: データベース ¥"". $database_name ."¥"
に接続しました。<br>¥n";
return $pdo;
}
```

データベースに接続するための設定を行う。

4.3 各関数の説明

(8) YouTube_API_Result.phpのcreate_table 関数: テーブルを作成する関数。

```
// テーブルを作成する関数
function create_table($pdo, $table_name, $table_info)
{
try
{
// テーブルが既に存在するかどうかを確認する
if (confirmation_table($pdo, $table_name))
{
// 既存のテーブルを削除する
delete_table($pdo, $table_name);
}

// $table_info の要素数(属性数)をカウントする
$table_info_num = count($table_info);
// カウント用変数
$count = 0;
$query = "CREATE TABLE {$table_name}(";
foreach ($table_info as $attribute => $array)
{
$query .= "{$attribute} {$array["definition"]}";
$count++;
// もし、テーブルの属性数よりカウント値が小さければ", "を連結する
if ($table_info_num > $count) {$query .= ", ";}
// さもなくば、")"を連結する
else {$query .= ")";}
}

$stmt = $pdo -> prepare($query);
$stmt -> execute();
}
catch (Exception $e)
{
print "&#x1f6ab; error: テーブル ¥"". $table_name ."¥" を作成中にエラーが発生しました。
<br>¥n";
print($e->getMessage());
die();
}
print "&#x2705; info: テーブル ¥"". $table_name ."¥" を作成しました。<br>¥n";
}
```

(1) confirmation_table 関数から、特定のテーブルがデータベース上に存在するかを確認する。

(2) (1) で true が返ったのならば、delete_table 関数で既存のテーブルを削除する。

(3) $table_info の要素数(属性数)をカウントする。

(4) テーブルを作成するためのクエリを生成する。

※↓下記のSQLを生成する

```
$query = "CREATE TABLE tbl_video_info(
ranking INTEGER,
thumbnail_URL VARCHAR(200),
channelTitle VARCHAR(50),
title VARCHAR(100),
publishedAt VARCHAR(30),
view_Count INTEGER,
like_Count INTEGER,
video_ID VARCHAR(100))";
```

4.3 各関数の説明

(9) YouTube_API_Result.phpのconfirmation_table 関数: 特定のテーブルがデータベース上に存在するかを確認する関数

```
// 特定のテーブルがデータベース上に存在するかを確認する関数
function confirmation_table($pdo, $table_name)
{
$query = "SHOW TABLES LIKE '". $table_name ."'";

$stmt = $pdo -> prepare($query);
$stmt -> execute();

$info = $stmt -> fetch(PDO::FETCH_ASSOC);

foreach ($info as $value)
{
if ($value == $table_name)
{
print "&#x2705; info: 既存のテーブル ¥"". $table_name ."¥" が見つかりました。
一度テーブルを削除した後、再度テーブルを作成します。<br>¥n";
// テーブルが存在するならば、true を返す
return true;
}
}
print "&#x2705; info: 既存のテーブル ¥"". $table_name ."¥" が見つかりませんで
した。新規でテーブルを作成します。<br>¥n";
// テーブルが存在しなければ、false を返す
return false;
}
```

(1) 引数にしていたテーブル名から、テーブルを探索する。

(2) 同一のテーブル名が存在したのならば、true を返す。さもなくば、false を返す。

4.3 各関数の説明

(10) YouTube_API_Result.phpのdelete_table関数:データベース上に登録されてあるテーブルを削除する関数

```
// データベース上に登録されてあるテーブルを削除する関数
function delete_table($pdo, $table_name)
{
try
{
$query = "DROP TABLES ". $table_name;

$stmt = $pdo -> prepare($query);
$stmt -> execute();
}
catch (Exception $e)
{
print("&#x1f6ab; error: 既存のテーブル ¥"". $table_name ."¥" を削除できませんでした。<br>¥n");
print($e -> getMessage());
die();
}
print "&#x2705; info: 既存のテーブル ¥"". $table_name ."¥"を削除しました。<br>¥n";
}
```

引数のテーブル名から、該当するテーブルを削除する。

4.3 各関数の説明

（11）YouTube_API_Result.phpのregistration_table関数: テーブルに動画情報を登録する関数

```
function registration_table($table_info, $pdo, $data, $rank)
{
try
{
// $table_info の要素数(属性数)をカウントする
$table_info_num = count($table_info);
// while 処理(文字列が完成)が終わったかどうかの判定用
$w_bool = false;
// 属性名(INSERT INTO tbl_video_info(〇, 〇, 〇 ← ココ))を記載したかどうかの判定用
$at_bool = false;
// ($array) 現在指しているポインタの次が"VARCHAR"かどうかを判定
$next_char_bool = false;

$query = "INSERT INTO tbl_video_info(";
while (!$w_bool)
{
// カウント用変数
$count = 0;
foreach ($table_info as $attribute => $array)
{
// 属性名を記載できていないのならば、
if (!$at_bool)
{
$query .= "{$attribute}";
$count++;
// もし、テーブルの属性数よりカウント値が小さければ", "を連結する
if ($table_info_num > $count) {$query .= ", ";}
// さもなくば、")VALUES(($rank) + 1, "を連結する
else {$query .= ") VALUES(";$at_bool = true;}
}
// 属性名を記載できているのならば、
else
{
// 属性名が "ranking" ならば、（APIにない要素なので、ここに書く必要あり）
if ($attribute == "ranking") {$query .= ($rank + 1);}
else if (gettype($data[$attribute]) == "string" && $count == 0) {$query .= "'". $data[$attribute] ."'";}
// さもなくば、APIから取得したデータをquery（文字列）に追加する
else {$query .= $data[$attribute];}

// 現在指しているポインタ($pointer)が "VARCHAR" ならば、(*)
if ($next_char_bool)
{
// 最後に"'"をつける ex('タイトル')
$query .= "'";
// false に戻す
$next_char_bool = false;
}
// 現在を指しているポインタの次を調べる
if ($table_info_num - 1 > $count)
{
$pointer = next($table_info)["definition"];
}
else {$pointer = NULL;}

// 次に何か型が定義されていたのならば、
if ($pointer)
{
// "VARCHAR"型(not INTEGER)ならば、
if ($pointer != "INTEGER")
{
// 終わりに", '"をつける ex('タイトル', ')
$query .= ", '";
// 文字列の最後に"'"をつけるため、フラグを立てておく (*)
$next_char_bool = true;
}
else {$query .= ", ";}
}
// さもなくば(ポインタが配列の最後を指している状態)、")"を連結する
else
{
$query .= ")";
reset($array);
// $query が完成したので while ループを抜ける
$w_bool = true;
}
$count++;
}
}
}
/*
$query = "INSERT INTO tbl_video_info(ranking, thumbnail_URL, channelTitle, title, publishedAt, view_Count, like_Count, video_ID)
VALUES(
($rank) + 1,
'{$data["thumbnail_URL"]}',
'{$data["channelTitle"]}',
'{$data["title"]}',
'{$data["publishedAt"]}',
{$data["view_Count"]},
{$data["like_Count"]},
'{$data["video_ID"]}')";*/
$stmt = $pdo -> prepare($query);
$stmt -> execute();
}
catch (PDOException $e)
{
print "&#x1f6ab; error: テーブルに動画情報を登録するためのクエリ作成中にエラーが発生しました。<br>¥n";
print($e->getMessage());
die();
}
}
```

テーブルに追加するためのクエリを生成する。

※↓下記のSQLを生成する。

```
INSERT INTO tbl_video_info(ranking, thumbnail_URL,
channelTitle, title, publishedAt, view_Count,
like_Count, video_ID)
VALUES(
{$rank} + 1,
'{$data["thumbnail_URL"]}',
'{$data["channelTitle"]}',
'{$data["title"]}',
'{$data["publishedAt"]}',
{$data["view_Count"]},
{$data["like_Count"]},
'{$data["video_ID"]}')
```

4.3 各関数の説明

（12）YouTube_API_Result.phpのprint_table_from_database関数: データベースからブラウザにテーブルを表示する関数

```
// データベースからブラウザにテーブルを表示する関数
function print_table_from_database($dbs_info) // array("table_name" => $table_name, "table_info" => $table_info, "pdo" => $pdo);
{
print "---------------------------------------------------------- <br>¥n";
print "<ランキング表> <br>¥n";
try
{
// ■ 各属性にあったHTMLタグ ----------------------------
$html = array(
"ranking" => array("<td style = ¥"width: 60px;¥">", " 位</td>"),
"thumbnail_URL" => array("<td> <img src = ¥"", "¥" name = ¥"", "¥" width = ¥"240¥" height = ¥"180¥"> </td>"),
"channelTitle" => array("<td style = ¥"width: 200px;¥">", "</td>"),
"title" => array("<td style = ¥"width: 500px;¥">", "</td>"),
"publishedAt" => array("<td style = ¥"width: 300px;¥">", "</td>"),
"view_Count" => array("<td style = ¥"width: 150px;¥">", "</td>"),
"like_Count" => array("<td style = ¥"width: 150px;¥">", "</td>"),
"video_ID" => array("<td style = ¥"width: 150px;¥">", "<a href = ¥"https://www.youtube.com/watch?v=", "¥">", "</a>", "</td>"));
// ----------------------------------------------------

// テーブルのデータを取得
$query_data = "SELECT * FROM ". $dbs_info["table_name"];
$stmt_data = $dbs_info["pdo"]->prepare($query_data);
$stmt_data -> execute();

// テーブルのカラムを取得
$query_col = "SHOW COLUMNS FROM ". $dbs_info["table_name"];
$stmt_col = $dbs_info["pdo"]->prepare($query_col);
$stmt_col -> execute();

// ブラウザ上に表示するテーブルを作成
print "<table border = ¥"1¥" style = ¥"text-align: center;¥">¥n";
print "<tr>¥n";
// テーブルの属性名を表示
while ($column = $stmt_col->fetch(PDO::FETCH_COLUMN))
{
print "<th>". $dbs_info["table_info"][$column]["japanese"] ."</th> ";
}
print "</tr>¥n";

// テーブルの値を表示
// <行> SQLの実行結果の取得 fetchで連想配列で取得
while ($info = $stmt_data -> fetch(PDO::FETCH_ASSOC))
{
print "<tr>";
// テーブルのカラムを取得
$query_col = "SHOW COLUMNS FROM ". $dbs_info["table_name"];
$stmt_col = $dbs_info["pdo"]->prepare($query_col);
$stmt_col -> execute();

// <列> 属性の並び順に沿ったものを表示する
while ($column = $stmt_col->fetch(PDO::FETCH_COLUMN))
{
// 各カラムによって生成するHTMLが変わる
if ($column == "thumbnail_URL")
{
print $html[$column][0] . $info[$column] . $html[$column][1] . $info["title"] . $html[$column][2];
}
else if ($column == "video_ID")
{
print $html[$column][0] . $html[$column][1] . $info[$column] . $html[$column][2] . $info[$column] . $html[$column][3] . $html[$column][4];
}
else
{
print $html[$column][0] . $info[$column] . $html[$column][1];
}
}
print "</tr>¥n";
}
print "¥n</table>¥n";
}
catch (Exception $e)
{
print "&#x1f6ab; error: 表の出力に失敗しました。<br>¥n";
print($e->getMessage());
die();
}
}
```

(1) HTMLタグと属性名を配列でリンクさせる。

(2) ブラウザにテーブルを表示する。

4.4 プログラムとWebAPIとの関係

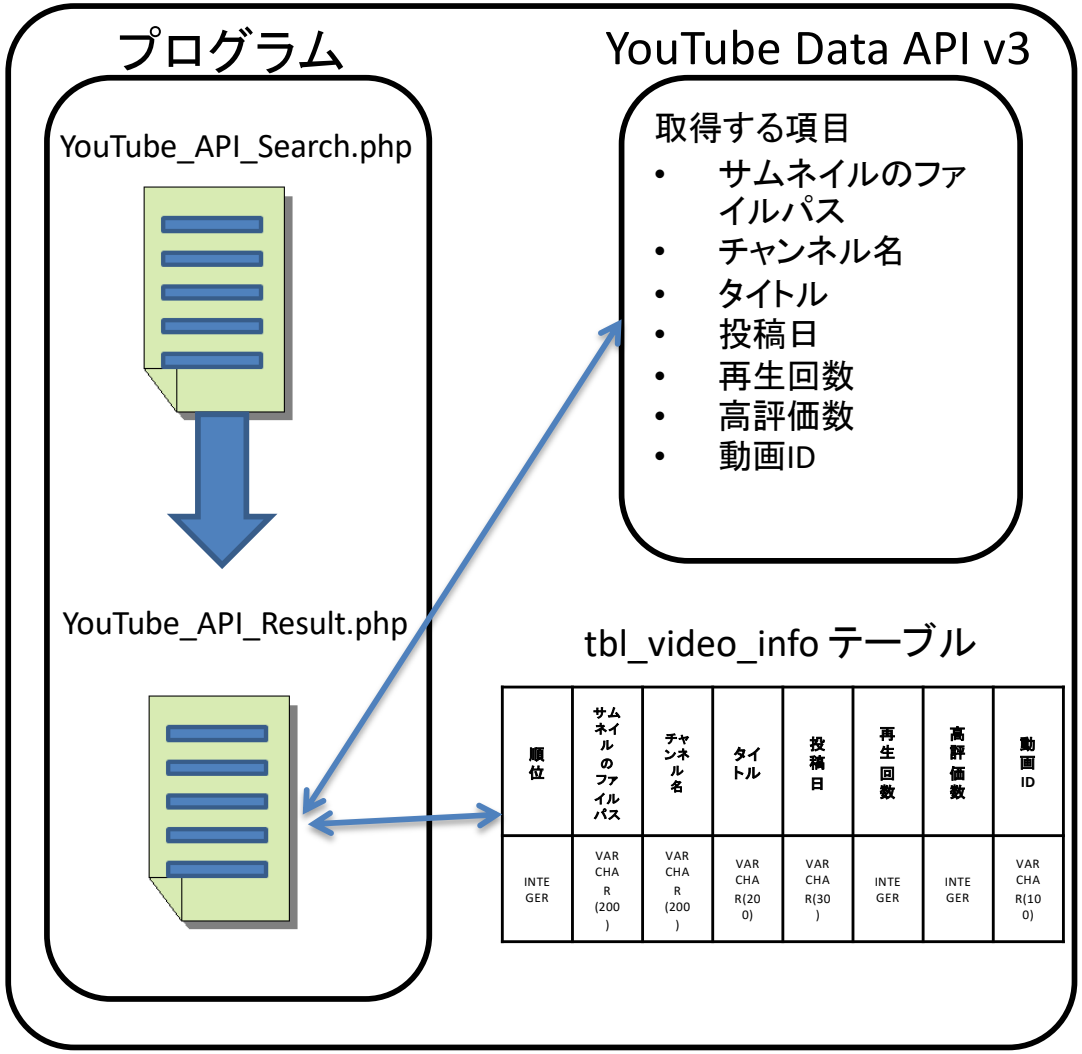


図3：プログラムとYouTube Data API v3の関係

- YouTube_API_Result.phpからYouTube Data API v3にリクエストを送り、
  - サムネイルのファイルパス
  - チャンネル名
  - タイトル
  - 投稿日
  - 再生回数
  - 高評価数
  - 動画ID

  を取得する。

- データベースには、上記のAPIデータを登録する。

5. データベースの仕様

(1)テーブル名: tbl_video_info

　このテーブルには、APIから取得した動画情報と、順位を登録する。

5. データベースの仕様

5.1データベースの定義と状況

(1)表形式で定義とその例の2種類の表を書く。

(1-1)テーブルの定義（項目名と型の定義）

順位	サムネイルのファイルパス	チャンネル名	タイトル	投稿日	再生回数	高評価数	動画ID
INTEGER	VARCHAR (200)	VARCHAR (200)	VARCHAR (200)	VARCHAR (30)	INTEGER	INTEGER	VARCHAR (100)

5. データベースの仕様

(1-2)テーブルの例

1	https://…（ファイルパス）	太郎	A	2021-07-23	60000	3000	kdjcnek
2	https://…（ファイルパス）	花子	B	2019-03-13	40000	1500	hfidhekd
3	https://…（ファイルパス）	海斗	C	2022-02-05	30000	2000	jdkdjekf
4	https://…（ファイルパス）	優子	D	2021-01-26	70000	1200	kdpfjrkj
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

6. WebAPIの仕様

(1) YouTube Data API v3

(2) https://www.googleapis.com/youtube/v3

(3)YouTube 上に存在する動画やチャンネル情報を取得できる。

(4)

- WebAPIの具体的な入力
  - 検索キーワード
    - “岡山理科大学”, “平成のニュース”…
  - 動画ID
    - 動画ごとに振り分けられている “46GdjfosP” などのランダムな文字列
- WebAPIの具体的な出力
  - 動画タイトル
    - “〇〇のニュース”, “平成時代のヒット曲”…
  - 再生回数
    - 3436, 67543, 204554…
  - チャンネルURL
    - https://www.youtube.com/channel...

(5) WebAPIを動かすためのPHPのプログラム

5.1 プログラムリスト

```
$info = array(
"request_URL" => "https://www.googleapis.com/youtube/v3",
"API_key" => "AIzaSyC0n0UHD5y6daQa6-84VDjctqNq7NM8wZw",
"search_keyword" => ”Apple",
"search_type" => "relevance",
"items_displayed" => 10);

// 文字列置換関数 URLチェック {str_replace("検索を行う文字列", "
置き換えを行う文字列", "対象の文字列", "str_replace処理の回数
")}
$conversion_string = str_replace(array(" ", "   "), "+",
$info["search_keyword"]);

// 検索用URL 生成
$URL = $info["request_URL"] ."/search?part=snippet&order=".
$info["search_type"] ."&maxResults=".
$info["items_displayed"] ."&q=".
$conversion_string ."&type=video&key=". $info["API_key"];
// print "Request_URL: {$URL}<br>¥n";

// ファイルの内容の読み込み
$json = file_get_contents($URL);

// 連想配列にする
$api_array = json_decode($json, true);

print "<pre>";
var_dump($api_array);
print "</pre><br>¥n";
```

5.2 実行結果


```
array(6) {
 ["kind"]=>
 string(26) "youtube#searchListResponse"
 ["etag"]=>
 string(27) "7k_SsXroO5WZND5OdhBC2JBow_w"
 ["nextPageToken"]=>
 string(6) "CAoQAA"
 ["regionCode"]=>
 string(2) "JP"
 ["pageInfo"]=>
 array(2) {
  ["totalResults"]=>
  int(1000000)
  ["resultsPerPage"]=>
  int(10)
 }
 ["items"]=>
 array(10) {
  [0]=>
  array(4) {
   ["kind"]=>
   string(20) "youtube#searchResult"
   ["etag"]=>
   string(27) "Wob7s8YM1TE4Gz1ZhhqkBe3bhIA"
   ["id"]=>
   array(2) {
    ["kind"]=>
    string(13) "youtube#video"
    ["videoId"]=>
    string(11) "q5D55G7Ejs8"
   }
   ["snippet"]=>
   array(8) {
    ["publishedAt"]=>
    string(20) "2022-06-06T19:06:16Z"
    ["channelId"]=>
    string(24) "UCE_M8A5yxnLfW0KghEeajjw"
    ["title"]=>
    string(26) "WWDC 2022 - June 6 | Apple"
    ["description"]=>
```


リクエストURL、APIキー、各パラメータを一つのURLに記載することでAPIにアクセスできる。

7. 実行結果(1)

図3:検索用の入力画面

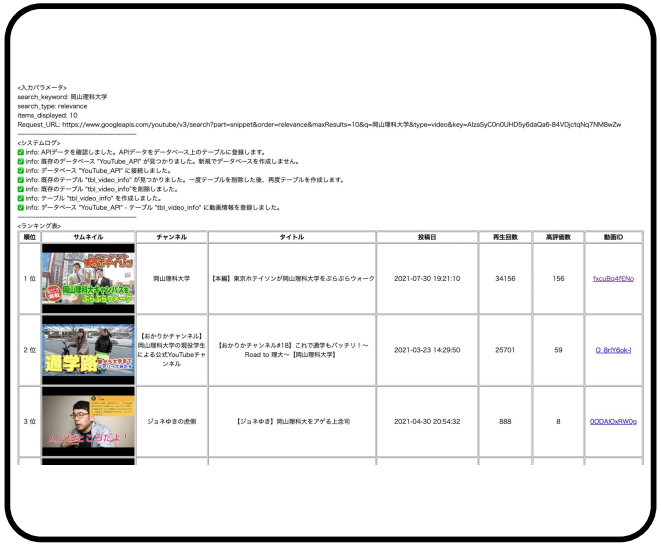
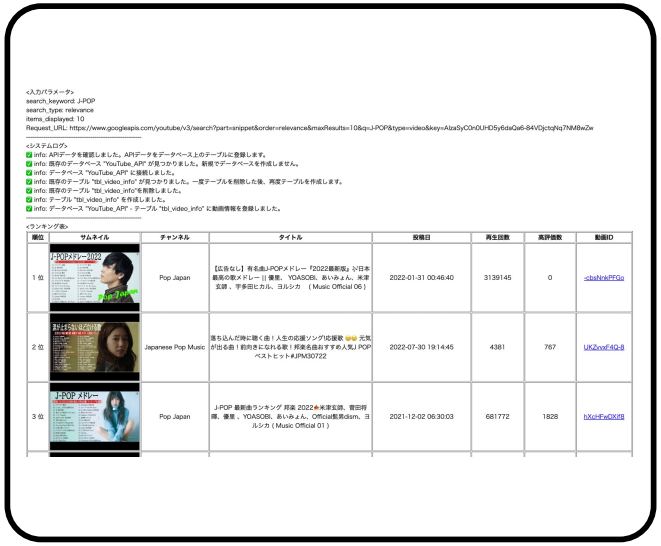


図4:検索結果の出力画面

検索キーワードに「岡山理科大学」
検索タイプに「再生回数」
表示件数に「10件」と選択すると、
岡山理科大学に関連した動画を再生回数の多い順に
並べた表が表示される。また、表示件数は10件である。

7. 実行結果(2)

図3:検索用の入力画面



<入力パラメータ>
search_keyword: J-POP
search_type: relevance
items_displayed: 10
Request_URL: https://www.googleapis.com/youtube/v3/search?part=snippet&order=relevance&maxResults=10&q=J-POP&type=video&key=AIzaSyC0n0UHD5y6daQa6-84VDjctqNq7NM8wZw

<システムログ>
✅ info: APIデータを確認しました。APIデータをデータベース上のテーブルに登録します。
✅ info: 既存のデータベース "YouTube_API" が見つかりました。新規でデータベースを作成しません。
✅ info: データベース "YouTube_API" に接続しました。
✅ info: 既存のテーブル "tbl_video_info" が見つかりました。一度テーブルを削除した後、再度テーブルを作成します。
✅ info: 既存のテーブル "tbl_video_info"を削除しました。
✅ info: テーブル "tbl_video_info" を作成しました。
✅ info: データベース "YouTube_API - テーブル "tbl_video_info" に動画情報を登録しました。

<ランキング表>

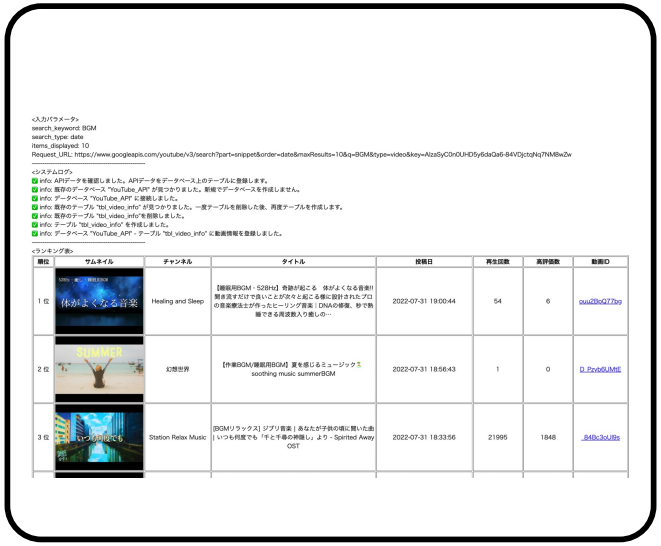
順位	サムネイル	チャンネル	タイトル	投稿日	再生回数	高評価数	動画ID
1 位		Pop Japan	【広告なし】有名曲J-POPメドレー「2022最新版」☆日本最高の歌メドレー!! 優里、YOASOBI、あいゆえん、米津玄師 、宇多田ヒカル、ヨルシカ （ Music Official 06 ）	2022-01-31 00:46:40	3139145	0	-cbsNnkPFGo
2 位		Japanese Pop Music	落ち込んだ時に聴く曲！人生の応援ソング!応援歌 💪💪 元気が出る曲！前向きになれる曲！邦楽名曲はすすめ人気J POP ベストヒット#JPM30722	2022-07-30 19:14:45	4381	767	UKZxoxF4Q-8
3 位		Pop Japan	J-POP 最新曲ランキング 邦楽 2022☘米津玄師、菅田将暉、優里 、YOASOBI、あいゆえん、Official髭男dism、ヨルシカ (Music Official 01)	2021-12-02 06:30:03	681772	1828	hXcHFwOXrf8

図4:検索結果の出力画面

検索キーワードに「J-POP」
検索タイプに「キーワードの関連性」
表示件数に「10件」と選択すると、J-POPに最も関連した動画を並べた表が表示される。また、表示件数は10件である。

7. 実行結果(3)

図3:検索用の入力画面



<入力パラメータ>
search_keyword: BGM
search_type: date
items_displayed: 10
Request_URL: https://www.googleapis.com/youtube/v3/search?part=snippet&order=date&maxResults=10&q=BGM&type=video&key=AIzaSyC0nOUHD5y6daQa6-84VDjctqNq7NMBwZw

<システムログ>
✅ info: APIデータを確認しました。APIデータをデータベース上のテーブルに登録します。
✅ info: 既存のデータベース "YouTube_API" が見つかりました。新規でデータベースを作成しません。
✅ info: データベース "YouTube_API" に接続しました。
✅ info: 既存のテーブル "tbl_video_info" が見つかりました。一度テーブルを削除した後、再度テーブルを作成します。
✅ info: 既存のテーブル "tbl_video_info"を削除しました。
✅ info: テーブル "tbl_video_info" を作成しました。
✅ info: データベース "YouTube_API" - テーブル "tbl_video_info" に動画情報を登録しました。

<ランキング表>

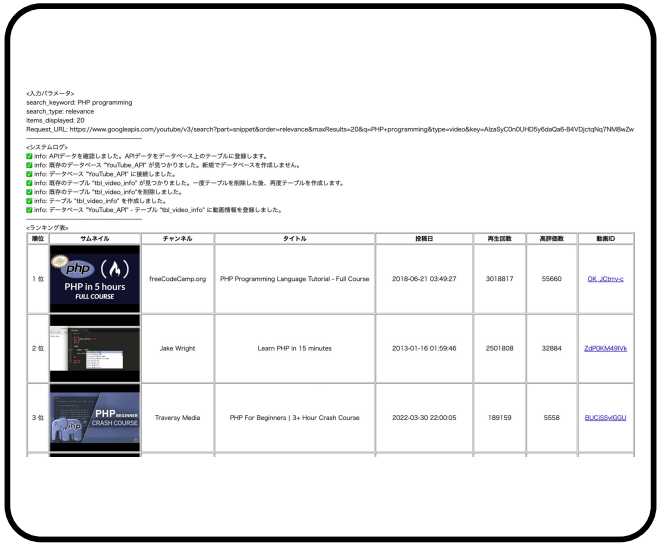
順位	サムネイル	チャンネル	タイトル	投稿日	再生回数	高評価数	動画ID
1 位		Healing and Sleep	【睡眠用BGM - 528Hz】 奇跡が起こる　体がよくなる音楽!! 聞き流すだけで良いことが次々と起こる様に設計されたプロの音楽療法士が作ったヒーリング音楽	DNAの修復、整で熟睡できる周波数入り癒しの…	2022-07-31 19:00:44	54	6	ouu2lbqQ77bg
2 位		幻想世界	【作業BGM/睡眠用BGM】夏を感じるミュージック🎋 soothing music summerBGM	2022-07-31 18:56:43	1	0	D_Pzvb6UMtE
3 位		Station Relax Music	[BGMリラックス] ジブリ音楽	あなたが子供の頃に聞いた曲	いつも何度でも「千と千尋の神隠し」より - Spirited Away OST	2022-07-31 18:33:56	21995	1848	_84Bo3oUBs

図4:検索結果の出力画面

検索キーワードに「BGM」
検索タイプに「投稿日」
表示件数に「10件」と選択すると、
BGMに関連した動画を投稿日が近い順に並べた表が表示される。また、表示件数は10件である。

7. 実行結果(4)

図3:検索用の入力画面



<入力パラメータ>
search_keyword: PHP programming
search_type: relevance
items_displayed: 20
Request_URL: https://www.googleapis.com/youtube/v3/search?part=snippet&order=relevance&maxResults=20&q=PHP+programming&type=video&key=AIzaSyC0n0UHD5y6dxQw6-84VDjctqNq7NMBwZw

<システムログ>
info: APIデータを確認しました。APIデータをデータベース上のテーブルに登録します。
info: 既存のデータベース "YouTube_API" が見つかりました。続開でデータベースを作成しません。
info: データベース "YouTube_API" に接続しました。
info: 既存のテーブル "tbl_video_info" が見つかりました。一度テーブルを削除した後、再度テーブルを作成します。
info: 既存のテーブル "tbl_video_info"を削除しました。
info: テーブル "tbl_video_info" を作成しました。
info: データベース "YouTube_API" - テーブル "tbl_video_info" に動画情報を登録しました。

<ランキング表>

順位	サムネイル	チャンネル	タイトル	投稿日	再生回数	高評価数	動画ID
1 位	PHP in 5 hours FULL COURSE	freeCodeCamp.org	PHP Programming Language Tutorial - Full Course	2018-06-21 03:49:27	3018817	55660	QK_JCtmv-c
2 位		Jake Wright	Learn PHP in 15 minutes	2013-01-16 01:59:46	2501808	32884	ZqPQKM4RVk
3 位	PHP BEGINNERS CRASH COURSE	Traversy Media	PHP For Beginners | 3+ Hour Crash Course	2022-03-30 22:00:05	189159	5558	BUCISSvIGGU

図4:検索結果の出力画面

検索キーワードに「PHP programming」
検索タイプに「キーワードの関連性」
表示件数に「20件」と選択すると、
PHP programmingに最も関連した動画を並べた表が表示される。また、表示件数は20件である。

8. 実行結果(想定と違う場合)

図3:検索用の入力画面

検索タイプに「高評価数」を選択しても、高評価の高い順に表示されない。

→ 後に、高評価数ではなく、総合的な評価？で順番が決まることが分かった。

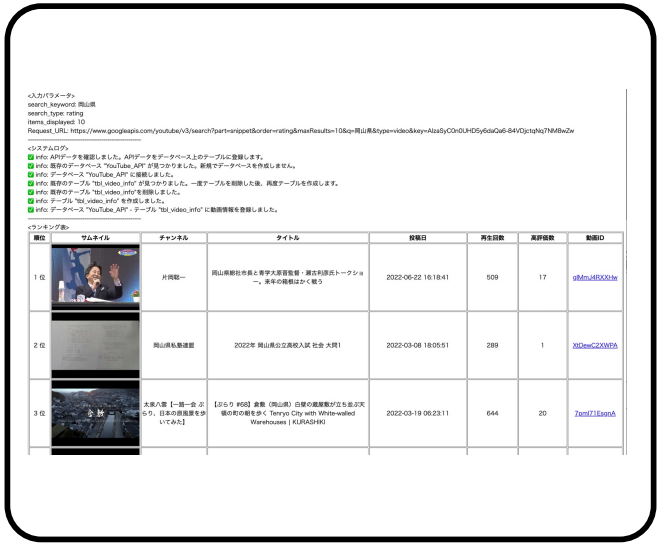


図4:検索結果の出力画面

`order`	`string` `order` パラメータには、API レスポンス内のリソースの並べ替え方法を指定します。デフォルト値は `SEARCH_SORT_RELEVANCE` です。 以下の値を指定できます。 • `date` – リソースを作成日の新しい順に並べます。 • `rating` – リソースを評価の高い順に並べます。 • `relevance` – リソースを検索クエリの関連性が高い順に並べます。このパラメータのデフォルト値です。 • `title` – リソースをタイトルのアルファベット順に並べます。 • `videoCount` – アップロード動画の番号順（降順）にチャンネルを並べます。 • `viewCount` – リソースを再生回数の多い順に並べます。

リファレンス: https://developers.google.com/youtube/v3/docs/search/list?hl=ja

9.アピールポイント

A) SQLのクエリを自動生成するプログラムを制作した。
B) 列（配列の順番）を入れ替えてもエラーを起こさないようにした。
C) 処理ごとに関数化した。
D) 画像の表示と、動画閲覧ページに推移できるようにした。
E) とにかく、汎用性を重視した！

10.予定と実際の相違点

- 予定していたプログラムは、CSSで表示を整えたり、ボタンを押すとテーブルの削除等を可能にするプログラムを作成する予定だった。

- 実際のプログラムは、制作時間の都合上、そこまで到達することができず、ランキング表の表示のみを行うプログラムを制作した。