占いコンテンツをつくろう

WebAPIを利用して問題解決

このワークシートでは，WebAPIを利用した問題解決として「占いコンテンツをつくろう」を学習します。

このワークシートは，パソコンなどで直接編集できるようにWord形式のデータで保存してあります。このワークシートに文字を入力したり，画像を貼り付けたりしながら学習していきましょう。

今回、皆さんは**「いつでも自分で見られる占いコンテンツが欲しいな」**という問題の解決に挑戦します。

それでは、Let’s プログラミング！

・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

【活動１】サンプルプログラムを実行する

下のサンプルプログラムをコピーして，ブラウザ上で開いているGoogle Colaboratory（グーグル コラボラトリー）のセルの中にペーストして，実行してみよう。

|  |
| --- |
| サンプルプログラム |
| ↓下の行からコピーしてください。  import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date][0]  print("星座:", horoscope\_data["sign"])  print("内容:", horoscope\_data["content"])  ↑ここまでの行をコピーしてください。 |

※本プログラムでは、「Web ad Fortune 無料版API」を利用しています。

powered by [JugemKey](http://jugemkey.jp/api/)　／

【活動２】サンプルプログラムを読み解く

サンプルプログラムを読み解いてみましょう。下の表の「説明」の部分に各プログラムの説明が記載されています。

|  |  |
| --- | --- |
| サンプルプログラム | 説明 |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date][0]  print("星座:", horoscope\_data["sign"])  print("内容:", horoscope\_data["content"]) | 各ライブラリを選び，機能を設定する  現在の日付を取得する  占いデータをWebAPIで取得する  必要な星座情報を取得する  取得した情報を表示する |

なお，6行目の [0] は星座を決めるところです。下の表を参考に数値を変えて実行すると，表示させる星座を変えることができます。

|  |  |
| --- | --- |
| No. | 星座名 |
| 0  1  2  3  4  5  6  7  8  9  10  11 | 牡羊座  牡牛座  双子座  蟹座  獅子座  乙女座  天秤座  蠍座  射手座  山羊座  水瓶座  魚座 |

（モジュールについて）  
このプログラムで使用するモジュール（追加機能のようなもの）として次の３つを最初に設定しました。  
requests: HTTPリクエストを送るためのモジュールで、今回はAPIにデータを送信するのに使います。  
json: JSON形式のデータを扱うためのモジュールです。今回は、取得したAPIのデータを使えるようにするために使います。  
datetime: 日付や時刻を扱うためのモジュールです。

【活動３】その他の占い結果を追加する

今回作った見本の占いコンテンツの改善について考えてみましょう。まずは、「星座」「内容」以外の占いの結果の表示にチャレンジです。動画内では，「ラッキーカラー」と「ラッキーアイテム」を追加させました。

|  |
| --- |
| プログラム |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date][0]  print("星座:", horoscope\_data["sign"])  print("内容:", horoscope\_data["content"])  print("ラッキーカラー:", horoscope\_data["color"])  print("ラッキーアイテム:", horoscope\_data["item"]) |

テキストを入力するプログラミングでは，打ち間違えや半角・全角などの入力ミスがよくあります。そのため，デバッグが重要な作業になります。しっかり確認しましょうね。

また，「ラッキーカラー」や「ラッキーアイテム」以外にも，下の表に示すものを表示させることができます。

|  |  |
| --- | --- |
| 占い項目 | プログラムの例 |
| 占いの内容  金運（5段階評価）  仕事運（5段階評価）  恋愛運（5段階評価）  総合運（5段階評価）  ラッキーアイテム  ラッキーカラー  ランキング  星座名 | print("内容:", horoscope\_data["content"])  print("金運:", horoscope\_data["money"])  print("仕事運:", horoscope\_data["job"])  print("恋愛運:", horoscope\_data["love"])  print("総合運:", horoscope\_data["total"])  print("ラッキーアイテム:", horoscope\_data["item"])  print("ラッキーカラー:", horoscope\_data["color"])  print("ランキング:", horoscope\_data["rank"])  print("星座:", horoscope\_data["sign"]) |

【活動４】身近な人が喜んで使えるようにカスタマイズしよう！

更なる改善を考えましょう。今度は，占いコンテンツを「身近な人が喜んで使えるようにカスタマイズする」という問題解決にチャレンジです。今回は友達にしましょう。

「友達を占いで元気にしよう」という課題を設定します。

動画では，星座名やラッキーカラーの表示の方法を変えてみました。

|  |
| --- |
| プログラム |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date][0]  print("Aさんの星座は，", horoscope\_data["sign"], "です。")  print("今日のラッキーカラーは，", horoscope\_data["color"], "です。　洋服や小物にラッキーカラーを取り入れてみませんか？　絶対に似合いますよ！！") |

それでは，皆さんもやってみましょう！身近な人は誰にしますか？どのような表示方法にしてみますか？下の「←」の部分に入力してからプログラムを考えましょう。

|  |  |
| --- | --- |
| 身近な人 | ←ここに入力しましょう。 |

|  |  |
| --- | --- |
| 表示方法 | ←ここに入力しましょう。 |

【活動５】画像の表示にチャレンジしよう！

天の声さんの要望を受けて，画像の表示にチャレンジしてみましょう。動画では，総合運が3以上だったら画像も追加させてみました。

|  |
| --- |
| プログラム |
| import requests  import json  import datetime  from IPython.display import Image, display  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date][0]  print("Aさんの星座は，", horoscope\_data["sign"], "です。")  print("今日のラッキーカラーは，", horoscope\_data["color"], "です。　洋服や小物にラッキーカラーを取り入れてみませんか？　絶対に似合いますよ！！")  if int(horoscope\_data["total"]) >= 3:  image\_url = 'https://drive.google.com/uc?id=1GSvAmmlWF\_T1LezjYhcty9ZPDWTfGHMv&.jpg'  display(Image(url=image\_url)) |

この活動では，画像を表示させるために，4行目と11行目以降を追加しています。上手に画像が表示されない場合は，星座の番号（7行目の[0]）を変えてみてもいいかもしれません。

11行目では「もし，～～ならばのifのプログラム」や「画像に関するURL」を入力しています

「画像に関するURL」は，URL内のアイディ イコール（id=）のあとに，ID & 拡張子を入力しています，

プログラムの例に沿って，IDと拡張子の部分を入力してみましょう。

　みなさんが普段目にする占いコンテンツには、どんな文字、画像、音声などの情報が表示されていますか？それらを追加したり、表し方を工夫したりして、あなたが対象としているひとが喜ぶようなコンテンツにしてみましょう！

【活動６】プログラムの基本的な構造について知る

今回は，占いプログラムをつくってみました。他にもプログラムをつくれるようになるためには，プログラムの構造には，型があることを知っておくと便利です。

|  |  |
| --- | --- |
| 順次 | サンプルプログラムでは，モジュールの指示から結果を表示させるところまでは，順番に命令を処理していました。身近な人が喜んで使えるようにカスタマイズした占いコンテンツのプログラムも同様です。  この構造を，「順次」といいます。 |
| 分岐 | 画像の表示にチャレンジし，操作性を向上させたプログラムは，ifを入れ，条件に応じて処理が変わるようになっています。  この構造を「分岐」といいます。 |

ここからは，応用編です。問題解決の学習を進めましょう。

【活動１】問題の発見

「動画で一緒に作った占いコンテンツは，使いづらいなぁ～」としたら，原因は何が考えられますか？下の記入欄に考えられる原因を入力してみましょう。

|  |  |
| --- | --- |
| 原因の記入欄 | 〇←ここに入力しましょう。  〇←ここに入力しましょう。 |

例えば，先生はこう考えました。下の〇以降の文字を黒色に変えてみてください。

|  |  |
| --- | --- |
| 先生が考えた原因 | 〇星座の番号が知らないと，占うことができない。  〇日付が分からない。 |

【活動２】課題の設定

次に，みなさんが発見した問題の原因を解決する方法を考えましょう。原因を解決するために，プログラムにどんな工夫をすればよいのかを考えてみましょう。

|  |  |
| --- | --- |
| 工夫点記入欄 | 〇←ここに入力しましょう。  〇←ここに入力しましょう。 |

先生の場合はこう考えました。下の〇以降の文字を黒色に変えてみてください。

|  |  |
| --- | --- |
| 先生が考えた工夫点 | 〇プログラム内に星座名を入力すると，その星座の占い結果が表示される。  〇占い結果に日付を表示させる。 |

応用編

【活動３】問題の原因の解決

みなさんが考えた工夫点を盛り込んだ占いコンテンツをつくってみましょう。みなさんがどんなプログラムを作るのか楽しみです。プログラムが完成したら，この下にプログラムの画像を貼り付けてみてください。

ちなみに，先生が考えたプログラムを作ってみたい人は，次のページを参考にしてください。

ここにプログラムの画像を貼り付けましょう！

【先生が作ったプログラム】

先生は，7行目以降を工夫しました。今回は「魚座」と入力してみました。みなさんは，できましたか？

|  |
| --- |
| プログラム |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/'+ date)  horoscope\_data = res.json()["horoscope"][date]  for data in horoscope\_data:  if data["sign"] == "魚座":  print("日付:", date)  print("星座:", data["sign"])  print("内容:", data["content"])  print("ラッキーカラー:", data["color"])  print("ラッキーアイテム:", data["item"]) |

…え？動画で「星座名を入力する」と言っていたのは、使う人がどの星座なのかを、プログラムを実行したときに聞くようにすることじゃないか…？ですって？

ほほう。ユーザーの入力に応じた応答、つまり双方向性のある占いコンテンツのプログラムをつくりたいのですね？

承知しました。では、次のページにいきましょう。

ユーザーの入力への応答を実現するために、１つ練習をしてみましょう。

先生は「すべての占い結果がみたいなぁ～」と思い，サンプルプログラムを改良してみました。みなさんも挑戦してみてください。このプログラムは，7行目に「反復」の構造をもっています。

【12星座すべての占い結果がわかるコンテンツ】

|  |
| --- |
| プログラム |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/' + date)  horoscope\_data\_list = res.json()["horoscope"][date]  for horoscope\_data in horoscope\_data\_list[:12]:  print("星座:", horoscope\_data["sign"])  print("内容:", horoscope\_data["content"])  print("----------") |

|  |  |
| --- | --- |
| 反復 | 7行目からは「星座」「内容」「----------（仕切り線）」を表示させる際に，リストから12星座分を繰り返す命令が書かれています。  この構造を，「反復」といいます。 |

【先生がつくったユーザーの入力した星座に応じた応答をするプログラム】

　さきほどの反復プログラムで使った命令「for」を使って次のプログラムをつくってみました。Google Colaboratory（グーグル コラボラトリー）のセルの中にペーストして，実行してみたあとに、プログラムを読み解いてみましょう。

|  |  |
| --- | --- |
| プログラム | 説明 |
| import requests  import json  import datetime  date = datetime.datetime.today().strftime("%Y/%m/%d")  input\_sign = input("星座名を入力してください（例：牡羊座、乙女座）: ")  res = requests.get(url='http://api.jugemkey.jp/api/horoscope/free/' + date)  horoscope\_data = res.json()["horoscope"][date]  found = False  for data in horoscope\_data:  if data["sign"] == input\_sign:  print("日付:", date)  print("星座:", data["sign"])  print("内容:", data["content"])  print("ラッキーカラー:", data["color"])  print("ラッキーアイテム:", data["item"])  found = True  break  if not found:  print("入力された星座が見つかりませんでした。星座名を正しく入力してください。") | 星座名を入力する  foundの変数をFalseにする  もし入力した星座が存在する星座であれば、以下のデータを表示する  foundの変数をTrueに変え繰り返しから抜ける  もし星座が見つからなかったら、エラーメッセージを出す |

found = False という行は、このプログラムにおけるフラグとして機能します。フラグを立てる、という表現を聞いたことがありませんか？これは、特定の条件が満たされたかどうかを追跡するために使用する変数として準備しました。

今回のプログラムでは、ユーザーが入力した星座が見つかったかどうかを追跡します。

プログラムの流れは以下のようになっています。

foundは最初にFalseに設定されています。これは、まだ星座が見つかっていないことを意味します。

その後、プログラムはユーザーによって入力された星座名を占いデータと照合します。0番から11番の星座まで、順番に入力された星座が当てはまるかを繰り返し照合するのです。そのため、forの命令、つまり繰り返しの命令が使われています。

もしユーザーが入力した星座が占いデータの中にあれば、占いの結果の中から、表示するよう命令されたデータを画面に表示し、found は True に設定されます。（フラグを立てます）

繰り返しが終了した後、foundの値をチェックします。もしfoundがまだFalse であれば（フラグが立っていなかったら）、それはプログラムがユーザーの入力した星座を見つけられなかったことを意味し、ユーザーに対してエラーメッセージを返します。

もし、このように星座が見つからなかった場合の動作をプログラムしておかないと、12星座以外の文字が入力された時、プログラムが繰り返しから抜けられなくなり、永遠にループしたままになってしまいます。

（試してみよう）

|  |
| --- |
| if not found:  print("入力された星座が見つかりませんでした。星座名を正しく入力してください。")  を削除して、プログラムを実行し、12星座以外の文字をいれてみよう。（例：ぎょう座）  **↓**  どうなりましたか？  アイコン  自動的に生成された説明☜このように、実行ボタンがくるくる回ったまま止まらなくなります。このままだと永遠に終わらないので、■停止ボタンを押してプログラムを中断しましょう。 |

　私たちが普段使っているコンテンツにも、このような、うまく実行できないときにエラーであることを知らせ、プログラムを中断する**「エラー処理」**のプログラムの工夫が隠れされています。（例：ブラウザソフトで、目的のページが見つからなかった場合に表示される404 Not Foundのページ）

　あなたがつくるコンテンツのページにも、エラー処理のプログラムを実装しましょう。

みなさんも、ぜひいろいろな楽しいコンテンツをつくってください。