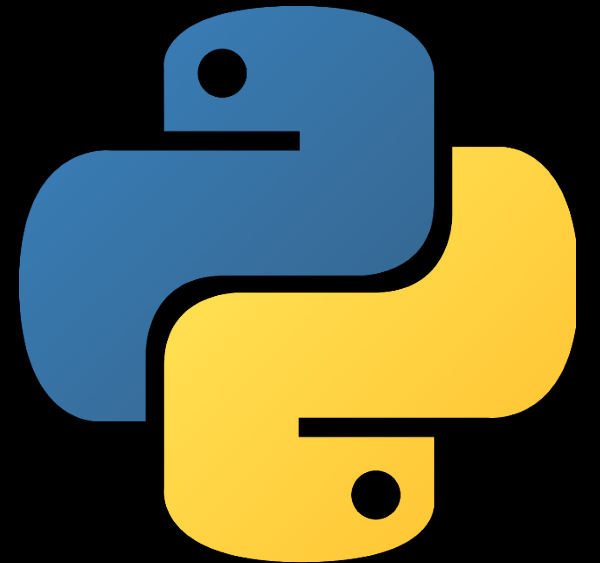


# Episode X: The Python Way Of The Force

May the web be with you





# STAR WARS - VOTE

Please choose side

Empire



Rebels



Vote

Current standing

Empire

14

Rebels

12

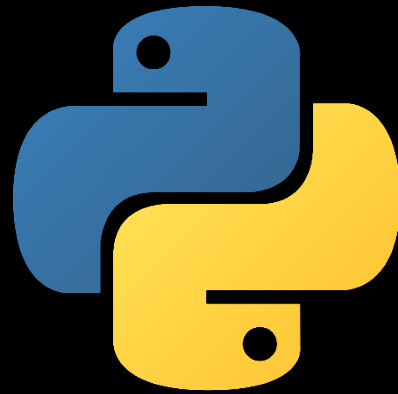


**Build our web site, you will**





**THE FORCE  
IS STRONG  
WITH THIS ONE**



# Dagens agenda

- Demo på vad vi ska bygga
- Bygga webbsidor med Python
  - Hur fungerar en webserver?
  - Webserver med Python
  - Server => Python
  - Klient => HTML/CSS
- Bottle som mikroramverk
  - Vad är ett mikroramverk?
  - Routes => Funktioner
- Bygga vår webbplats



**Vad vi ska bygga**





Important, this is!



# Webbserver?



**En webserver är antingen ett datorprogram som tillhandahåller webbsidor för en viss webbplats eller en serverdator på vilken sådan programvara körs.**

I vårt fall ett datorprogram – skrivet i Python

# Webbserver

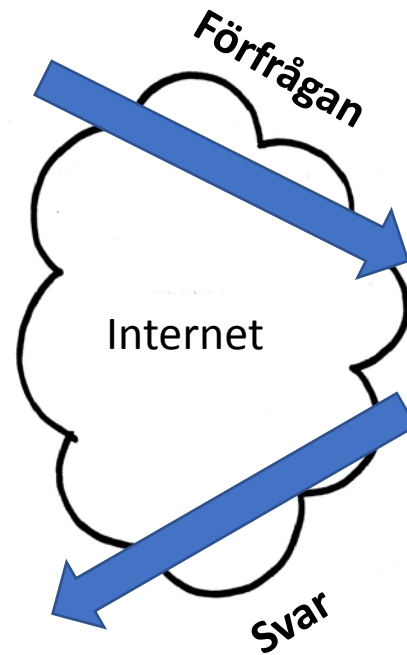
- Ett webbserverprogram har som uppgift att tillhandahålla webbsidor och andra filer via datakommunikationsprotokollet HTTP eller HTTPS, **vanligen över Internet.**
- Det förekommer också att **en enskild användare** kör en webbserver för eget lokalt bruk på sin egen dator. Vanligen kommunicerar användaren med webbservern med hjälp av en webbläsare.
- Användaren väljer webbsidor och webbläsaren beställer webbsidorna från webbservern och visar dem på användarens datorskärm. Webbläsaren är klient till webbservern.



Klient



Klient



Server





There's no place like

127.0.0.1



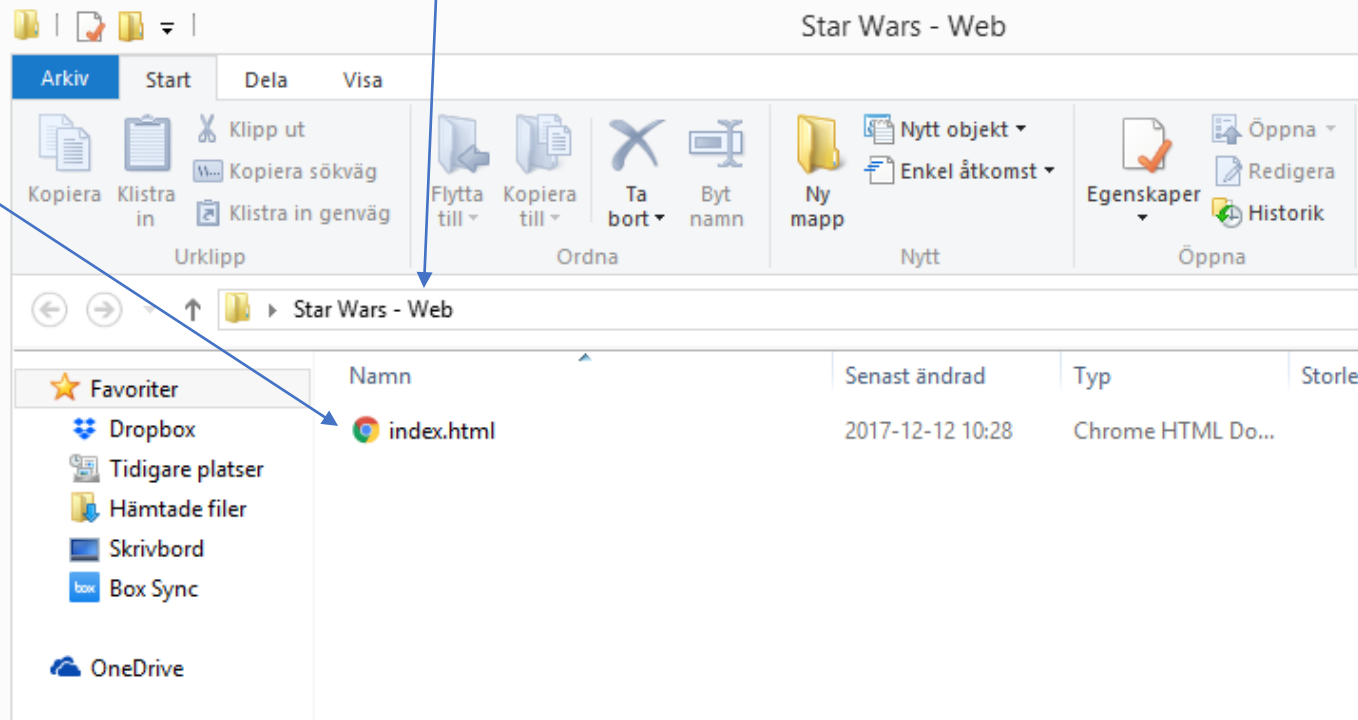
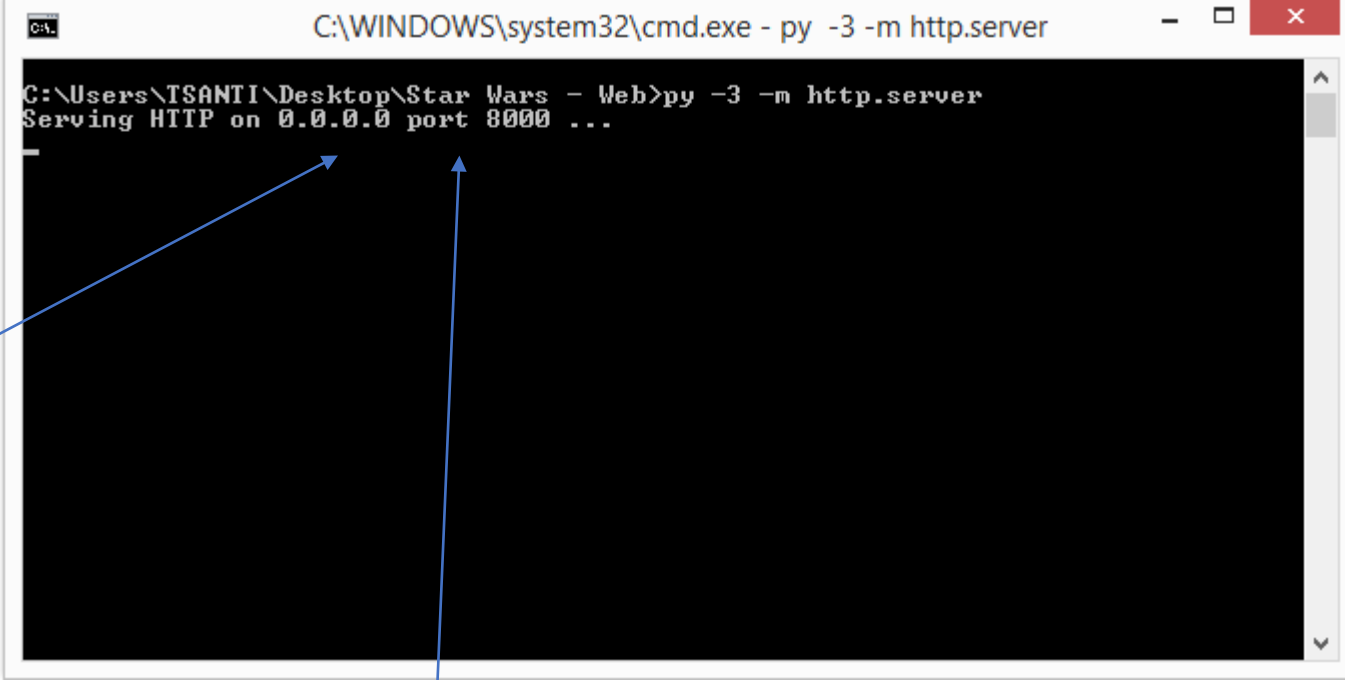
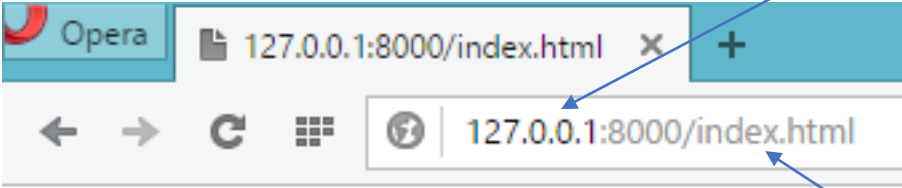
No place like 127.0.0.1, there is

# py -3 -m http.server

- Att starta en webbrowser från en specifik mapp genom Python (från konsolen/terminalen)



# Star Wars



DO, OR DO NOT...  
THERE IS NO TRY

-YODA





# Men – hur mycket Python var detta egentligen?



*Feel the power of the server side!*

# STAR WARS - VOTE

Please choose side

Empire



Rebels



Vote

Current standing

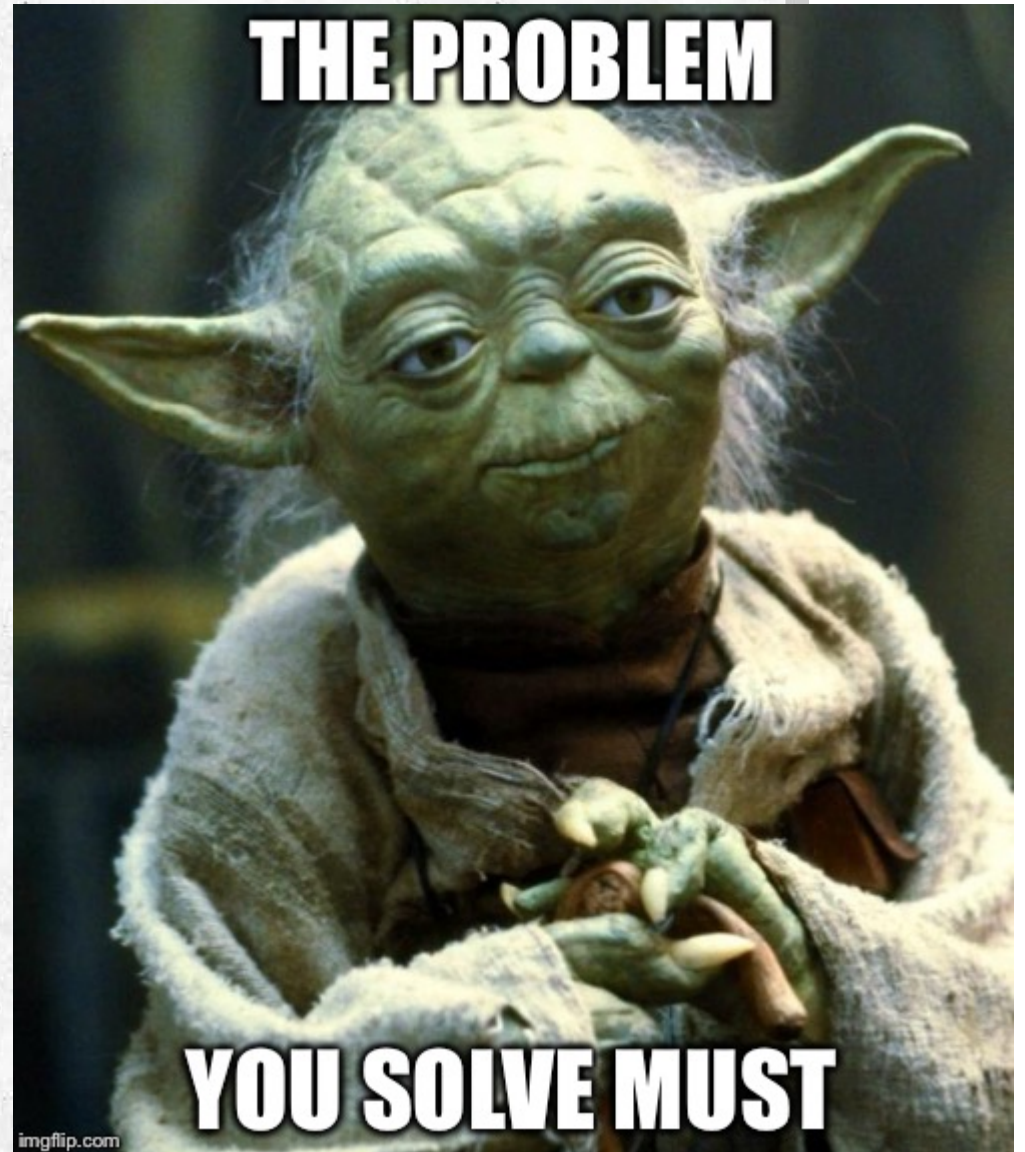
Empire

14

Rebels

12

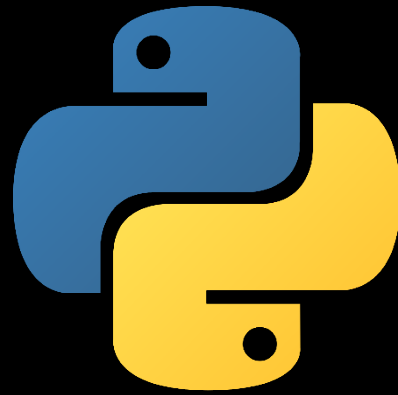
THE PROBLEM



YOU SOLVE MUST

imgflip.com

**THE FORCE  
IS STRONG  
WITH THIS ONE**



# Vad kan vi hittills?

- Skapa webbsidor
  - HTML
  - CSS
  - (JavaScript)
- Programmera i Python
  - Variabler
  - Funktioner
  - Läsa/skriva till/från filer
  - M.m.
- Vad behöver vi lära oss? => Bygga en webbtjänst!





# Behöver vi bygga allt själva?



A simpler way, there is



**To be efficient, a tool you need.**



# Mikroramverk

- En samling funktioner för att bygga webbplatser, med fokus på
  - Snabbt
  - Smidigt
  - Resurssnålt
- Exempel på mikroramverk
  - Bottle (Python)
  - Flask (Python)
  - Slim (PHP)
  - Silex (PHP)
  - Camping (Ruby)
  - Sinatra (Ruby)





# Bottle – Ett mikroramverk i Python

- Bottle har många inbyggda funktioner, men fokuserar på:
  - **Routing**      Mappa URL:er mot funktioner
  - **Templates**    Skapa mallar (HTML) för att presentera innehåll
  - **Utilities**      Tillgång till data, ladda upp filer, cookies, HTTP-möjligheter, m.m.
  - **Server**        Inkluderar en inbyggd HTTP-server
- Bottle är endast en enda fil
- Bottle använder bara standard-biblioteket för Python
- Man installera bottle genom PIP, easy\_install, eller laddar hem py-filen.

# Hur fungerar Bottle?

Jämfört med en "vanlig" webbsida

# Koncept 1

## - Routes



# Bottle - Routes

- Vi vill kunna mappa URL till olika funktioner i Python
  - /home           =>     def home():
  - /contact       =>     def contact():
  - /about          =>     def about():

127.0.0.1/home

Starwarsvote.com/home

Thedarkside.com/home

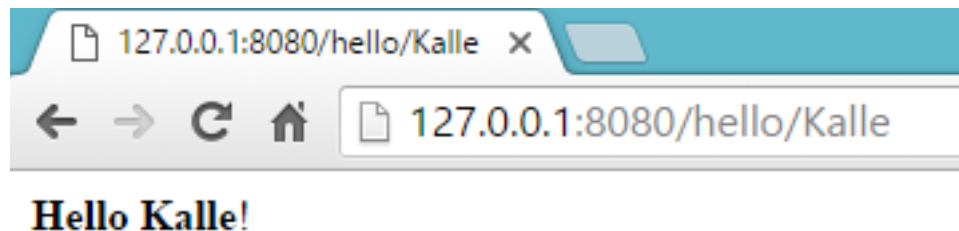
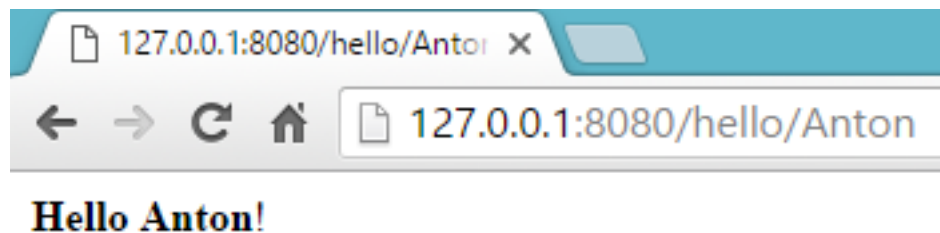
**Kom igång med bottle!**

# Hello <name>

```
from bottle import route, run, template

@route('/hello/<name>')
def index(name):
    return template('<b>Hello {{name}}</b>!', name=name)

run(host='localhost', port=8080)
```





**Att använda sig  
utav flera routes...**

```
from bottle import route, run, error

@route('/')
def index():
    return "Startsida"


@route('/contact')
def contact():
    return "Kontaktsida"

@route('/about')
def about():
    return "Om oss-sida"

@error(404)
def error404(error):
    return "Denna sida finns inte..."

run(host='localhost', port=8080)
```

**Demo – olika routes**

A man in a dark suit, white shirt, and purple tie is holding a white rectangular sign with his right hand. The sign has the text "Make a good first impression" written on it in a black, sans-serif font. The background is plain white.

Make a  
good first  
impression

# Templates - Mallar för vår information



*Join the dark side, we have the prettiest web sites!*

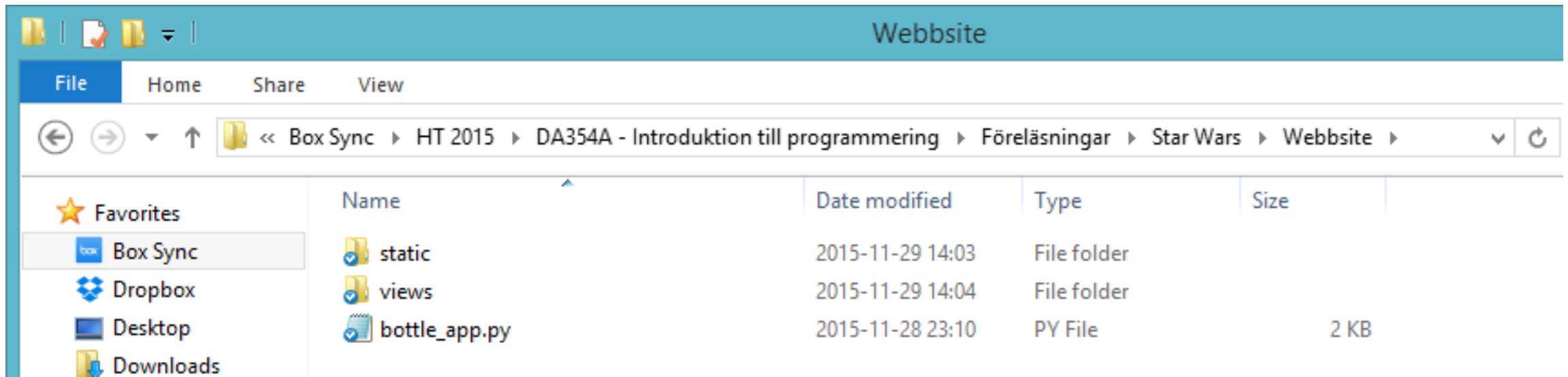
# Bottle - Templates

- Templates är mallar för hur vi ska presentera vår information
  - HTML
  - CSS
  - JavaScript
- Vi kan skicka data (bearbetad utav Python) till våra malla för att skapa dynamiska webbsidor



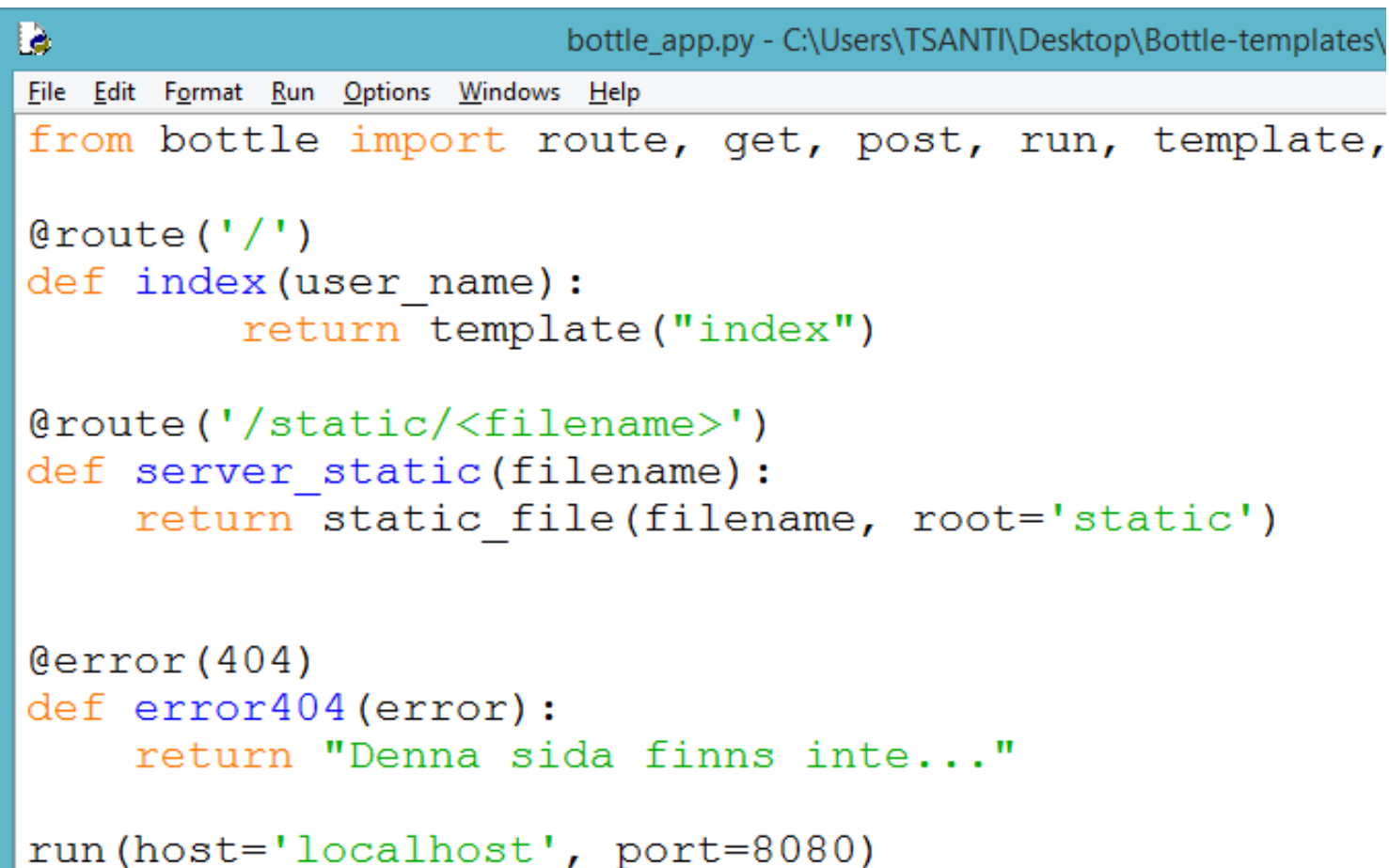
# Bottle – Templates och statiska filer

- Alla templates ska:
  - Ligga i mappen "view"
  - Ha filändelsen \*.tpl eller \*.html
- Statiska filer (bilder, css, js, etc.) ska:
  - Ligga i mappen "static"



# Hur ser detta ut i bottle?

```
<html>
  <head>
    <title>Star Wars</title>
    <meta charset="utf-8">
    <link href="/static/style.css" rel="stylesheet">
  </head>
  <body>
    <h1>Star Wars!</h1>
    <footer>
      Made by the force
    </footer>
  </body>
</html>
```



The screenshot shows a Python IDE window titled "bottle\_app.py - C:\Users\TSANTI\Desktop\Bottle-templates\". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code defines a Bottle application with three routes: a root route returning an "index" template, a static file route, and a 404 error handler. The application is run on localhost:8080.

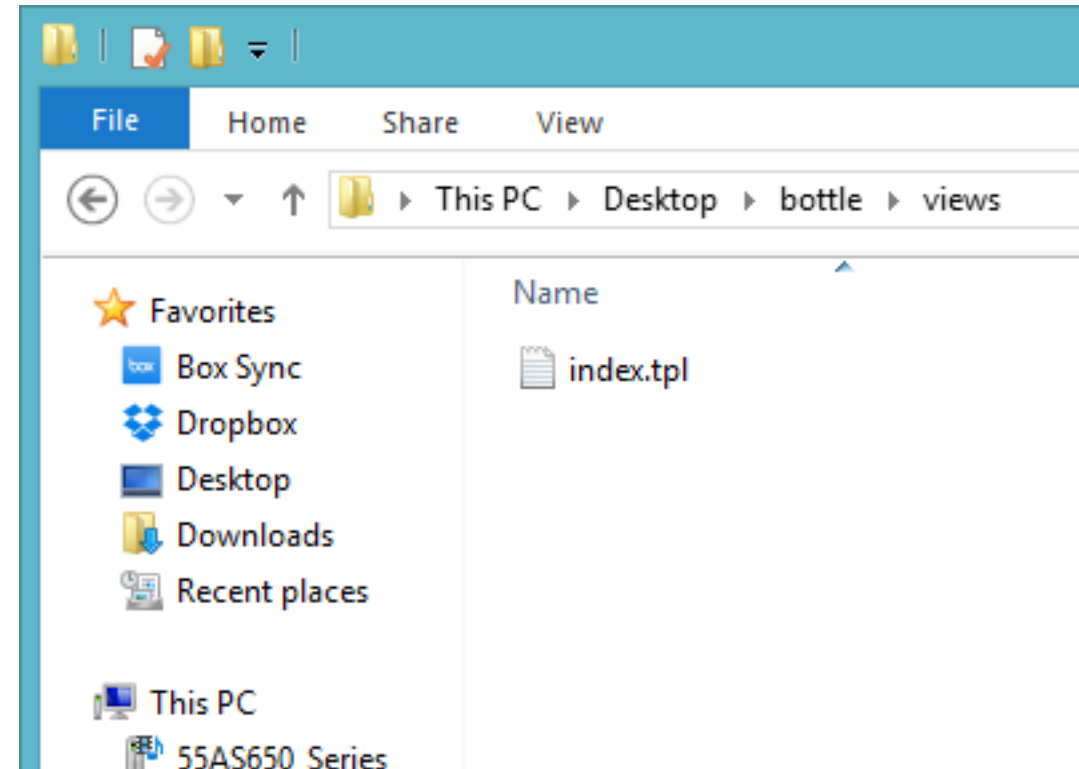
```
from bottle import route, get, post, run, template,

@route('/')
def index(user_name):
    return template("index")

@route('/static/<filename>')
def server_static(filename):
    return static_file(filename, root='static')

@error(404)
def error404(error):
    return "Denna sida finns inte..."

run(host='localhost', port=8080)
```



**Demo - templates**

**Nu till vårt projekt**

# STAR WARS - VOTE

Please choose side

Empire



Rebels



Vote

Current standing

Empire

14

Rebels

12

# Routes => Vilka behöver vi?

Route	Syfte	Metod	Template
/	Till vår startside	GET	index
/vote	Rösta på en sida	POST	index
/disqus	Visa gästbok	GET	disqus
/new-post	Skriv ett inlägg	POST	disqus
*error404*	En sida som inte finns	Alla	error



```
*Untitled*
File Edit Format Run Options Windows Help
from bottle import route, get, post, run, template, error, static_file, request

@route('/')
def index():
    # View start page
    return template("index")

@route('/vote', method='POST')
def vote():
    # Save vote and view start page
    return template("index")

@route('/disqus')
def disqus():
    # View disqus page
    return template("disqus")

@post('/disqus-post', method='POST')
def save_post():
    # Save new disqus post and view disqus page
    return template("disqus")

@route('/static/<filename>')
def server_static(filename):
    return static_file(filename, root='static')

@error(404)
def error404(error):
    return template("error")

run(host='localhost', port=8080, debug=False)
```

# Testa så att allt fungerar



*Let the battle begin*



**1. Fixa röstningen**

**2. Fixa gästboken**

# Röstningen

- Spara alla röster i filen "votes.txt"
- Spara rösterna som JSON-format, enligt följande mall:

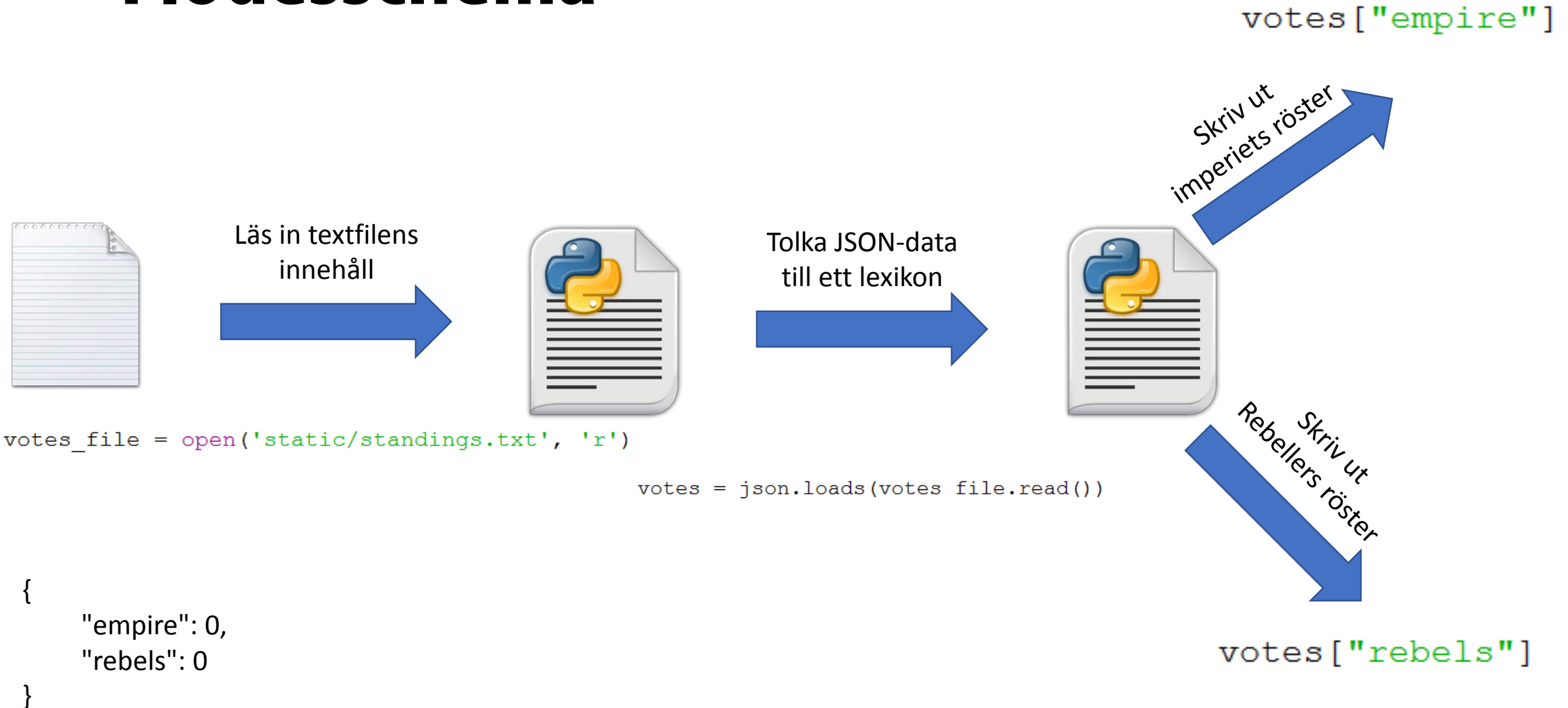
```
{  
  "empire": 0,  
  "rebels": 0  
}
```
- Vi kan omvandla ett lexikon i Python till JSON genom:
  - `json_votes = json.dumps(votes)`
- Vi kan omvandla JSON till Python-datatyper genom:
  - `Votes = json.loads(json_votes)`



{JSON}



# Flödesschema





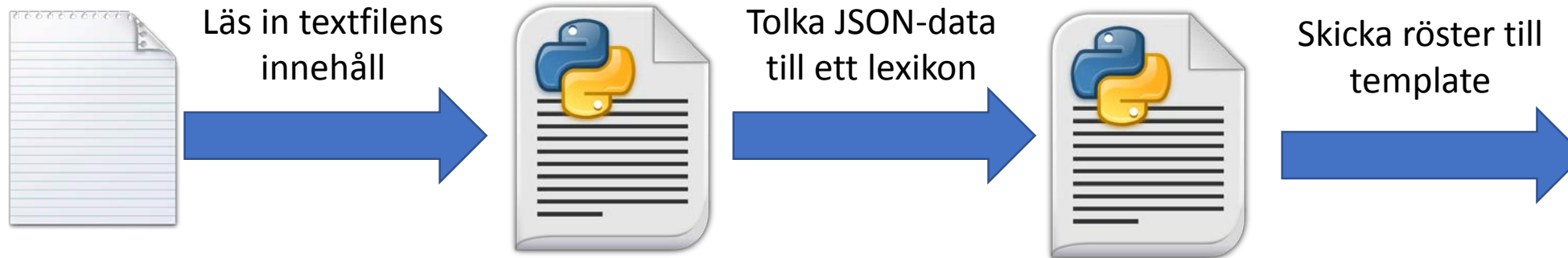
# Flödesschema

Webbsida på internet

Skriv ut  
imperiets röster



Skriv ut  
Rebellers röster



**Nu bygger vi! ;)**

## **2. Fixa gästboken**

**Frågor?**