

Web Development & Python (Not another tutorial)

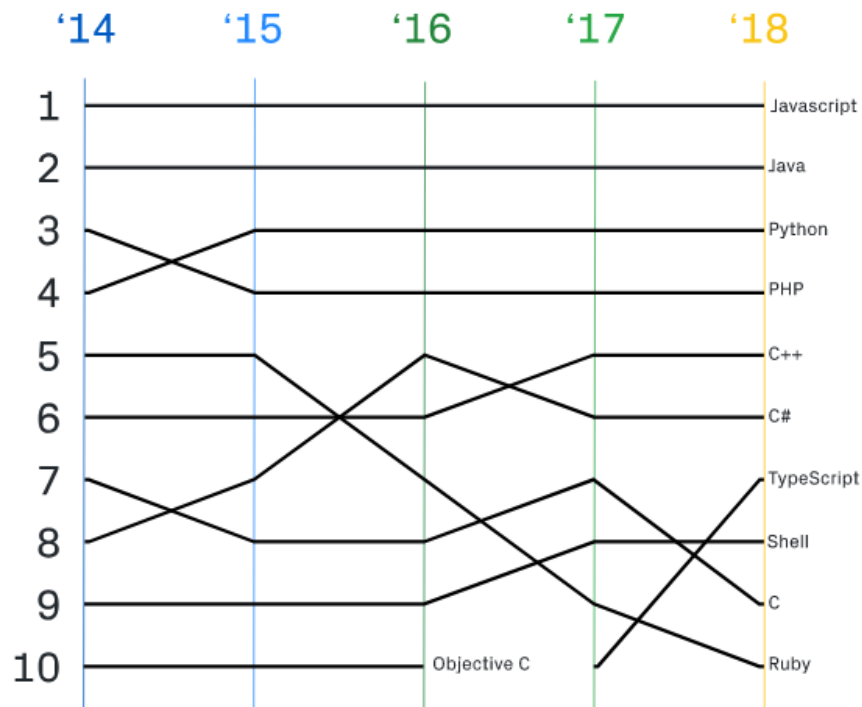
G Sharmila

Blog: <https://www.minvolai.com>

Sharmila.gopirajan@gmail.com

Founder: Sitefitnesshq.com

Python The Language



Growth in contributors	
1 Kotlin	2.6x
2 HCL	2.2x
3 TypeScript	1.9x
4 PowerShell	1.7x
5 Rust	1.7x
6 CMake	1.6x
7 Go	1.5x
8 Python	1.5x
9 Groovy	1.4x
10 SQLPL	1.4x

Web Development in Python

Data analysis



Web development



Machine learning



System administration / DevOps / Automation scripts



Programming of web parsers / scrapers / crawlers



Educational purposes



Software prototyping



Software testing / writing automated tests



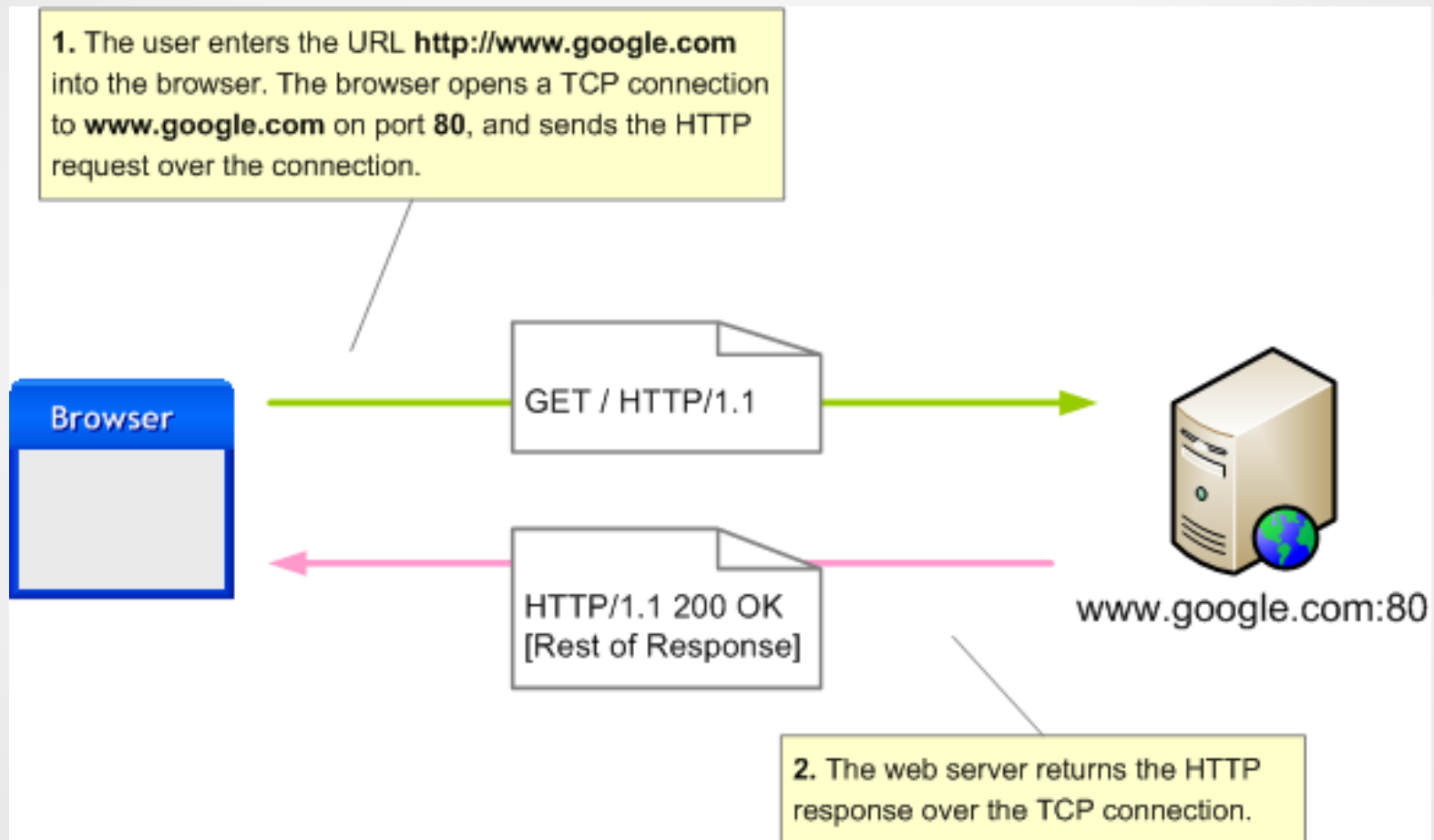
Network programming



Desktop development



What is a webserver?



HTTP Protocol

SAFE METHODS NO ACTION ON SERVER	GET	HTTP/1.1 MUST IMPLEMENT THIS METHOD
	HEAD	INSPECT RESOURCE HEADERS
MESSAGE WITH BODY SEND DATA TO SERVER	PUT	DEPOSIT DATA ON SERVER — INVERSE OF GET
	POST	SEND INPUT DATA FOR PROCESSING
	PATCH	PARTIALLY MODIFY A RESOURCE
	TRACE	ECHO BACK RECEIVED MESSAGE
	OPTIONS	SERVER CAPABILITIES
	DELETE	DELETE A RESOURCE — NOT GUARANTEED

http.server

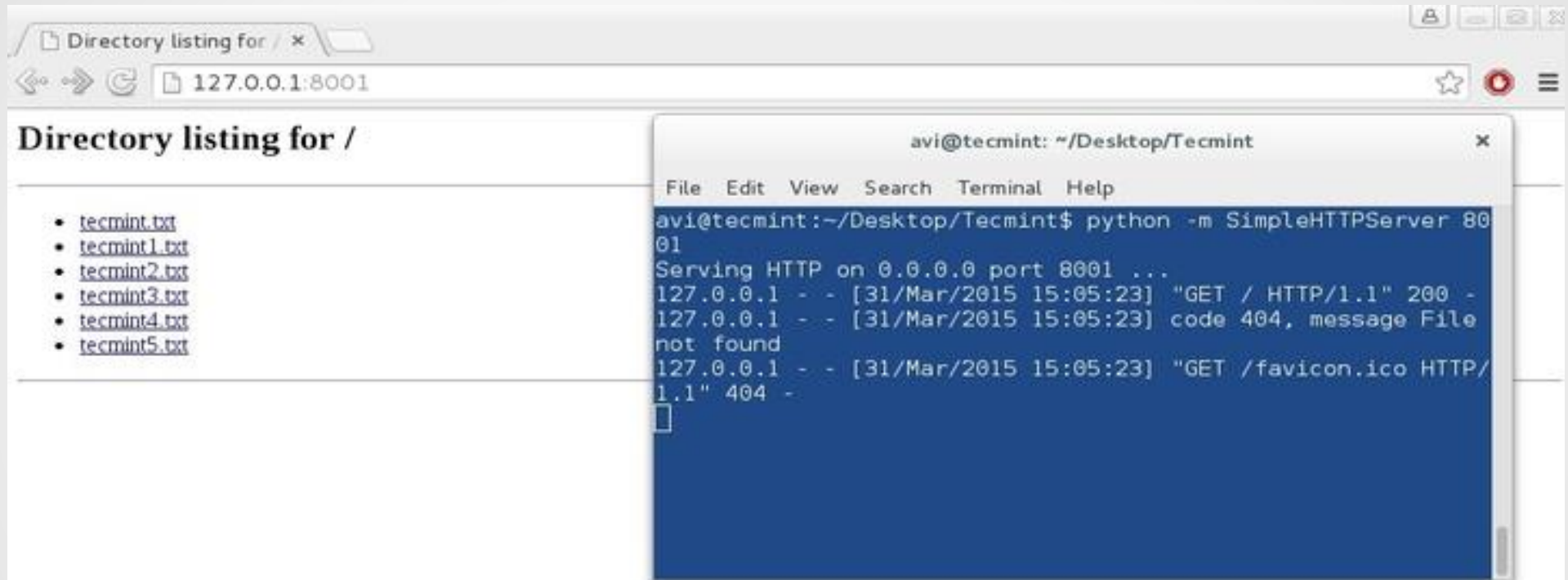
- A simple file server
- Standard Library
- Extremely simple
- Serves files from the directory it is run in

`python2 -m SimpleHTTPServer 8000`

or

`python3 -m http.server 8000`

SimpleHTTPServer



SimpleHTTPServer to Serve Files in Linux

Beyond simple file serving - MicroFrameworks

- Bottle.py - <https://bottlepy.org/docs/dev/>
- Web2py - <http://www.web2py.com/>
- Flask - <http://flask.pocoo.org/>

The 3 Main Components

- **View function** – that contains the logic
- **Templating** – embedding python variables/routines in html code
- **Routing** – Routing incoming http requests to the appropriate function, where every path can be represented by a function

Bottle.py

```
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)
```

Flask.py

```
from flask import Flask  
app = Flask(__name__)  
  
@app.route("/hello/<name>")  
def hello(name):  
    return "Hello %s!" %(name)
```

What flask brings

- An awesome templating engine -Jinja2
- Werkzeug – routing and basic session and cookie management

Database Management

- Flask – Flask-SQLAlchemy
- Django – Django ORM
 - Version controlled database schema
 - No need to remember all the changes you made
 - Change models.py
 - Run `makemigrations` and `migrate`

Django – Database Management

```
• class Author(models.Model):  
•     """Model representing an author."""  
•     first_name = models.CharField(max_length=100)  
•     last_name = models.CharField(max_length=100)  
•     date_of_birth = models.DateField(null=True, blank=True)  
•     date_of_death = models.DateField('Died', null=True, blank=True)  
  
class Book(models.Model):  
•     """Model representing a book (but not a specific copy of a book)."""  
•     title = models.CharField(max_length=200)  
•     author = models.ForeignKey('Author', on_delete=models.SET_NULL,  
•         null=True)  
•     summary = models.TextField(max_length=1000, help_text='Enter a brief  
•         description of the book')  
•     isbn = models.CharField('ISBN', max_length=13, help_text='13 Character  
•         <a href="https://www.isbn-international.org/content/what-isbn">ISBN  
•         number</a>')  
•     genre = models.ManyToManyField(Genre, help_text='Select a genre for  
•         this book')
```

Django Database Management

- `python manage.py makemigrations`
 - ... generates a python file
`app/migrations/0001_timestamp.py`
- `python manage.py migrate`
- ... makes the changes to the db
- `python manage.py migrate my_app`
`<name_of_last_needed_migrate_file>`
 - Reverts all changes made since
`<name_of_last_needed_migrate_file>`
- `python manage.py migrate my_app zero`
- - Revert all

Power of templates

- # base_generic.html
- <!DOCTYPE html>
- <html lang="en">
- <head>
- {% block title %}<title>Local Library</title>{% endblock %}
- </head>
- <body>
- {% block sidebar %}{% endblock %}
- # index.html
- {% extends "base_generic.html" %}
- {% block content %}
- <h1>Local Library Home</h1>
- <p>Welcome to LocalLibrary, a website developed by Mozilla Developer

User Management

- Flask – flask-login
- Django – builtin auth
 - Django all-auth

Form Management

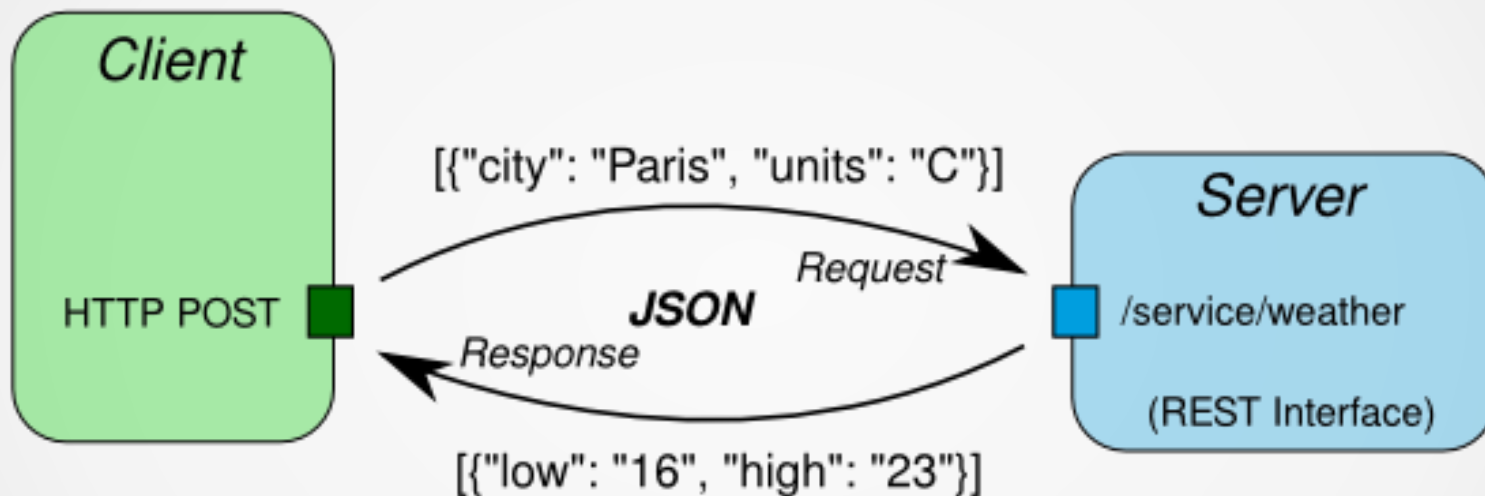
- Automatic HTML Form creation from Model Definition
- Form Validation
- Flask – WTForms
- Django – Built in

Security

- Password Encryption
- Cross site scripting (XSS) protection
- Cross site request forgery (CSRF) protection
- SQL injection protection
- Clickjacking protection¶
- SSL/HTTPS
- Host Validation

Rest Framework

JSON / REST / HTTP



Other Django Features

- Localization
- Integrations with AWS (for storage, emailing)
- Asynchronous Processing with Celery
- Integrations with MailChimp and other SaaS third party providers

Contributing to Open Source - Why

- Obviously, Looks good on your resume
- Experience
 - Tech
 - Coding Practices
 - Soft Skills
 - Project Size

Contributing to Open Source - How

- Bug triaging can help core contributors save a lot of time. For example, many new bugs filed are either duplicates or are fixed already.
- Community Building
 - Bringing in new contributors to replace older contributors when they move on
 - Increasing diversity in the group of contributors
- Improving documentation
- Supporting new users in different communities – Stackoverflow, reddit, irc, slack, mailing lists etc

Beginner Friendly Options

- Mozilla
 - Web Development
<https://wiki.mozilla.org/Webdev/GetInvolved>
 - DXR – Code Search and Navigation tool
<https://wiki.mozilla.org/DXR>
 - More ..
<https://whatcanidofromozilla.org/#!/progornoprogram/proglang/py/web-development>
- Django -
<https://code.djangoproject.com/query?status=!closed&easy=1>

Bottle Resources

- <http://pwp.stevecassidy.net/> - Python Web Programming
 - By Steve Cassidy, Department of Computing, Macquarie University, Australia

Thank you!

Django Resources

- <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>
- <https://overiq.com/django-1-10/intro-to-django/>
- <https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/>