



FORMS

FORMS

 Secure payment

First name

Last name

Card number 

Exp MM  Exp YYYY  CVV 

Billing address

Address 1

Address 2 (optional)

Country United States 

City or town

State -- State --  Zip code

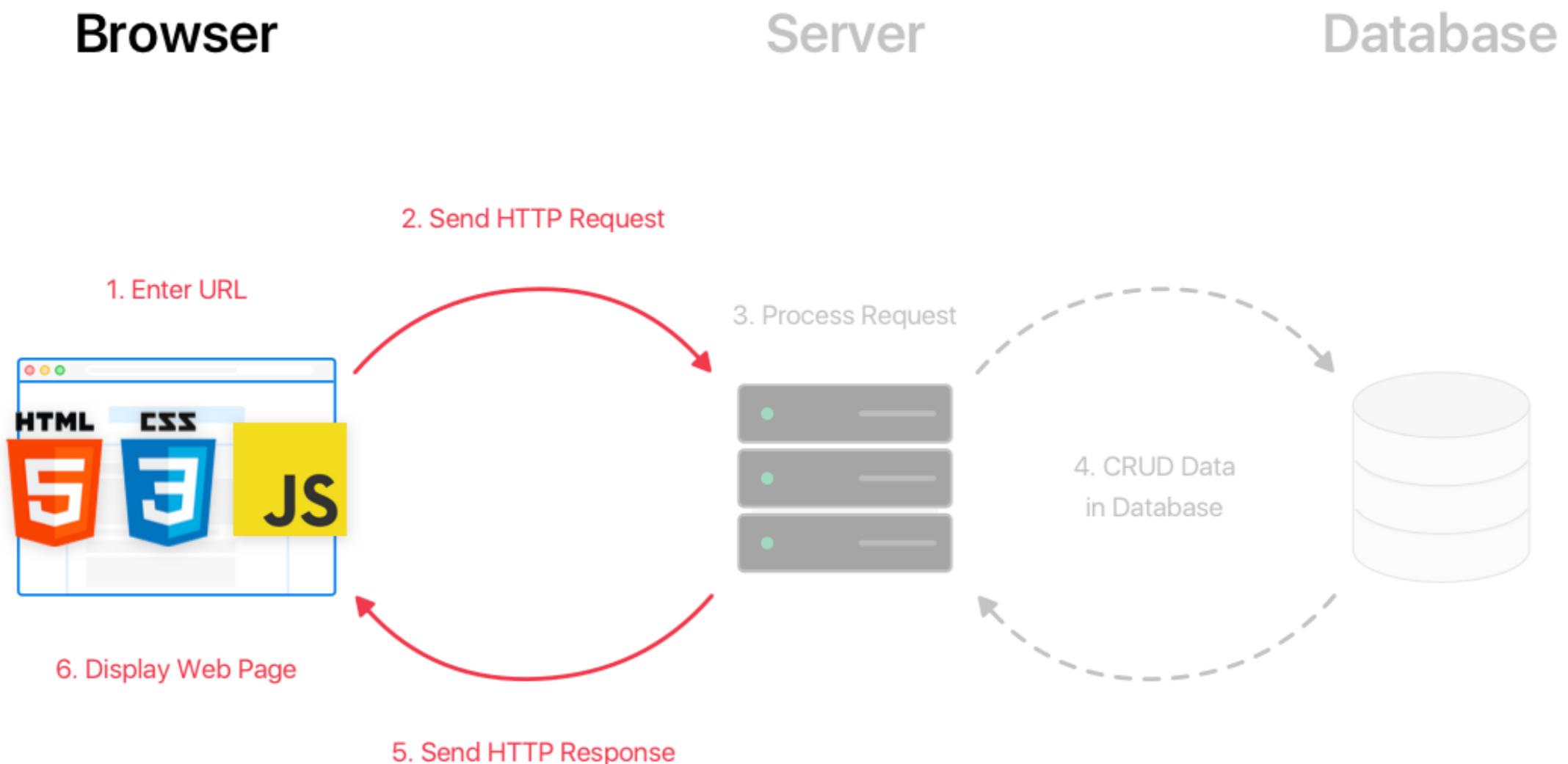
Save billing info?

 Yes  No

WHAT FORMS DO

- Forms capture user input from the web and send it off to the backend of a website to be processed in some way
- Any time a user puts any content into a webpage, they are using a form

REQUEST RESPONSE CYCLE



FORM TAG

```
<form action="send.php" method="POST">  
</form>
```

The action attribute tells the browser where to send the form information to. This is usually a backend programming language - like PHP, Ruby or Python.

GET VS POST

- 1- GET transactions take the form of query parameters and get appended to URLs (insecure).
- 2- POST transactions get interpreted by their destination and sent back to the host (secure).

<INPUT>

```
<form>
  <input type="text" />
</form>
```

Input is the workhorse of the form world - it can do a great many things.

<INPUT> TYPES

button

checkbox

color

date

email

file

number

password

radio

range

submit

tel

time

url

<LABEL>

```
<form>
  <label for="field1">New Field</label>
  <input type="text" id="field1">
  <button type="submit">
</form>
```

Label is the tag for telling a user what label of a field is. Note the 'for' attribute - what is that doing?

<FIELDSET>

```
<form>
  <fieldset>
    <input type="text">
    <button type="submit">
  </fieldset>
</form>
```

Fieldset wraps around a set of field inputs on the form and groups them together. Optional but useful in certain cases

<TEXTAREA>

```
<textarea placeholder="Write something  
here"></textarea>
```

Creates a really big text area. Don't size it with HTML attributes, use CSS please! Don't be that guy.

<SELECT>

```
<select id="select">
  <option value>Choose</option>
  <optgroup label="Option Group 1">
    <option value>Option 1</option>
    <option value>Option 2</option>
  </optgroup>
</select>
```

Selects are for those lists you click on to select an item.

CHECKBOXES

```
<div class="checkbox">  
  <input type="checkbox" id="checkbox2">  
  <label for="checkbox2">5</label>  
</div>
```

You have to put the labels after the checkbox for them to make any sense. I usually group them like this.

RADIO BUTTONS

```
<div class="radio">
  <input type="radio" id="radio1" name="radio-group">
  <label for="radio1"><span>Good</span></label>
</div>
<div class="radio">
  <input type="radio" id="radio2" name="radio-group">
  <label for="radio2"><span>Bad</span></label>
</div>
```

Radio buttons are very similar to checkboxes but function as a group with the 'name' attribute.

STYLING FORMS



NEW SELECTOR

```
input[type=text] {  
  margin: 25px 0;  
}
```

```
input[type="text"] {  
  margin: 25px 0;  
}
```

If you want to target a particular attribute in HTML, use the brackets. Both syntaxes are the same, no spaces.

:FOCUS

```
textarea:focus {  
    outline: none;  
}
```

There may be cases where you want to style what it looks like when a user is active on a field - use the :focus pseudo-selector. Outline property is helpful here.

STYLING NOTE

In general - don't style the text boxes, checkboxes and radio buttons themselves. You can do this but the techniques will make you want to jump out of window.

If you want to try this, let's talk outside of class - it's quite a bit of work and takes a lot of cross browser testing.

YOUR TURN

Assignment 1

User Info

First Name

Last Name

Email

Password

Birthday mm / dd / yyyy

Gender Male

Create a form that has inputs for:

First name
Last name
Email

Password
Birthday
Gender (select!)

BONUS: Style your form!

INPUTS + JQUERY

Sometimes we will want to retrieve values from inputs within our Javascript.
We can do this using JQuery!

```
var firstName = $('#first-name').val()
```

The val() method will get what has been entered into the input

JQUERY ON CHANGE

We can also add an event listener to check whenever the value of an input changes

```
$( '#password' ).on( 'change', validatePass)
```

This is sometimes used when checking if the input is valid (forms that as you type say if your password is strong enough)

JQUERY ON SUBMIT

Sometimes we will want to handle our form submission from our javascript

```
$('form').on('submit', sendData)
```

We can use the “submit” event listener to handle these events

JQUERY ON SUBMIT

Often times if we are handling the submission of a form using javascript, we won't want the form to automatically try to submit using the form "action" and "method"

JQUERY ON SUBMIT

We can prevent this behavior within the function handling the event

```
$('form').on('submit', sendData)

function sendData(event) {
  event.preventDefault();
}
```

Notice anything different about the function?

CODEALONG

Assignment 2

YOUR TURN

Assignment 3