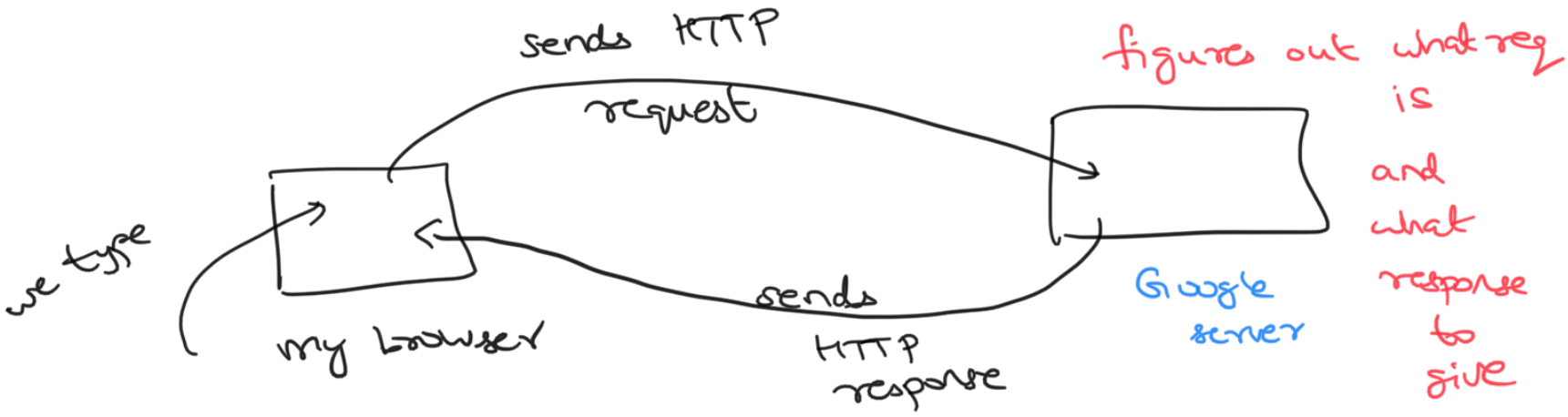
Lecture : Flask and HTML

HTTP : hyper text transfer protocol



- web application : code server side & things
  - taking req
  - figuring out req
  - figuring out response
  - sending response

Flask - micro framework written in Python

Flask application

example : application.py

flask module

```
from flask import Flask
```

Flask web server

```
app = Flask(__name__)
```

```
# create new web application
# type Flask and __name__
=> this file is rep my web application

@app.route("/")
def index():
    return "Hello, world"
```

decorator

default page of website

route = part of URL of which page you request

this index function tied to * line

URL = uniform resource locator

[so if user goes to "/", this index function is the function that is going to run]

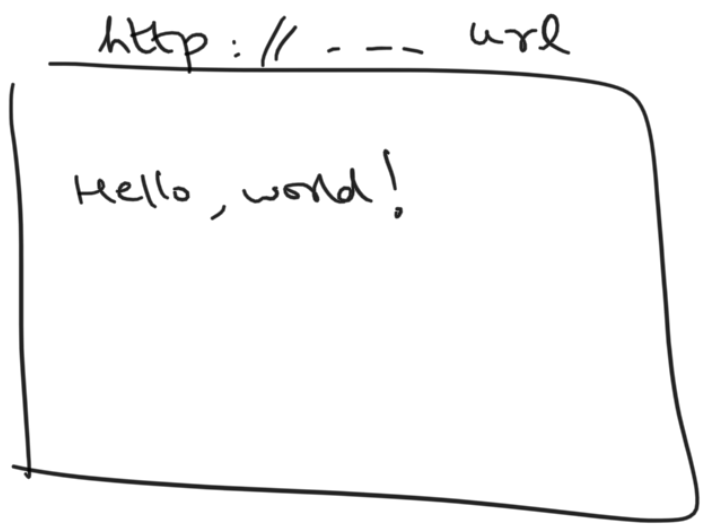
To run flask files:

- go to directory where flask files for your web application is stored
- commandline flask run

```
$ flask run
Serving Flask app "application"
Running on http://127.0.0.1:...
Restarting with stat
Debugger active
Debugger PIN: #---
```

your application is running on this URL

5 digit numeric code



If in debug mode

→ can edit python code in between

→ website will change!

→ no need to rerun "flask run" ...

Flask: relies on "environment variable" set inside terminal

to know what file

to be looking for as source of the application

In terminal, set environment variable

export FLASK_APP = application.py

↑ tells flask that the file I want to run application from is application.py

Can add more routes

```
@ app.route("/new")
def newf():
    return "Hello, new person"
```

Can generalize routes...

```
@ app.route("/<string:name>")
def hello(name):
    return f"Hello, {name}"
```

→ Can intersperse HTML tags in format strings

(eg) return f"Hello, {name}"

↔ return f"<h1> Hello, {name} </h1>"

Tying HTML files with Flask files more cleanly

```
from flask import Flask, render_template
app = Flask(__name__)
@ app.route("/")
def index():
    return render_template("index.html")
```

— app.py

— templates → index.html

Flask will search inside a "template" folders right in current proj directory for "index.html" if using render-template function.

more flask features

your html page can have place holders

(eg) index.html

written in templating language called Jinja 2

```
<!DOCTYPE html>
<html>
  <head>
    <title> my website
    </title>
  </head>
  <body>
    <h1> {{ head }} </h1>
  </body>
</html>
```

place holder

```
@ app.route ("/")
def index ():
```

headline = "Hello"

return render_template("index.html",
headl = headline)

Jinja 2 syntax for if, else statements... ...

new_year ← Boolean variable fed to render_template.

{% if new_year %}

<h1> yes! </h1>

{% else %}

<h1> no ! </h1>

{% endif %}

JS works well with flask

↑ client side

↑ server side

Jinja 2 - loops

@app.route("/")

def index()

names = ["A", "B", "C"]

```
return render_template("index.html",
                       names = names)
```

In index.html

```
<ul>
    {% for name in names %}
        <li> {{name}} </li>
    {% endfor %}
</ul>
```

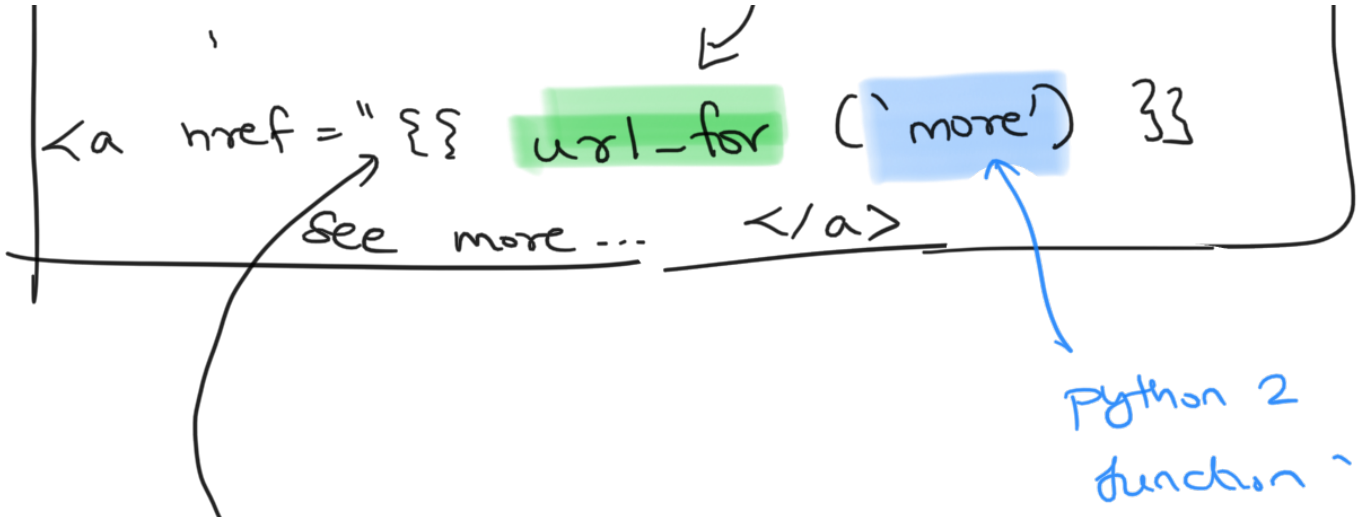
Linking to different parts of web application

```
@app.route("/")
def index():
    return render_template("index.html")

@app.route("/more")
def more():
    return render_template("more.html")
```

index.html

Special Jinja 2 / Flask function



This code tells that I want to link to the route to which 'more' python function is associated to

in this case, it is /more.

The url_for('more') can be used in Python-Flask code also!

Use a layout page to creak multiple similar pages which are slightly different

Template inheritance

→ Flask file remains same

```
@app.route("/")
def index():
    return render_template("index.html")
```

```
@app.route("/more")
def more():
    return render_template("more.html")
```

ask

→ In templates

- layout.html
- index.html
- more.html

index, more inherit from layout.html

layout.html

creates a variable section called x

```
<h1> {% block x %} {% endblock %} </h1>
```

```
{% block y %}
```

creates a variable section called y

```
{% endblock %}
```

Index.html

```
{% extends "layout.html" %}

{% block x %}

First page
```

```
{% endblock %}
```

```
{% block y %}
```

```
<p> Blah_ blah blah blah </p>

<a href = " {{ url_for('more') }} "

    .See more </a>
```

```
{% endblock %}
```