LECTURE 35

MAKING SIMPLE APIS WITH FLASK

MCS 260 Fall 2020

Emily Dumas

REMINDERS

- Complete Worksheet 12
- Quiz 12 available
- Project 4 proposals ASAP, due Nov 16

FLASK

Flask is a Python **web framework**. It lets you build APIs and web sites with Python.

Competitors include:

- Bottle — minimalist like Flask
- Django — huge and full-featured

INSTALLING FLASK

Using `pip` to install, if the interpreter name is `python`:

```
python -m pip install flask
```

Confirm installation by testing import in the REPL:

```
>>> import flask
>>>
```

If Flask is not installed, this will produce an error.

DECORATORS

Flask uses a lot of decorators, a Python feature we haven't covered yet.

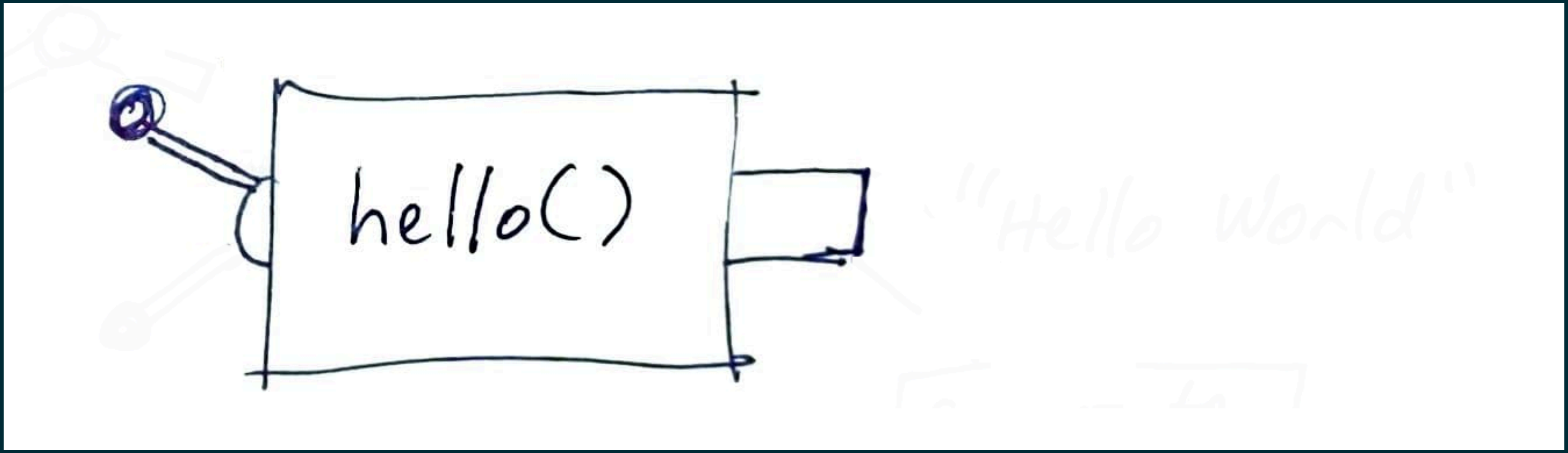
Basically,

```
@name
def f(x,y,z):
```

means: After defining `f`, it should be replaced with `name(f)`. Here `name` should be a higher-order function that modifies the operation of `f`.

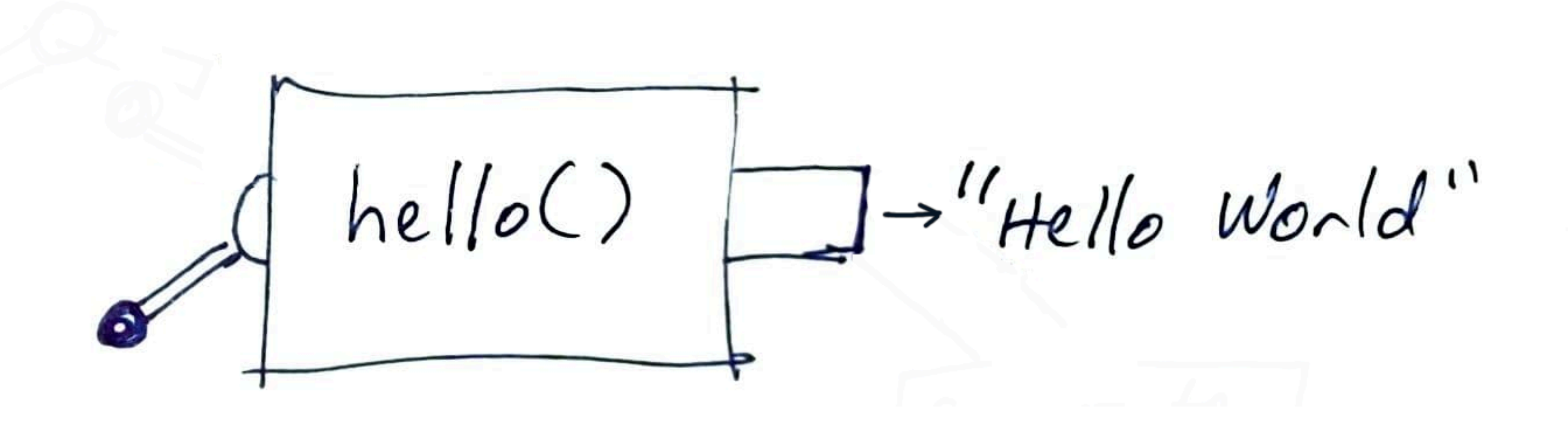
WHY DECORATORS IN FLASK?

We might write a function that returns a string:



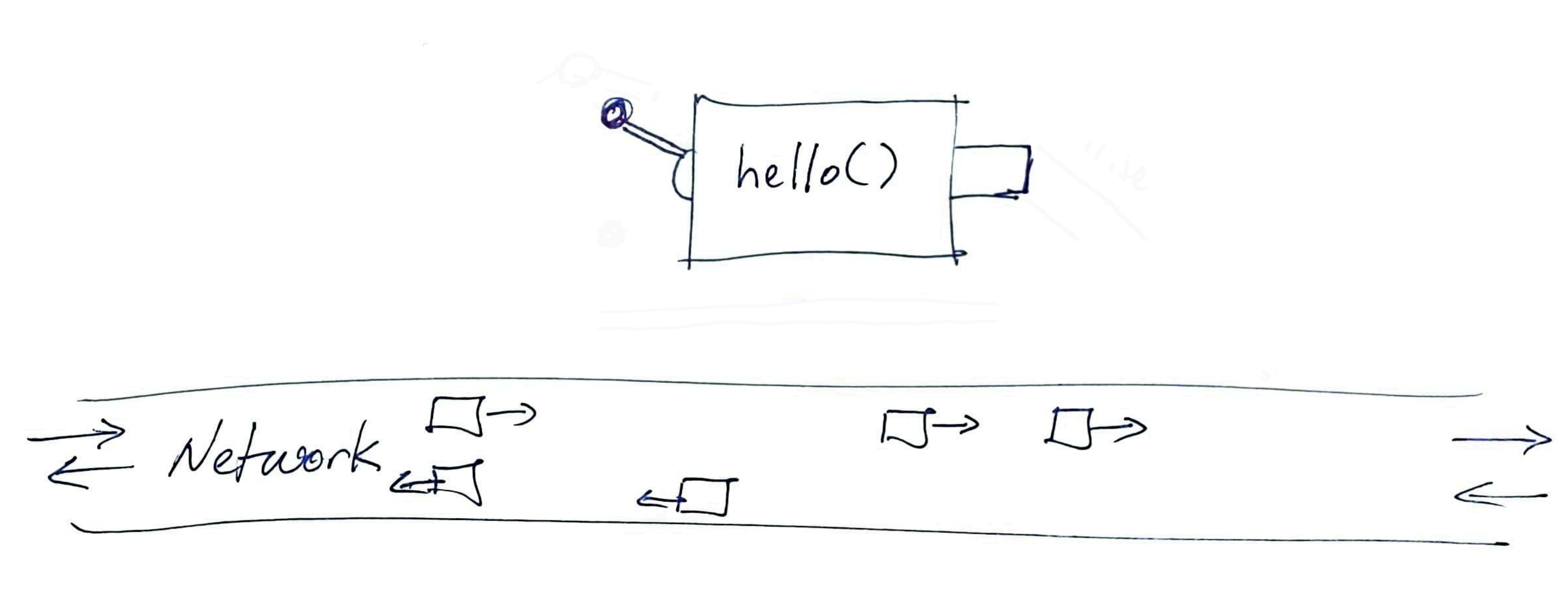
WHY DECORATORS IN FLASK?

We might write a function that returns a string:



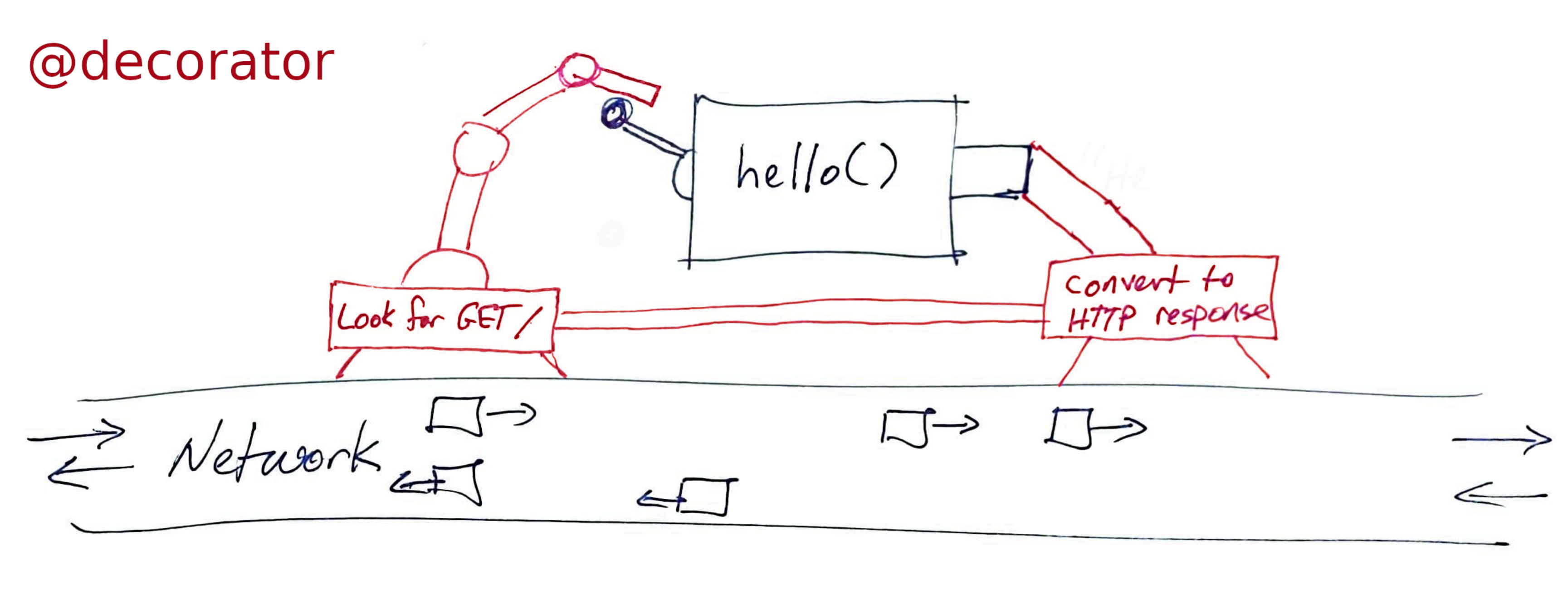
WHY DECORATORS IN FLASK?

A Flask decorator lets us link it directly to a URL:



WHY DECORATORS IN FLASK?

A Flask decorator lets us link it directly to a URL:



GOAL 1

Build API so that

```
/
```

returns

```
"Hello World".
```

GOAL 2

Build API so that

```
/apple/random
```

returns a JSON object with info about an apple variety selected at random from a list.

GOAL 2'

Build API so that

```
/metal/random
```

returns a JSON object with info about a randomly-selected metal.

GOAL 3

Build API that checks whether a given word is a palindrome, so

```
/palindrome/test?word=banana
```

would return `False` in JSON.

SUMMARY

- `localhost` refers to the host a request originates from.
- `127.0.0.1` is an address for localhost using the loopback network interface.
- `http://domain.com:1234/rest/of/URL/` means to use port 1234 instead of the default (which is 80 for HTTP).
- If `app` is a `flask.Flask` object, the decorator `@app.route("/path/part/of/URL/")` makes a function into a URL handler.
- Flask doesn't care about names of functions that handle URLs.
- `flask.jsonify` converts a Python value into a suitable return value from a URL handler.

REFERENCES

- The Flask tutorial is nice but is focused on building web pages.
- This Flask API development tutorial by Patrick Smyth is nice.

REVISION HISTORY

- 2020-11-13 Minor updates (goal 2', 127.0.0.1)
- 2020-11-12 Initial publication