

Overview



View logic

- Take argument from URL
- Return HTTP error
- Serving data as REST API

Template logic

- Jinja For and If statements
- Creating links to cards

Demo



Card detail page

- Parameter in URL to select card
- Return 404 when not found

Browse cards

- Create links to other cards
- If statement in card template

Card list

- For loop in template



Demo



Serving data as REST Api



```
@app.route( ' /example/<int:x>' )  
def my_view(x):  
    ...
```

Parameters in URLs

`<int:x>` will match a number in the url

This number will be assigned to the view parameter `x`

`/example/5` : `x` will be `5`

If URL contains no number: 404



```
@app.route(' /card/<x> ' )  
def my_view(x):  
    ...
```

Parameters in URLs 2

If you leave out the type (int), this will match any string



Returning an Error Code

```
from Flask import abort

@app.route('/card/<int:index>')
def card_view(index):
    try:
        ...
    except IndexError:
        abort(404)
```



```
<a href="{{ url_for('welcome') }}"> Homepage </a>
```

```
<a href="{{ url_for('card_view', index=3) }}"> Card 3 </a>
```

Reverse URL building

`url_for()` returns the URL for a view

First argument is name of view function

Other arguments will be passed to the view (use keyword args)



Jinja For Loops

```
<ul>
{% for card in cards %}
    <li>
        <a href="{{ url_for(...) }}">
            {{ card.question }}
        </a>
    </li>
{% endfor %}
</ul>
```



Jinja If Statements

```
{% if index < max_index %}
```

There is another card after this

```
{% else %}
```

This is the last card

```
{% endif %}
```



Summary



Card detail page

- Parameter in URL to select card
- Return 404 when not found
- Creating links with `url_for`
- If statement in card template

Card list

- For loop in template

Serving data as REST Api

