

WebDev: Flask + JQuery

John McDonnell

August 7, 2014

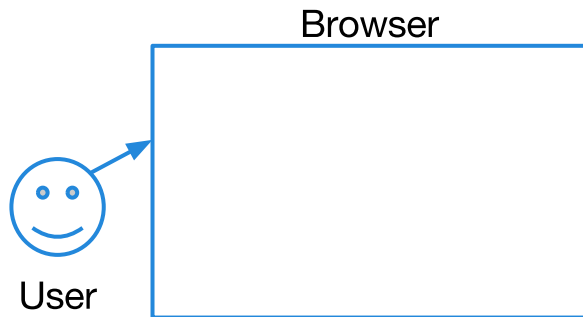
Coding the web with JavaScript

- Contextualize the environment.
 - Browsers are a crazy place with lots of crazy stuff going on. Let's get on the same page.
- Build out some dynamic content.
 - A button that does things.
 - Pulling down data on the fly with JQuery.

The web environment

- Web Browser
- DNS
- Page load
 - HTML first, sets up the DOM & renders the page
 - Loads other resources: Images, stylesheets, etc.
- Dynamic Javascript code runs
 - Manipulates the page (jQuery selectors)
 - Might pull down its own resources: (AJAX in jQuery)

Where the magic happens

**Action**

Download and use a browser. *Hopefully you've got this one taken care of!*

The web environment

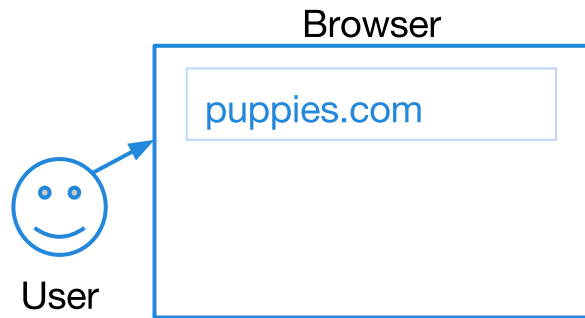
- ~~Web Browser~~
- DNS
- Page load
 - HTML first, sets up the DOM & renders the page.
 - Loads other resources: Images, stylesheets, etc.
- Dynamic Javascript code runs
 - Manipulates the page (jQuery selectors)
 - Might pull down its own resources: (AJAX in jQuery)

Internet White Pages: DNS

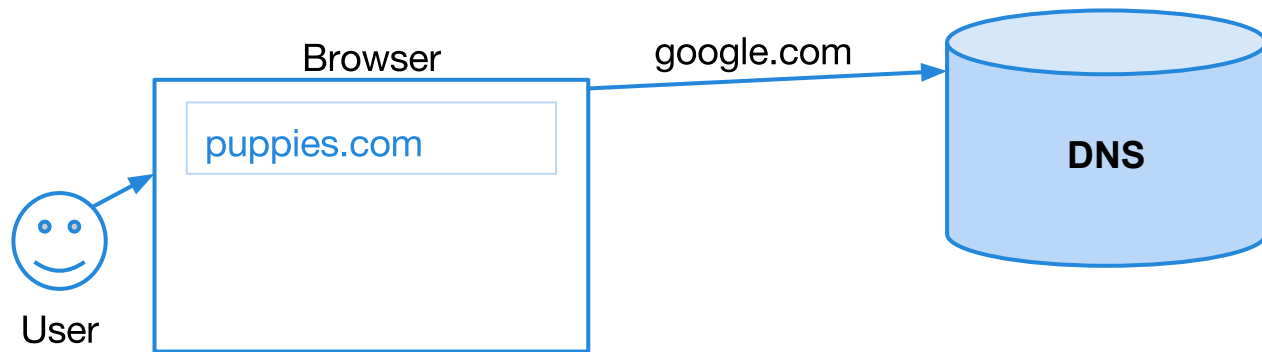
DNS

Domain Name Service

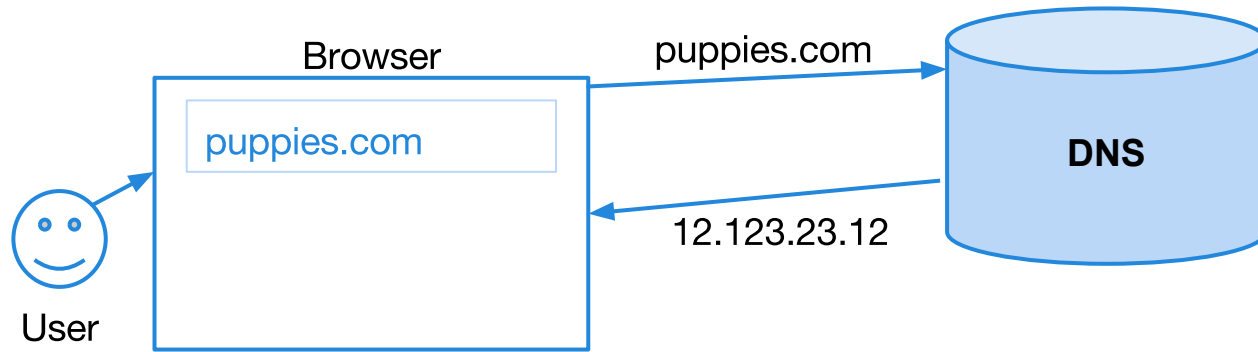
Internet White Pages: DNS



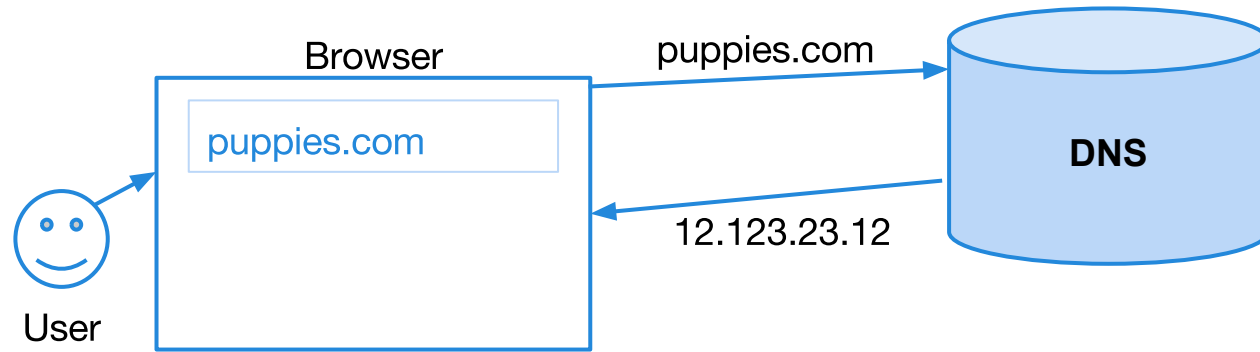
Internet White Pages: DNS



Internet White Pages: DNS



Internet White Pages: DNS



Action

Use [gandi.net](https://www.gandi.net) or [nearlyfreespeech.net](https://www.nearlyfreespeech.net) to sign up your URL (puppies.data?) in a DNS registry (next week).

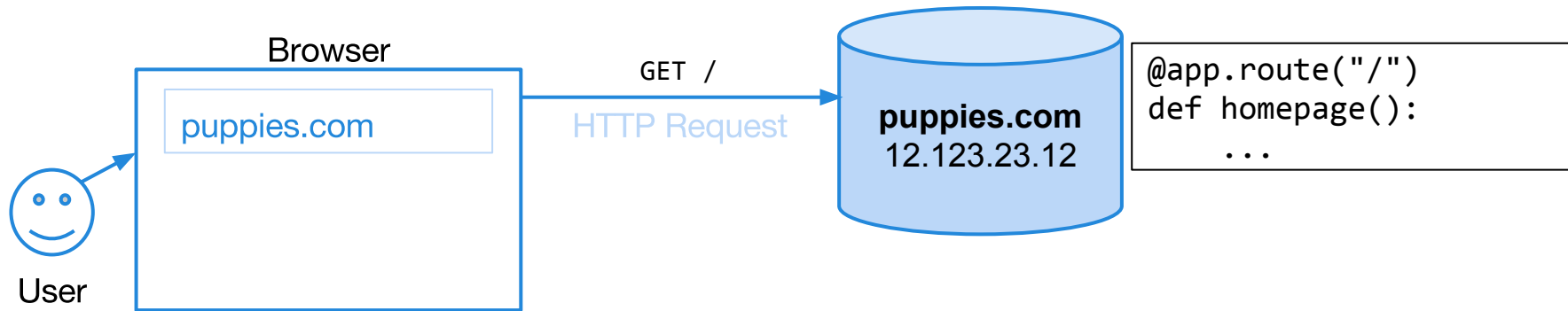
The web environment

- ~~Web Browser~~
- ~~DNS~~
- Page load
 - HTML first, sets up the DOM & renders the page.
 - Loads other resources: Images, stylesheets, etc.
- Dynamic Javascript code runs
 - Manipulates the page (jQuery selectors)
 - Might pull down its own resources: (AJAX in jQuery)

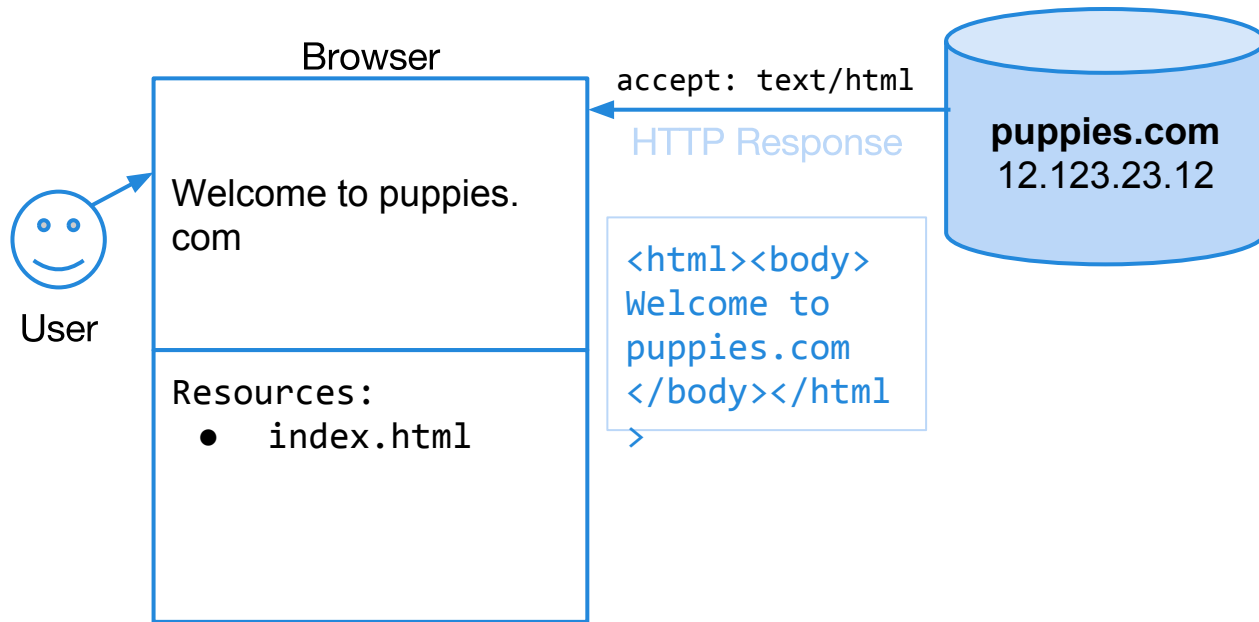
Page load



Page load



Page load



Page load

index.html

```
<html><head>  
  <link rel="stylesheet" ref="puppies.css" />  
  <script src="puppies.js"></script>  
</head><body>  
    
  Welcome to puppies.com  
</body></html>
```

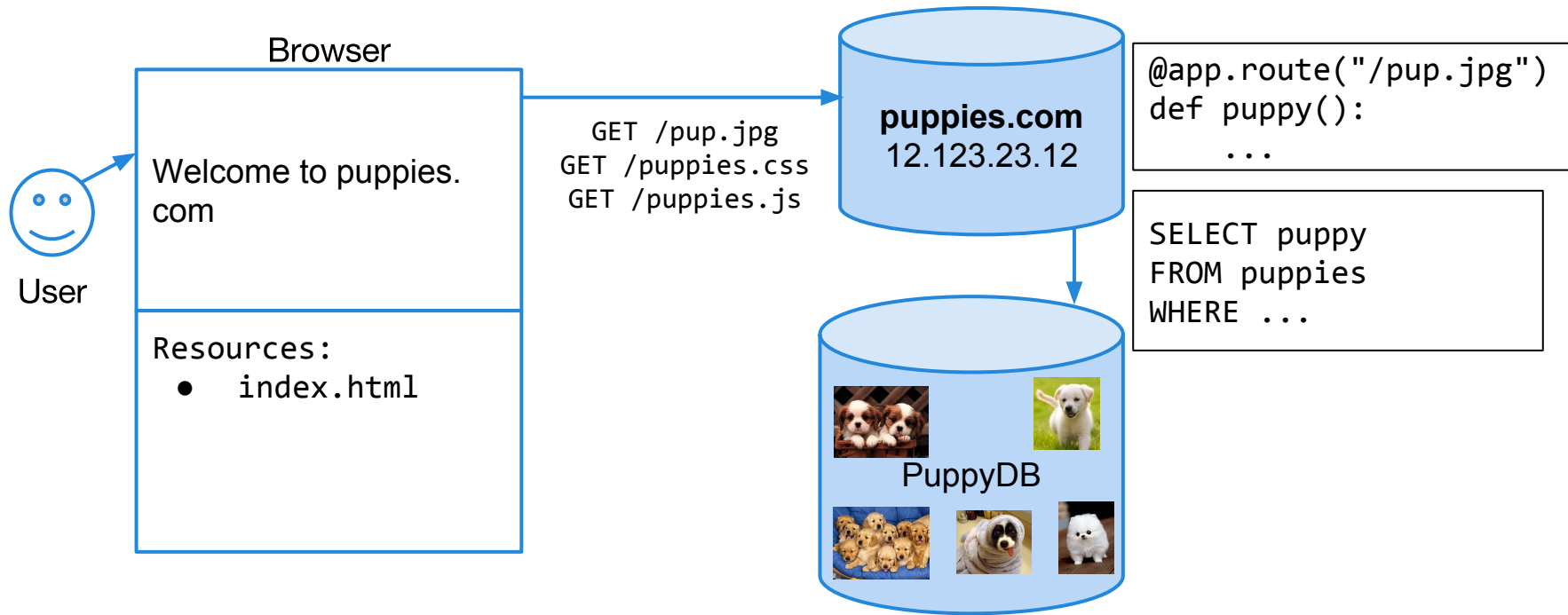
Page load

index.html

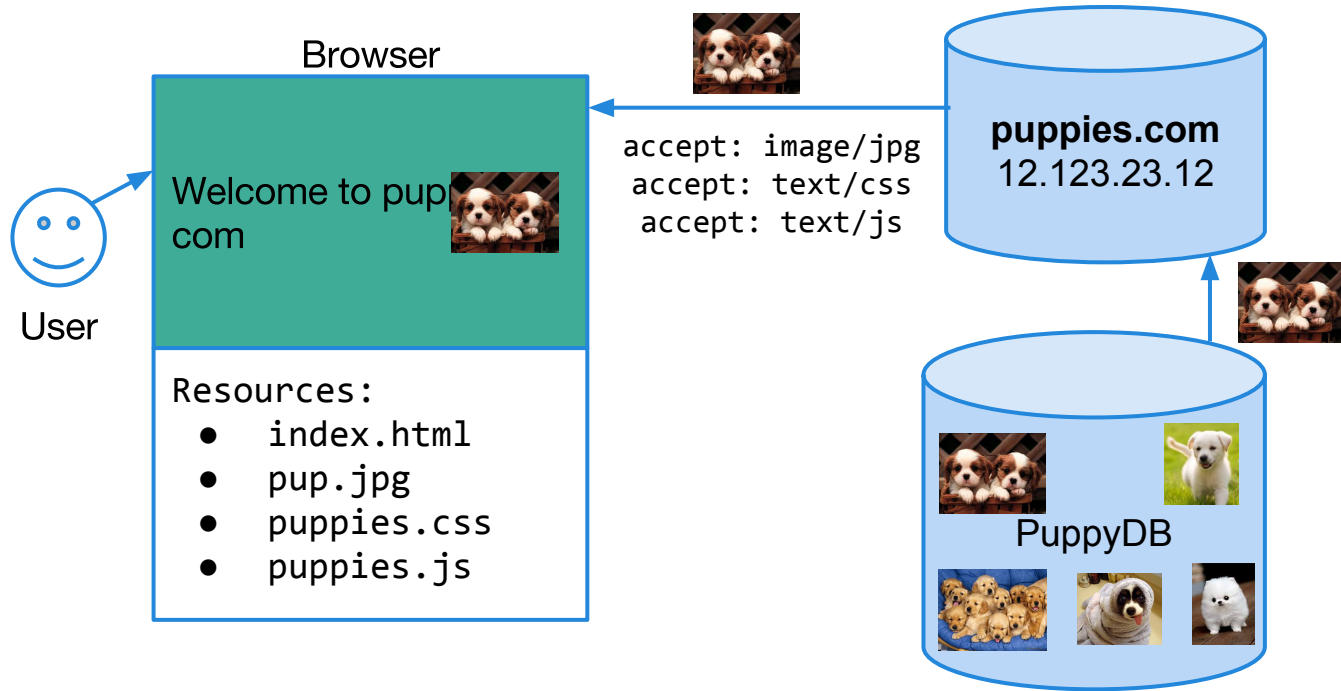
```
<html><head>
  <link rel="stylesheet" ref="puppies.css" />
  <script src="puppies.js"></script>
</head><body>

Welcome to puppies.com
</body></html>
```

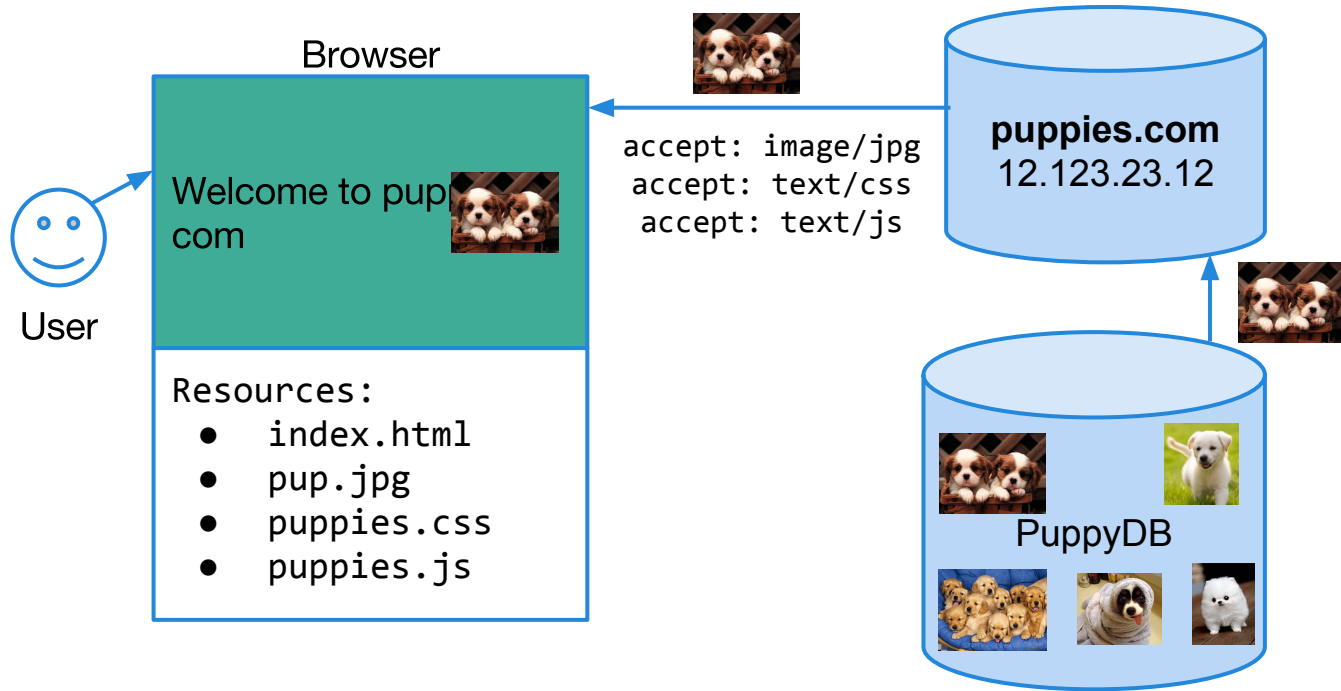

Page load



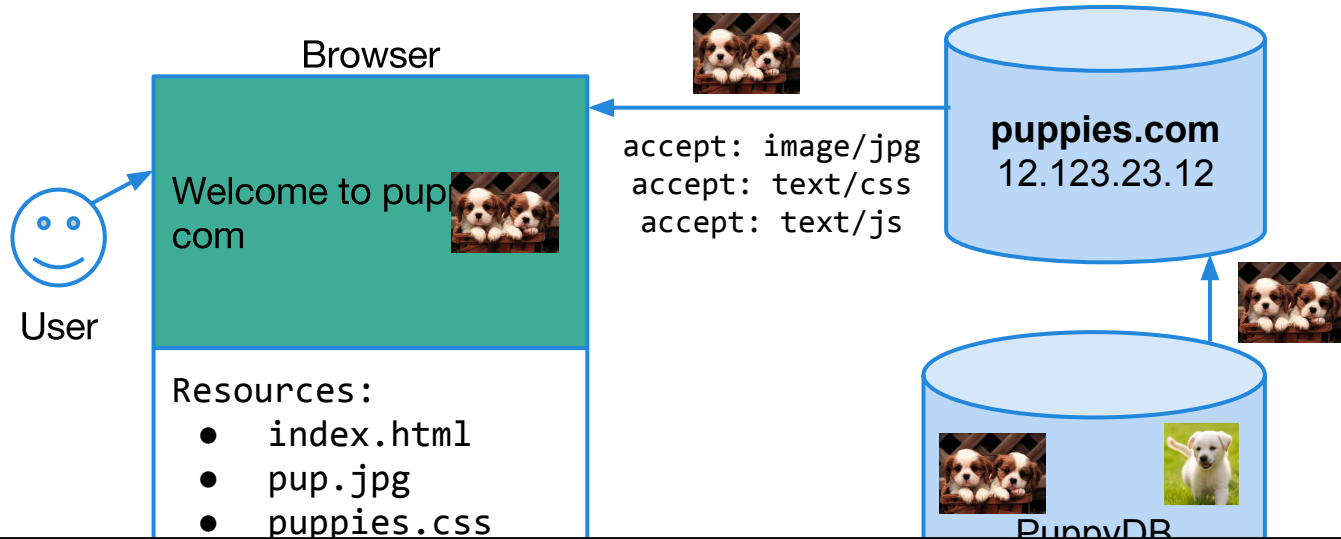
Page load



Page load



Page load



Action

Set up a basic website using Flask.

The DOM


DOM

Document Object Model

HTML: Initializing the DOM

index.html

```
<body>
  <div id="main">
    <h1>Welcome to puppies.com</h1>
    
    <p>
      This is one of our favorite puppies.
    </p></div></body>
```




The diagram illustrates the DOM tree structure for the provided HTML code. A red arrow points from the text "Nested in body" to the opening `<body>` tag. Another red arrow points from the text "Nested in div #main" to the opening `<div id="main">` tag, demonstrating how elements are nested in the DOM.

Tags in body become DOM objects nested in one another when the browser parses the HTML.

The DOM

Featured Breed




Cavalier King Charles Spaniel


Country of Origin: The Cavalier King Charles Spaniel (affectionately known as a 'Cav') is an offshoot of the King Charles Spaniel. The King Charles Spaniel was created by crossing small Spaniels with a short snouted breed such as the Pug or Japanese Chin. It was brought to Scotland from Continental Europe (possibly by Mary, Queen of Scots) in the 1500's or 1600's, where it became a fashionable lap

Puppies *for Sale*

Featured Listings



Harry
Sex: Male
Age: 8 Weeks
Price: \$1050 USD



Boy2
Sex: Male
Age: 5 Weeks
Price: \$550 USD

[More Puppies for Sale](#)

Sources Timeline Profiles Sources Audits Console

```
<tr>
  > <td>...</td>
  > <td>
    > <a href="/breed_gallery/?breed_id=36" title="Cavalier King Charles Spaniel Photos">
      
```

CSS: Styling the DOM


puppies.css


```
body {  
  font-family: 'MS Comic Sans'  
}
```

```
#main {  
  background-color: #271950  
}
```


CSS: Styling the DOM

puppies.css

```
body {  Affects everything nested in body  
  font-family: 'MS Comic Sans'  
}
```

```
#main {  Affects everything nested in div 'id=main'  
  background-color: #271950  
}
```

CSS: Referencing notation

Reference type	Example	Example notes
Reference by tag	<code>img</code>	Pulls out the 'img' tag DOM objects.
Reference by id	<code>#main</code>	Pulls out the DOM object with id 'main'
Reference by class	<code>.red</code>	Pulls out all DOM objects with class 'red'

Id: Appears once in a page

Class: Appears multiple times

CSS: Referencing notation

Reference type	Example	Example notes
Reference by tag	<code>img</code>	Pulls out the 'img' tag DOM objects.
Reference by id	<code>#main</code>	Pulls out the DOM object with id 'main'
Reference by class	<code>.red</code>	Pulls out all DOM objects with class 'red'

Foreshadowing: JQuery uses similar notation to grab items on the page.

CSS: Referencing notation

Reference type	Example	Example notes
Reference by tag	<code>img</code>	Pulls out the 'img' tag DOM objects.
Reference by id	<code>#main</code>	Pulls out the DOM object with id 'main'
Reference by class	<code>.red</code>	Pulls out all DOM objects with class 'red'

Action

Add some CSS styling to your website. If this is completely new, play around a bit.

The web environment

- ~~Web Browser~~
- ~~DNS~~
- ~~Page load~~
 - HTML first, sets up the DOM
 - Then other resources: Images, stylesheets, etc.
- Dynamic Javascript code runs
 - Manipulates the page (jQuery selectors)
 - Might pull down its own resources: (AJAX in jQuery)

Javascript: Dynamic content

JavaScript

A scripting language that runs in your browser.

Javascript: Dynamic content

- Javascript can...
 - Alter existing DOM elements, changing the page.
 - Add new DOM elements
 - Pull down and manipulate data (AJAX)
- Javascript acts without a page reload, so it can make your page seem very zippy.

Javascript: Packages

- JavaScript (out of the box).
 - Powerful but hard to use.
- JQuery.
 - Makes JS much easier, but no “magic.”
- Bootstrap
 - Javascript without Javascript, widget interaction.
- d3.js
 - Web visualization made powerful and challenging.

What is this Bootstrap thing?

- **Bootstrap** is a web development framework.
- It provides default CSS and JS.
 - Sensible and attractive styling
 - Buttons and widgets
 - Pop-ups
- Provides sane defaults, but can be difficult to override.
 - If this is an issue, try purecss.io for a more hands-off framework.

What is this Bootstrap thing?

Action

Boot up Bootstrap and play with some of the widgets highlighted in the Javascript tab of the Bootstrap website.

Break/Questions

???

Javascript: Playing with the DOM

Playing with the DOM

We can manually manipulate the DOM in our browser.

[live demo]

JQuery: Manipulating the DOM

Command	Info
<code>\$</code>	This just references JQuery itself.
<code>\$('#main')</code>	Grab onto the object with id="main"
<code>\$('#main').css({color: 'red'})</code>	Change its CSS so its color is red.
<code>\$('#main').hide()</code>	Stop rendering. Use show to bring it back.
<code>\$('#main').remove()</code>	Delete it entirely from the document.

JQuery basics

Try opening the basic flask app and running these commands in the console:

- `$('h1').css({'color': 'red'})`
- `$('p').css({color: '#881188', "font-weight": 'bold', "font-size": "1.1em"})`

Javascript: Scenario 1

Scenario 1

Make a button that does something when you click it.

JQuery: More tools

Command	Info
<code>\$('body').append('')</code>	Add something new into the DOM.
<code>\$('button').click(function() {...})</code>	Execute the given function when all buttons are clicked.

JQuery: More tools

Try opening the basic flask app and running these commands in the console:

- `$("#p").append('')`
- ```
puppy_maker = function() {
 $("#p").append('')
}
```
- `$('#h1').click(puppy_maker);`

# Javascript: Scenario 2

## **Scenario 2**

Pull in some data from a DB  
and do something with it.

# JSON: Machine-friendly data

`@app.route("/db_json")`

```
@app.route("/db_json")
def cities_json():
 with db:
 cur = db.cursor()
 cur.execute("SELECT Name, CountryCode, Population FROM city ORDER BY
Population DESC;")
 query_results = cur.fetchall()
 cities = []
 for result in query_results:
 cities.append(dict(name=result[0], country=result[1], population=result
[2]))
 return jsonify(dict(cities=cities))
```


# JSON: Machine-friendly data

[http://localhost:8000/db\\_json](http://localhost:8000/db_json)

```
{
 "cities": [
 {
 "City": "Kabul",
 "CountryCode": "AFG",
 "Population": 1780000
 },
 {
 "City": "Qandahar",
 ...
 }
]
}
```


# JQuery: AJAX

## Accessing in Javascript

```
mycontainer = [];
jQuery.ajax( AJAX function makes the call
 'db_json',
 {success:
 function(data) {
 mycontainer = data
 }
 }
)
```


# JQuery: AJAX

## Accessing in Javascript

```
mycontainer = [];
jQuery.ajax(
 'db_json',  Requests data from /db_json
 {success:
 function(data) {
 mycontainer = data
 }
 }
)
```

# JQuery: AJAX

## Accessing in Javascript

```
mycontainer = [];
jQuery.ajax(
 'db_json',
 {success:  Calls success function once it's come back.
 function(data) {
 mycontainer = data
 }
 }
)
```

# JQuery: More tools

Try opening the basic flask app and running these commands in the console:

- `mycontainer = [];`
- `jQuery.get('db_json', function(data) {mycontainer = data})`



# JQuery: AJAX

## Dynamically adding content

```
for (i=0; i < mycontainer['cities'].length; i++) {
 if (mycontainer['cities'][i]['Population'] > 1000000) {
 $("body").append('<p>' + mycontainer['cities'][i]
['City'] + ": " + mycontainer['cities'][i]['Population'] +
'</p>')
 }
}
```

# Javascript: Scenario 3

## **Scenario 3**

Incorporate JS into your page.

# Javascript: Scenario 3

```
<script>
$(document).ready(function() {
 /* do your thing here */
}
</script>
```

# JQuery: More tools

Live demo: Take a look at the  
index\_js.html file.

# Javascript

## **Action**

Add a button to your page that changes something.  
Dynamically load data into your page.

# More resources

**Jquery reference** [learn.jquery.com](https://learn.jquery.com)

**JS tutorial** [codecademy.com](https://www.codecademy.com)

**Convenience  
functions** [underscorejs.org](https://underscorejs.org)

**Templating** [handlebarsjs.com](https://handlebarsjs.com)

End/Questions

Thank you!