

テンプレートタグ

- テンプレート内では、特別な意味をもつタグを利用できます
 - 「{% ... %}」という形式
 - テンプレートタグを使うことでview関数での処理を簡潔にできる
- 以下の3つがよく利用される
 - 条件分岐
 - ループ処理
 - URLの埋め込み
- ※詳細はCS概論 MOOCs 10-1を参照してください

7. forecast/views.py とテンプレートの修正

- forecast/views.pyとテンプレートを以下のように修正します
 - random.choicesを使い、weathersリストから3個(k=3)ランダムに復元抽出し、リストとしてparams['forecasts']に格納します
(例えば["Rainy", "Rainy", "Sunny"])というように、weathersからランダムに3個抽出されたリストが生成されます)

```
from django.shortcuts import render
import random

# Create your views here.
weathers = ["Sunny", "Rainy", "Cloudy", "Snowy"]
def index(request):
    params = {}
    params['title'] = '3 days forecast'
    params['forecasts'] = random.choices(weathers, k=3)
    return render(request, 'forecast/index.html', params)
```

forecast/views.py

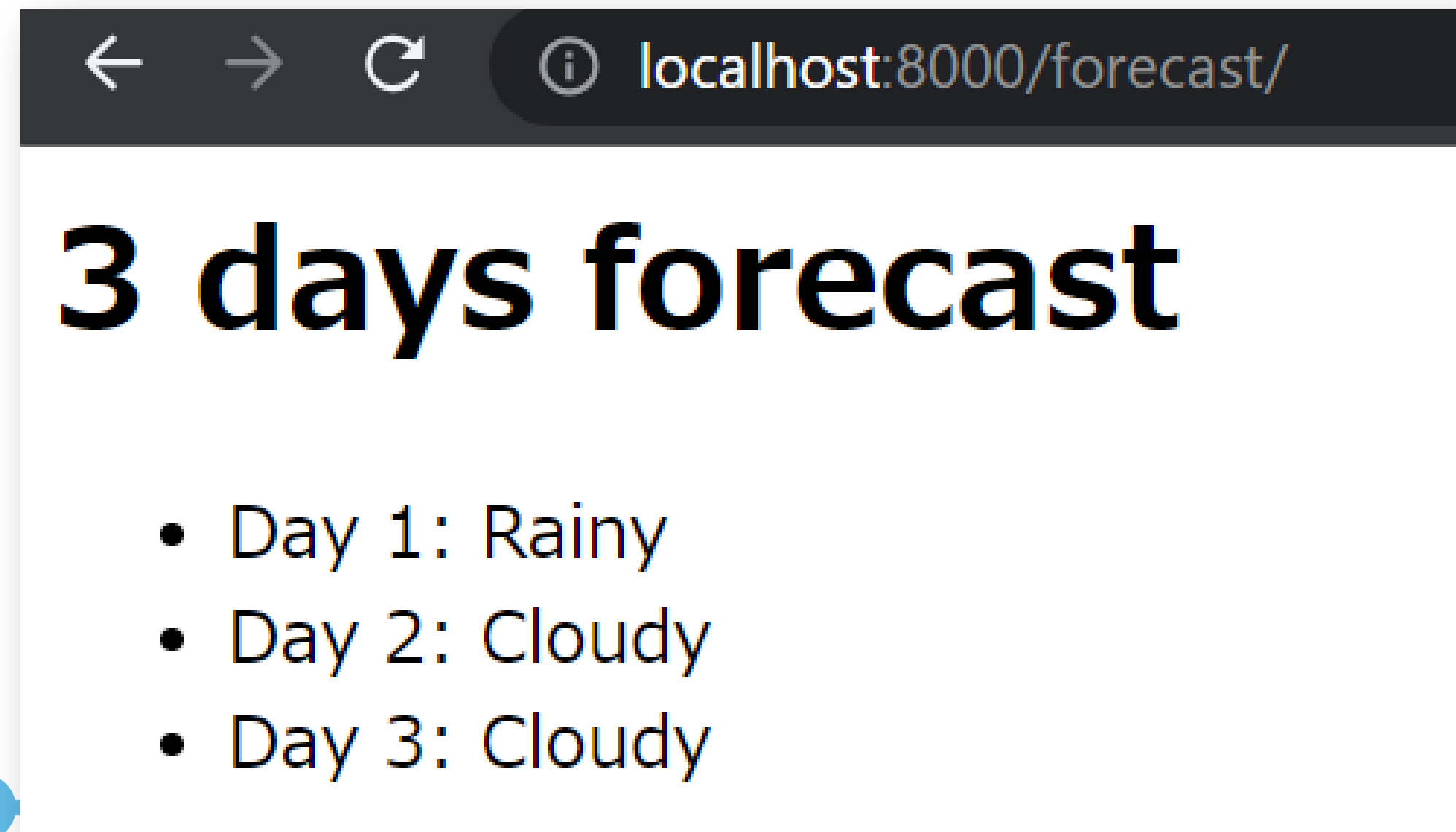
8. テンプレートの修正（ループ）

- forecast/templates/forecast/index.htmlを右のように修正します
 - for は、基本的にPythonのfor(イテレーター)と同じです
 - {% 要素 in 要素群 %} : 要素群から要素を取りだしてループ
 - {% endfor %} : ループ終了位置の目印
 - ループの回数は forloop.counter 変数に自動的に保存されます
 - ループの回数を1から始まる数字で取得できます

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <title>Template</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>{{title}}</h1>
    <ul>
      {% for item in forecasts %}
        <li>Day {{forloop.counter}}: {{item}}
        </li>
      {% endfor %}
    </ul>
  </body>
</html>
```

9. 動作確認

- manage.pyがあるディレクトリで、以下のコマンドを入力し、djangoを起動します
 - python manage.py runserver
 - ※ mac はpython3 manage.py runserver
- Chromeで定義したURLにアクセスし、表示を確認してみましょう
 - ※確認が終わったら、Ctrl-Cで停止して次の手順に進むこと



<http://127.0.0.1:8000/forecast/>

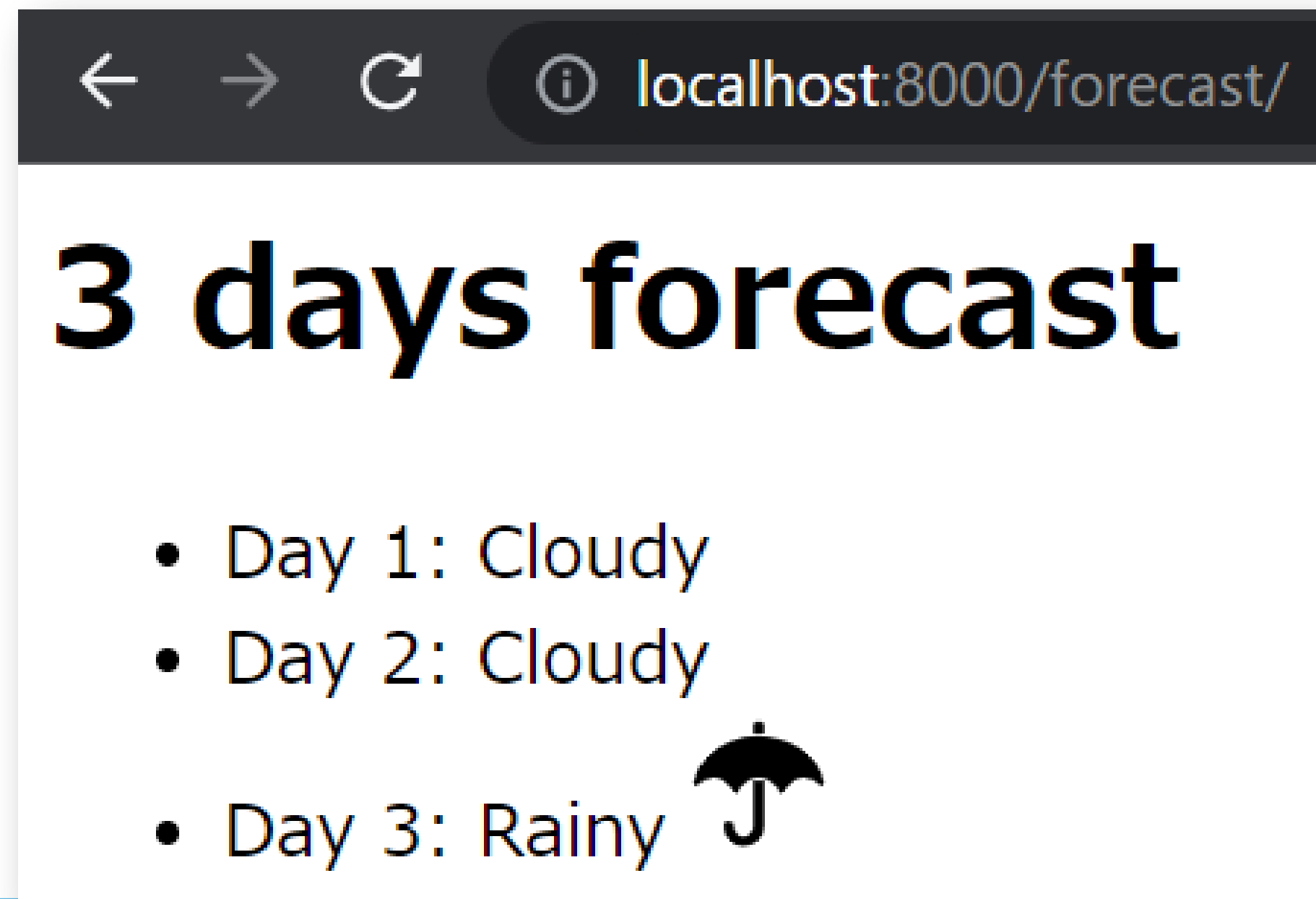
10. テンプレートの修正（条件分岐）

- 雨の予報に傘マークをつけましょう
- forecast/templates/forecast/index.htmlを右のように修正します
 - 条件分岐はpythonのif/elseと同じように利用できます。elseは省略可能。
 - {% if 条件 %} : 条件がTrueなら続くテンプレートを実行
 - {% else %} : 条件がFalseのときに実行するテンプレートを記述
 - {% endif %} : ifおよびelseの終了位置の目印

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <title>Template</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>{{title}}</h1>
    <ul>
      {% for item in forecasts %}
        <li>Day {{forloop.counter}}: {{item}}
          {% if item == "Rainy" %}
            
          {% endif %}
        </li>
      {% endfor %}
    </ul>
  </body>
</html>
```

11. 動作確認

- manage.pyがあるディレクトリで、以下のコマンドを入力し、djangoを起動します
 - python manage.py runserver
 - ※ mac はpython3 manage.py runserver
 - ※確認が終わったら、Ctrl-Cで停止して次の手順に進むこと
- Chromeで定義したURLにアクセスし、表示を確認してみましょう



<http://127.0.0.1:8000/forecast/>

12. テンプレートの修正 (URL埋め込み)

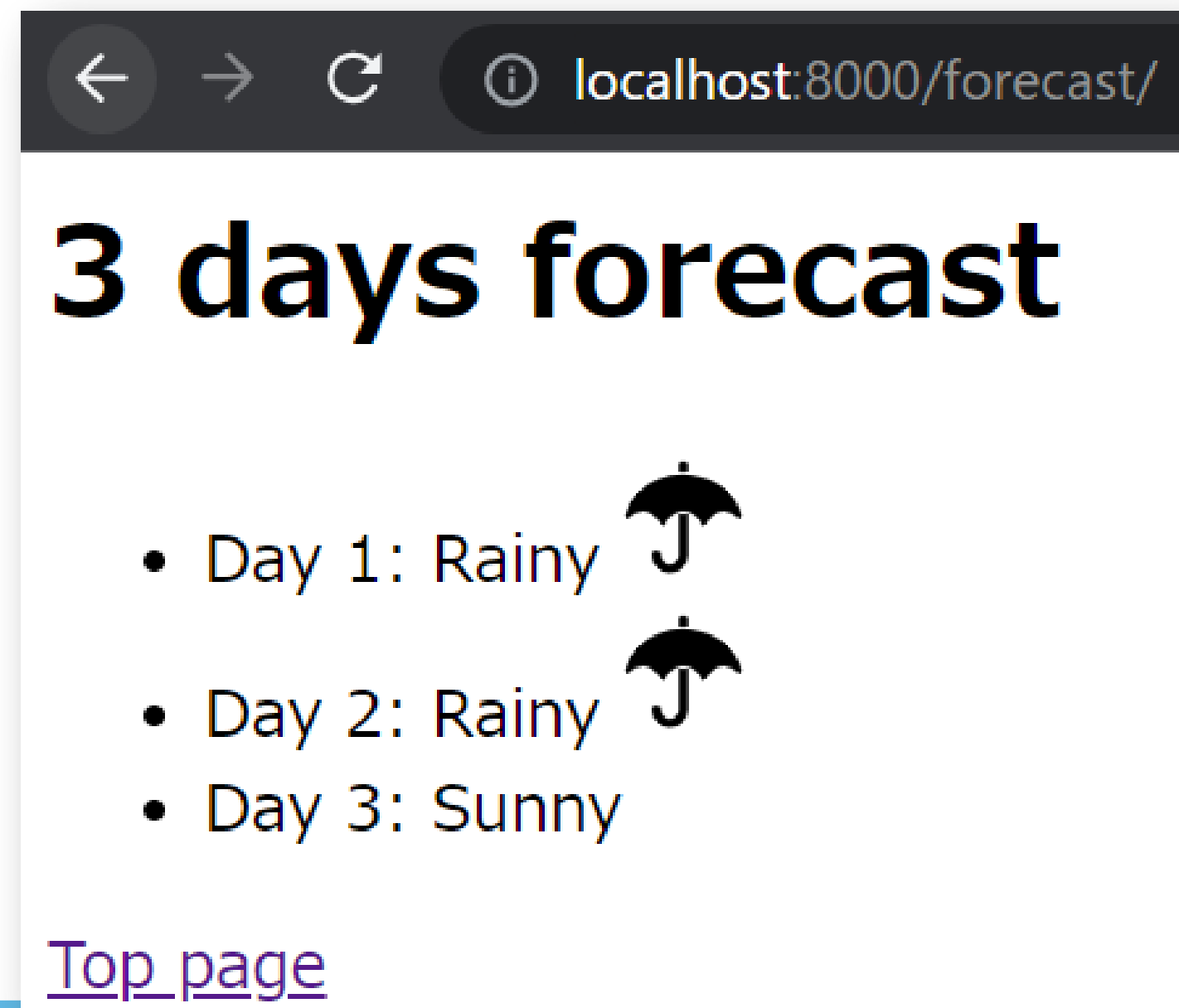
- トップページへのリンクをつけましょう
- forecast/templates/forecast/index.htmlを右のように修正します
 - {% url "urls.pyで定義済みの名前" %} とすることで、urls.pyでページにつけた名前からテンプレート内で使えるリンクを自動的に作れます

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <title>Template</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>{{title}}</h1>
    <ul>
      {% for item in forecasts %}
        <li>Day {{forloop.counter}}: {{item}}
          {% if item == "Rainy" %}

          {% endif %}
        </li>
      {% endfor %}
    </ul>
    <p><a href="{% url 'root' %}">Top page</a></p>
  </body>
</html>
```

13. 動作確認

- manage.pyがあるディレクトリで、以下のコマンドを入力し、djangoを起動します
 - python manage.py runserver
 - ※ mac はpython3 manage.py runserver
 - ※確認が終わったら、Ctrl-Cで停止して次の手順に進むこと
- Chromeで定義したURLにアクセスし、表示を確認してみましょう



<http://127.0.0.1:8000/forecast/>

課題3

- ここまでを実現したforecast/views.py とindex.htmlを提出してください

発展課題) 課題4

- これまで実現したプログラムに以下に例示するような任意の拡張を実現可能な範囲で行い、実現した内容の概要と forecast/views.py とテンプレートファイルを提出してください
 - 天気予報の日数を1週間にする
 - Bootstrapを適用してレイアウトをきれいにする
 - 天気のアイコンを増やす
 - 演習で使った傘マークは icongram のFont Awesomeから取得したものです
<https://icongr.am/>
 - Open Weather Map のAPIから実際の天気予報を取得する
 -