

Why Javascript?

► By: METIS



Why not just use forms?

- ▶ **Forms need to be submitted**

- Forms don't respond to adjusting controls.

- Need to click a button

- ▶ **Forms tie us to Flask**

- Web developers will use their own frameworks (React, Angular, etc)

- ▶ **Templates need to be rewritten for plugins/mobile/....**

- Every time we want to use our model, we need to write a template

We are not web developers!



Why should data scientists build web apps at all?

- ▶ **Allow stakeholders to see what models are doing**
Non-technical people have difficulty visualizing inputs and outputs
- ▶ **Check we aren't using forbidden inputs**
Make inputs explicit to check if we use inputs that won't be available.
Easier for subject matter experts to check this as well.
- ▶ **Check user experience**
A great model that requires 100+ manually entered inputs probably won't get used
- ▶ **Prototypes for software engineers**
A Flask app shows a software engineer how a "productionalized" web app works

BUT we don't want web devs to reimplement our models!





Application Programming Interfaces (APIs)

SEPARATING THE MODEL FROM THE PRESENTATION

Basic idea

I. Client (browser) sends a request to Flask app. This is like calling a function

```
To: "/convert_temperature"  
send {'temperature': 0, 'unit': 'C'}
```

2. Flask app responds with JSON (think dictionary)

```
{'K': 273.15, 'F': 32, 'C': 0}
```

3. Client (browser) decides what to do with information

Javascript is the language that runs in the browser.
It can take the response and write it to the page



Flask implementation

```
@app.route('/convert_temperature', methods=['POST'])
def convert_temps():
    inputs = request.json
    # simple example: ignore inputs!
    # always give same answer
    output = {'K': 273.15, 'F': 32, 'C': 0}
    return jsonify(output)
```

- Flask defines `request` and `jsonify`
- Input found in `request.json` (not function arguments)
- Example in class implements function properly (i.e. output depends on input)



Accessing info: Python

```
import requests

input = {'temperature': 0, 'unit': 'C'}
url = 'http://localhost:5000/convert_temperature'

# Make a request
r = requests.post(url, json=input)

# See the returned result
print(r.json())
```

Flask app must be running for this to work!





Javascript and jQuery

....AND A REMINDER ON HTML TAGS

HTML tags

```
<div class='quotation' id='quote'>  
    “Programs must be written for people to read, and only  
    incidentally for machines to execute.”  
    -- Harold Abelson, Structure and Interpretation of Computer Programs  
</div>
```

This is an example of [link](http://google.com)



HTML tags

```
<div class='quotation' id='quote'>
    “Programs must be written for people to read, and only
    incidentally for machines to execute.”
    -- Harold Abelson, Structure and Interpretation of Computer Programs
</div>
```

This is an example of [link](http://google.com)

tag: div



HTML tags

```
<div class='quotation' id='quote'>  
    “Programs must be written for people to read, and only  
    incidentally for machines to execute.”  
    -- Harold Abelson, Structure and Interpretation of Computer Programs  
</div>
```

This is an example of [link](http://google.com)

tag: div has two attributes (class and id)



HTML tags

```
<div class='quotation' id='quote'>  
    “Programs must be written for people to read, and only  
    incidentally for machines to execute.”  
    -- Harold Abelson, Structure and Interpretation of Computer Programs  
</div>
```

This is an example of [link](http://google.com)

tag: div has two attributes (class and id).

It has (inner) html between opening tag `<div>` and closing tag `</div>`



HTML tags

Javascript changes quote by selecting element, then using "html" method.

```
<div class='quotation' id='quote'>  
  “Programs must be written for people to read, and only  
  incidentally for machines to execute.”  
  -- Harold Abelson, Structure and Interpretation of Computer Programs  
</div>
```

```
// select by div (warning: changes html of ALL divs on page)  
$('div').html('Java is to javascript as car is to carpet');  
  
// select by class (warning: ALL elements with class quotation changed)  
$('.quotation').html('Java is to javascript as car is to carpet');  
  
// select by id  
$('#quote').html('Java is to javascript as car is to carpet');
```



HTML tag: quiz

```
<div class='quotation' id='quote'>  
    “Programs must be written for people to read, and only  
    incidentally for machines to execute.”  
    -- Harold Abelson, Structure and Interpretation of Computer Programs  
</div>
```

This is an example of `link`

- What is the tag for the highlighted element?
- Does the element have a class? If so, what is it?
- Does it have any attributes? If so, what?
- What is the (inner) html of the tag?



Let's see it on a page!

Go to thisismetis.com and open the console →



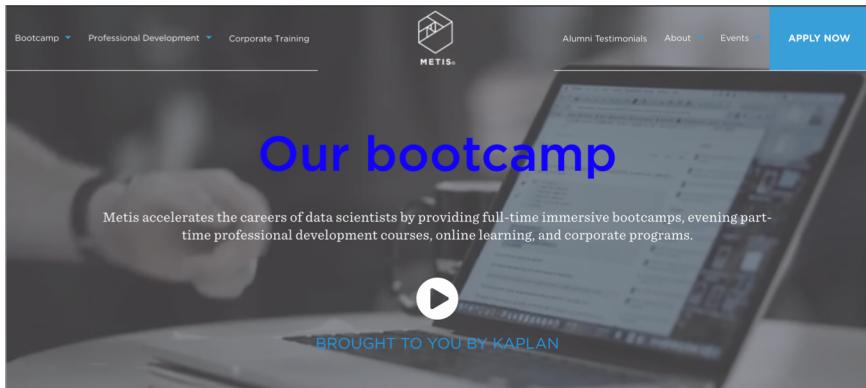
Example



h1 element has class "page-title"

In console, type

```
$( 'h1.page-title' ).html('Our bootcamp');  
$( 'h1.page-title' ).css('color', 'blue');
```



Result!





AJAX

KEEP THIS AS REFERENCE

Call '/convert_temperature' from JS

```
// same inputs as before
const features = {'temperature': 0, 'unit': 'C'};

// call Flask
$.post({
  url: 'http://localhost:5000/convert_temperature',
  contentType: 'application/json',
  data: JSON.stringify(features),
  success: ..... // function to call if successful
  error: ..... // function to call if there's an error
});
```

- Always need to serialize input to string. Built in `JSON.stringify` does this for us
- Usually the "success" function will contain code to take the response, select elements, and overwrite the HTML to reflect how the server responded





THANK YOU!
