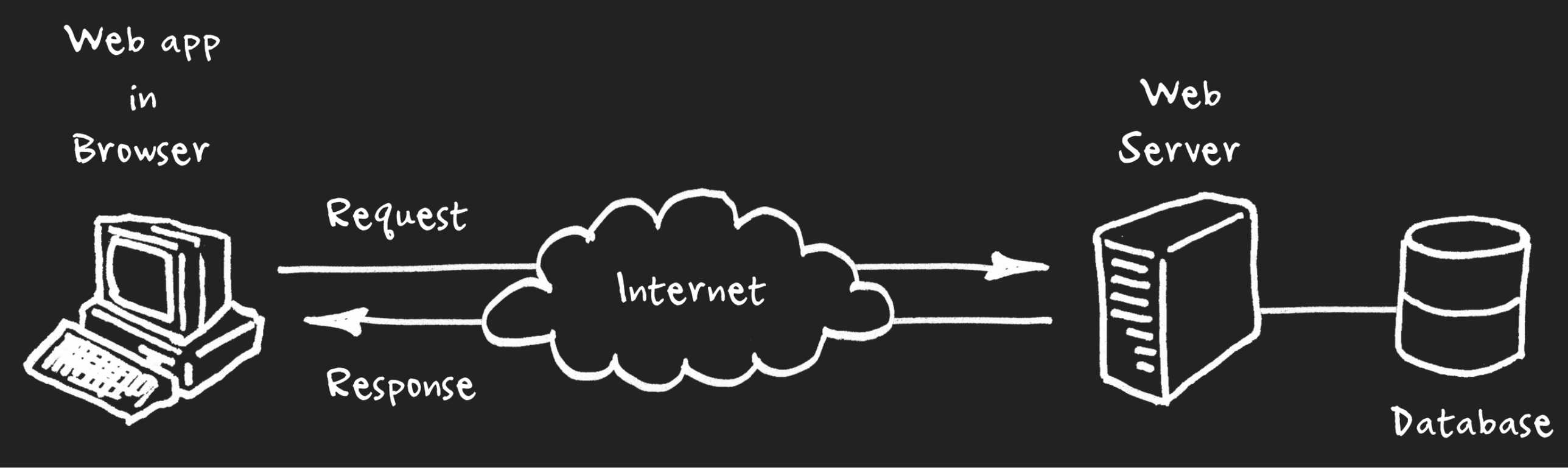
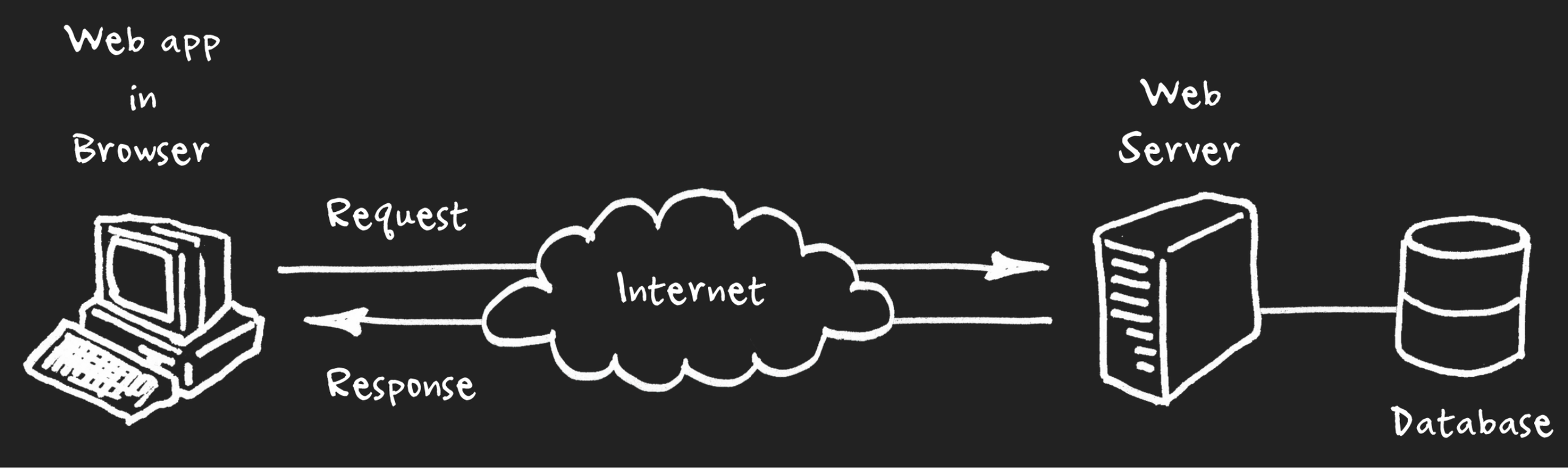
web applications



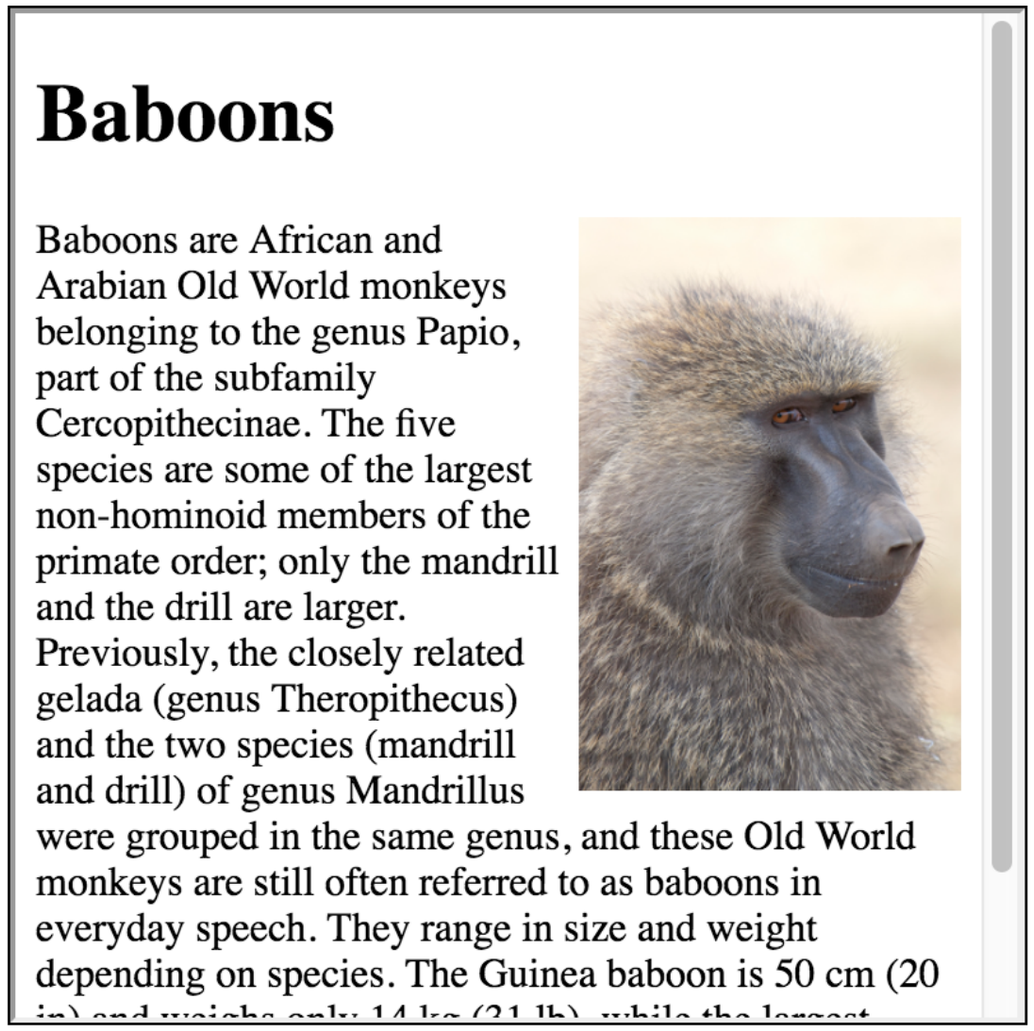
1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
    A. responsiveness (modify HTML with js, DOM)
    B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)



1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)

review: displaying info

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Fundamentals of Web Programming</title>
5      <style type="text/css">
6        img {
7          float: right;
8        }
9        </style>
10   </head>
11   <body>
12
13   <h1>Baboons</h1>
14
15   <img src="baboon.jpg" width="150">
16
17   <p>  Baboons are African and Arabian Old World monkeys
       belonging to the genus Papio, part of the subfamily
       Cercopithecinae. The five species are some of the
       largest non-hominoid members of the primate order; only
       the mandrill and the drill are larger. Previously, the
       closely related gelada (genus Theropithecus) and the
       two species (mandrill and drill) of genus Mandrillus
       were grouped in the same genus, and these Old World
```



HTML: content

css: style

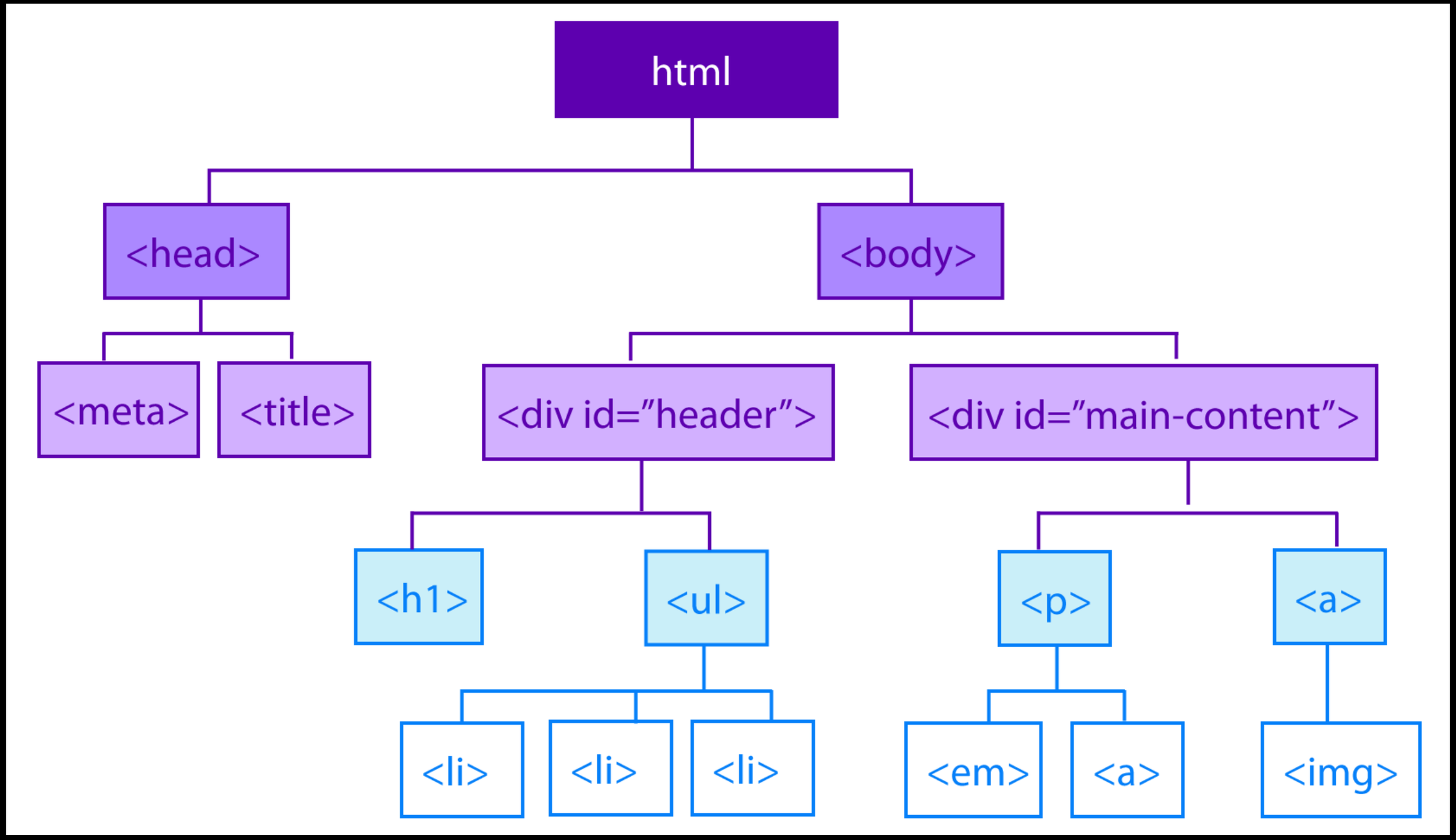
review: combining HTML and js

```
<html>
  <body>
    <p> This is a simple web page that contains a script.</p>

    <script>
      // some Javascript code
      alert("There be dragons here!");
    </script>

  </body>
</html>
```

review: combining HTML and js

```
<html>
  <body>
    <p> This is a simple web page that contains a script.</p>

    <script>
      // some Javascript code
      alert("There be dragons here!");
    </script>

  </body>
</html>
```

document object model (DOM)



browser gives js access to the HTML doc using getElement

review: modifying HTML with js

```
<html>
  <body>
    <p> Here are two pictures of the same baboon:</p>
    <img id="first_image" src="baboon.jpg" width="100">
    <img id="second_image" src="baboon.jpg" width="100">

    <script>
      var imgElement = document.getElementById("second_image");
      // make the second image small
      imgElement.width = "50";
    </script>
  </body>
</html>
```

Callbacks

```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Callbacks

```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

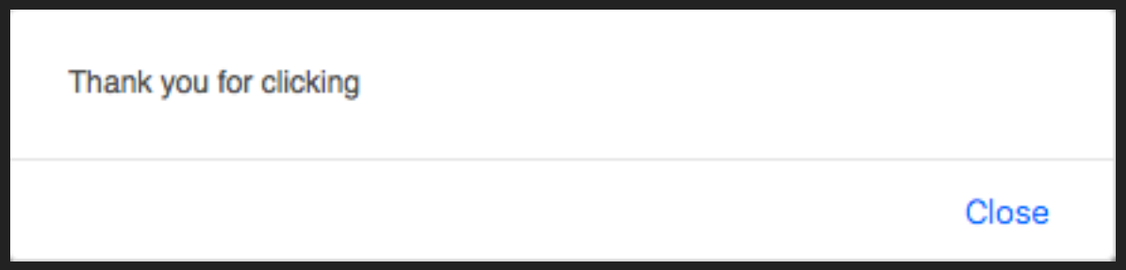
Callbacks

```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Callbacks

```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Callbacks




```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Callbacks

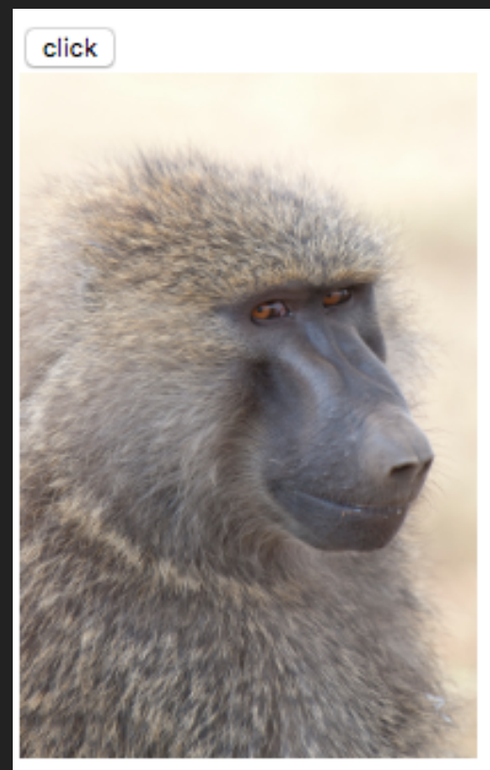
```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Callbacks

```
<html>

    <script>
      var clicked = function() {
        alert("Thank you for clicking");
      };
    </script>

  <body>
    <button onclick="clicked();">Click me!</button>
  </body>

</html>
```

Exercise

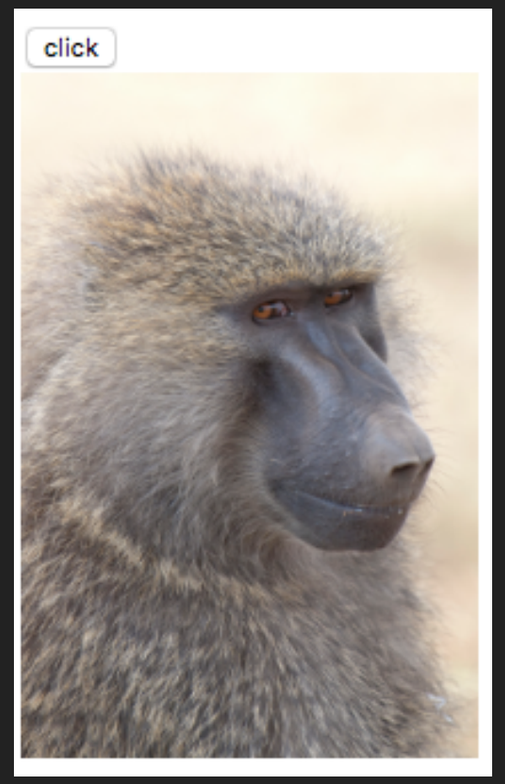
Create a webpage with a button and the baboon image. If the image is shown, then clicking the button will hide the image. If the image is not shown, then clicking the button will reveal the image.



Start slow and build up to your solution:

1) create a page with image and button.
2) write javascript function to show or hide image
3) ensure that clicking button calls the function

```
<html>
  <body>
    <button>click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">
  </body>
</html>
```



```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    function toggleImage() {
        document.getElementById("im").style.display = "none";
    }
    </script>
  </body>
</html>
```

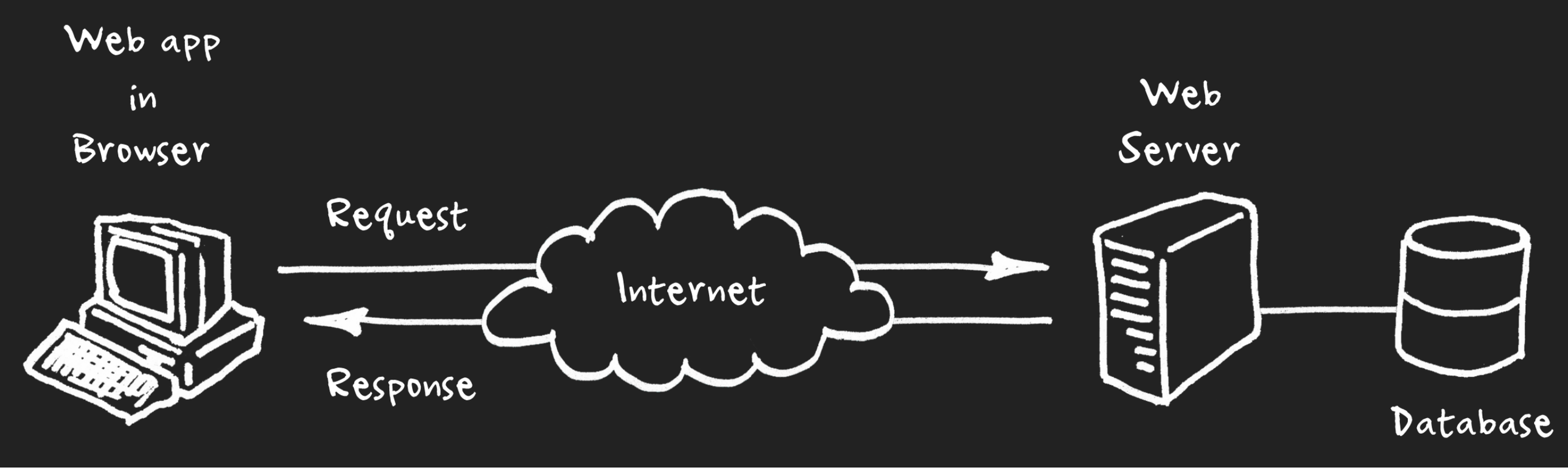
```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    var show = true;
    function toggleImage() {
        if( show ) {
            document.getElementById("im").style.display = "none";
            show = false;
        } else {
            document.getElementById("im").style.display = "";
            show = true;
        }
    }
    </script>
  </body>
</html>
```

```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    var show = true;
    function toggleImage() {
        if( show ) {
            document.getElementById("im").style.display = "none";
        } else {
            document.getElementById("im").style.display = "";
        }
        
    }
    </script>
  </body>
</html>
```

```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    var show = true;
    function toggleImage() {
        if( show ) {
            document.getElementById("im").style.display = "none";
        } else {
            document.getElementById("im").style.display = "";
        }
        show = !show;
    }
    </script>
  </body>
</html>
```

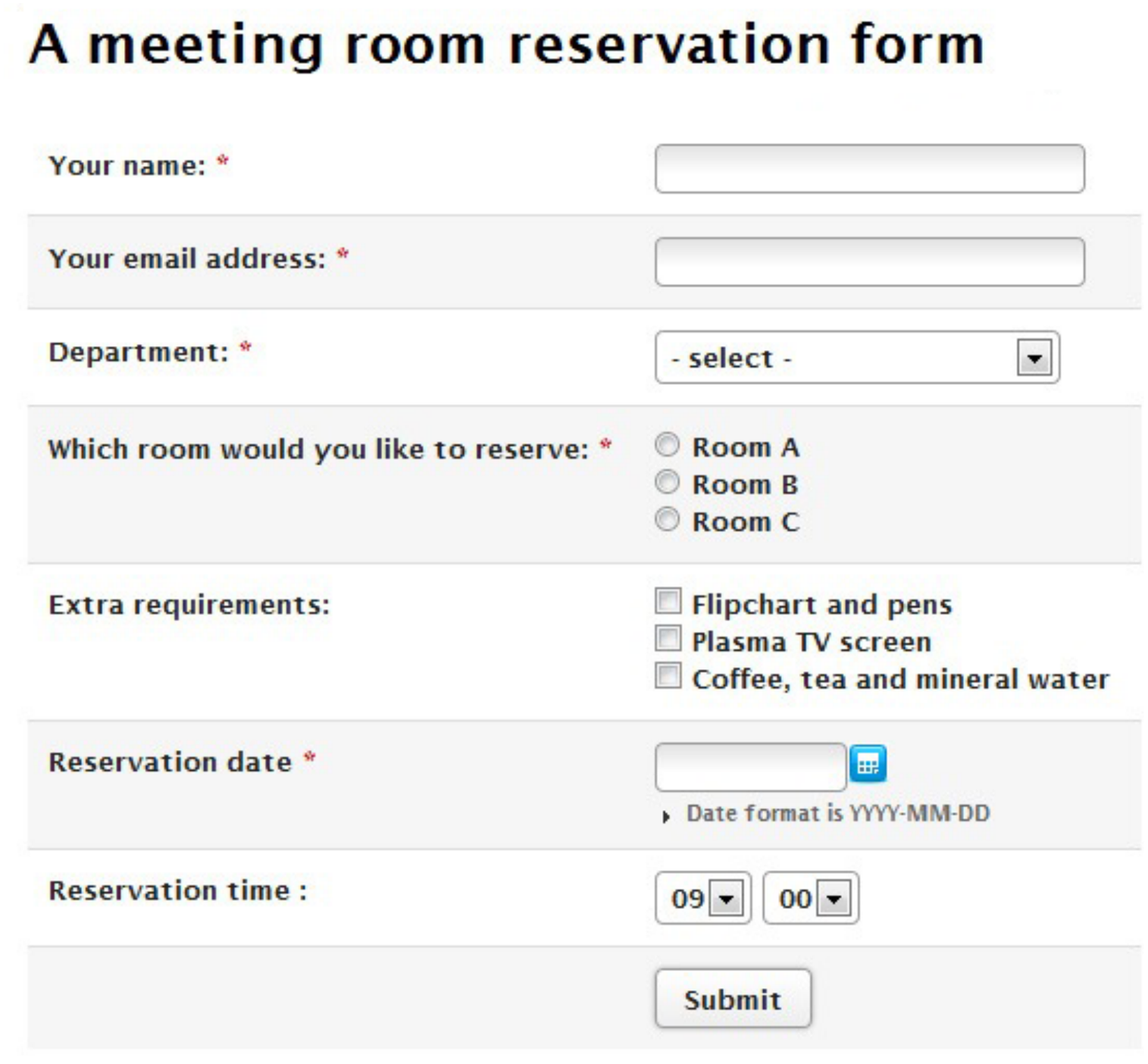
```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    function toggleImage() {
        if(                          ) {
            document.getElementById("im").style.display = "";
        } else {
            document.getElementById("im").style.display = "none";
        }
    }
    </script>
  </body>
</html>
```

```
<html>
  <body>
    <button onclick="toggleImage();">click</button>
    <br>
    <img id="im" src="baboon.jpg" height="300">

    <script>
    function toggleImage() {
        var visible = document.getElementById("im").style.display;
        if( visible === "none" ) {
            document.getElementById("im").style.display = "";
        } else {
            document.getElementById("im").style.display = "none";
        }
    }
    </script>
  </body>
</html>
```



1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)

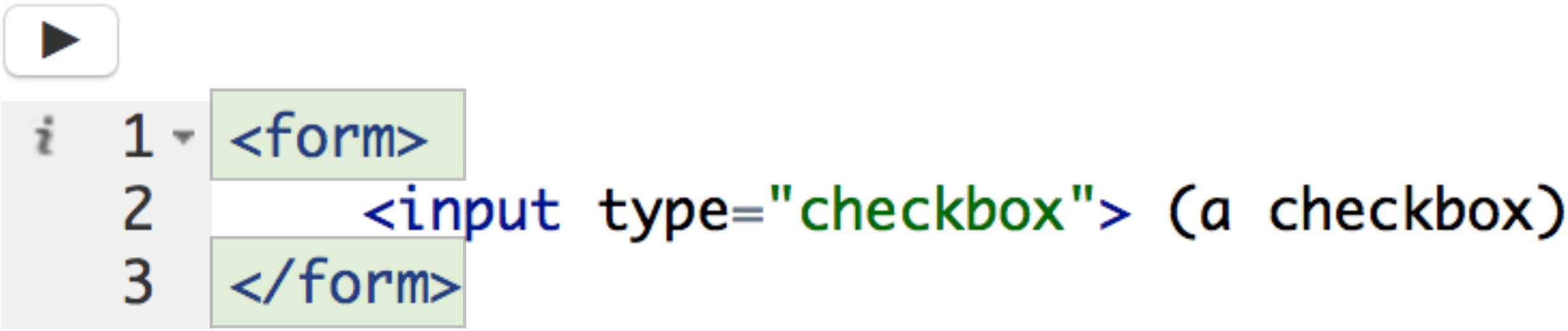
collecting input: forms



Forms (checkbox input)

```
1 <form>
2     <input type="checkbox"> (a checkbox)
3 </form>
```

Forms (checkbox input)



Forms (checkbox input)

```
1 <form>
2     <input type="checkbox"> (a checkbox)
3 </form>
```

Forms (checkbox input)

```
1 <form>
2     <input type="checkbox"> (a checkbox)
3 </form>
```

Forms (checkbox input)

```
1 <form>
2     <input type="checkbox"> (a checkbox)
3 </form>
```

Forms (checkbox input)

```
1 <form>
2     <input type="checkbox" checked> (a checkbox)
3 </form>
```

Forms (text input)

```
1 <form>
2     <input type="text" size="30" value="default text">
3 </form>
```

Forms (text input)

```
1 <form>
2     <input type="text" size="30" value="default text">
3 </form>
```

Forms (password input)

```
1 <form>
2     <input type="password" size="30">
3 </form>
```

Forms (password input)

```
1 <form>
2     <input type="password" size="30">
3 </form>
```

Forms (radio input)

```
1 <form>
2     <input type="radio" value="A" id="choice">
3     <input type="radio" value="B" id="choice" checked>
4     <input type="radio" value="C" id="choice">
5     <input type="radio" value="D" id="choice">
6 </form>
```

Forms (radio input)

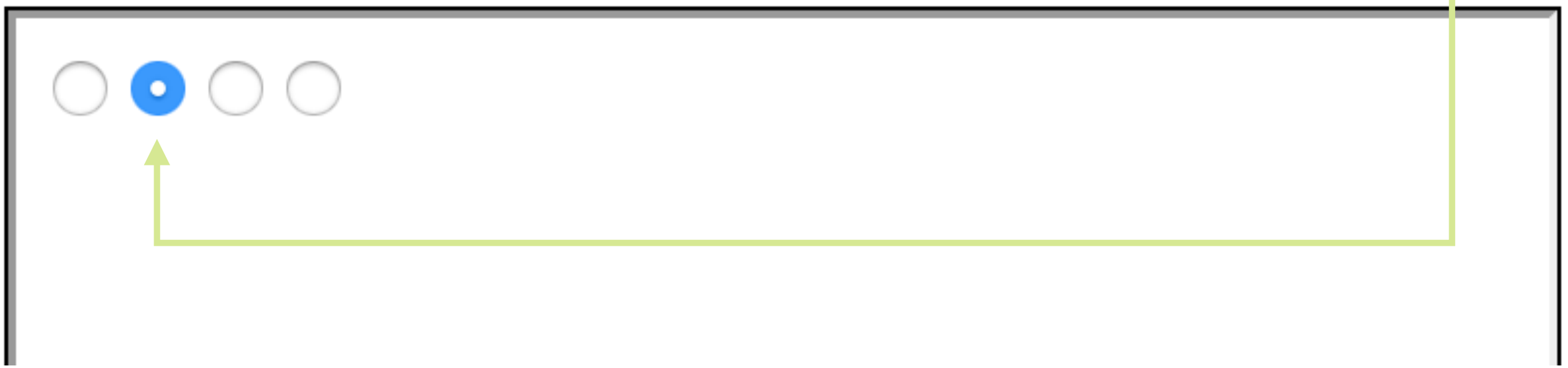
```
1 <form>
2     <input type="radio" value="A" id="choice">
3     <input type="radio" value="B" id="choice" checked>
4     <input type="radio" value="C" id="choice">
5     <input type="radio" value="D" id="choice">
6 </form>
```

Forms (radio input)

```
1 <form>
2     <input type="radio" value="A" id="choice">
3     <input type="radio" value="B" id="choice" checked>
4     <input type="radio" value="C" id="choice">
5     <input type="radio" value="D" id="choice">
6 </form>
```

Forms (radio input)

```
1 <form>
2     <input type="radio" value="A" id="choice">
3     <input type="radio" value="B" id="choice" checked>
4     <input type="radio" value="C" id="choice">
5     <input type="radio" value="D" id="choice">
6 </form>
```

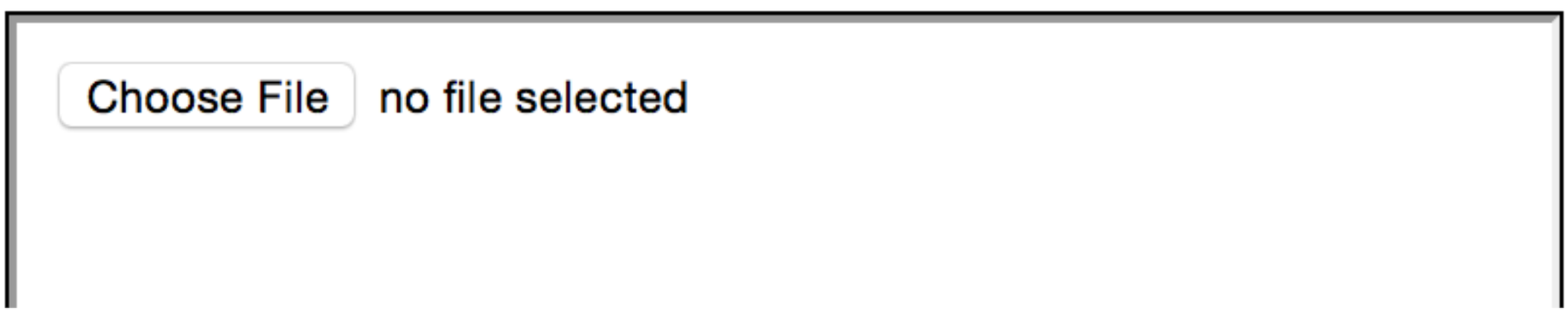


Forms (radio input)

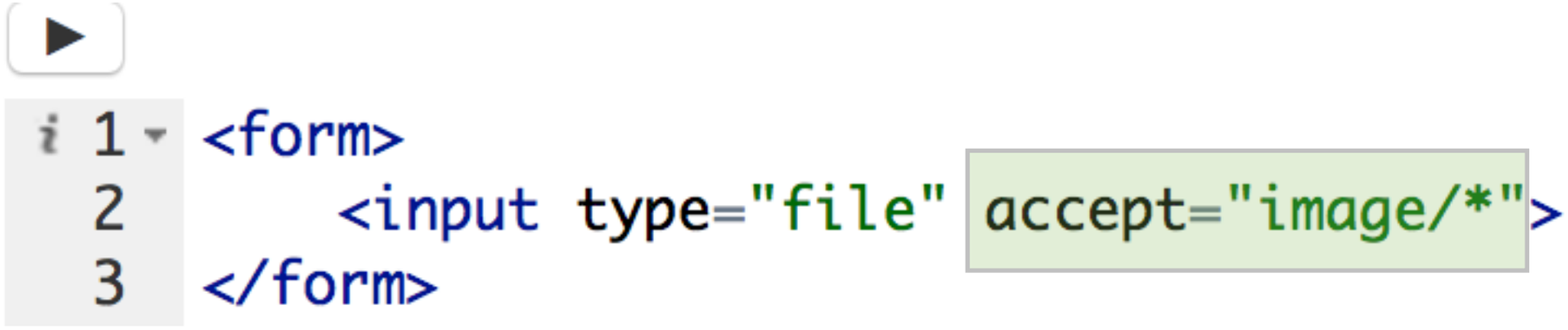
```
1 <form>
2     <input type="radio" value="A" id="choice">
3     <input type="radio" value="B" id="choice" checked>
4     <input type="radio" value="C" id="choice">
5     <input type="radio" value="D" id="choice">
6 </form>
```

Forms (file upload)

```
1 <form>
2     <input type="file">
3 </form>
```

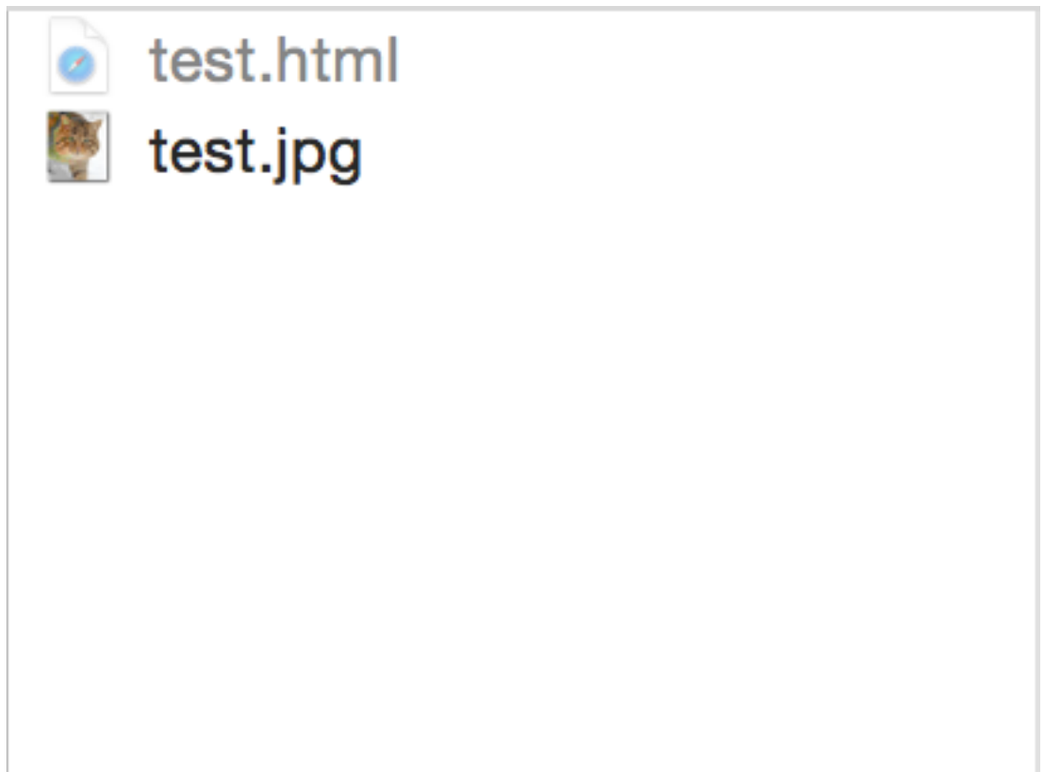


Forms (file upload)



Forms (file upload)

```html
<form>
    <input type="file" accept="image/*">
</form>
```



Forms (select input)

```
1 <form>
2     <select>
3       <option>artichoke</option>
4       <option>eggplant</option>
5       <option>mushrooms</option>
6       <option>peppers</option>
7     </select>
8 </form>
```

Forms (buttons)

```
1 <button onclick="console.log('OK')">OK</button>
```

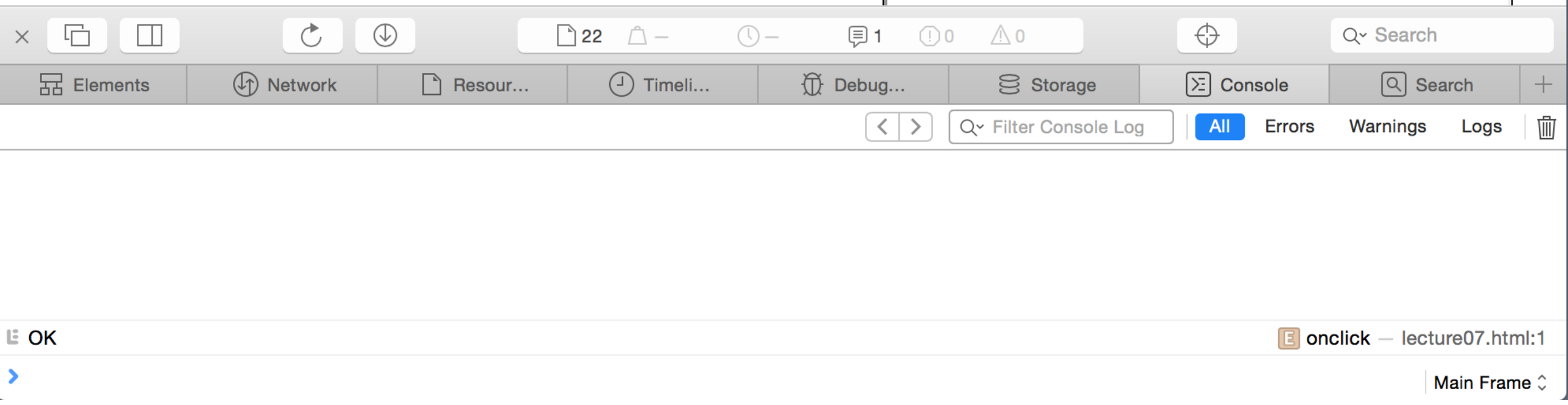
Forms (file upload)

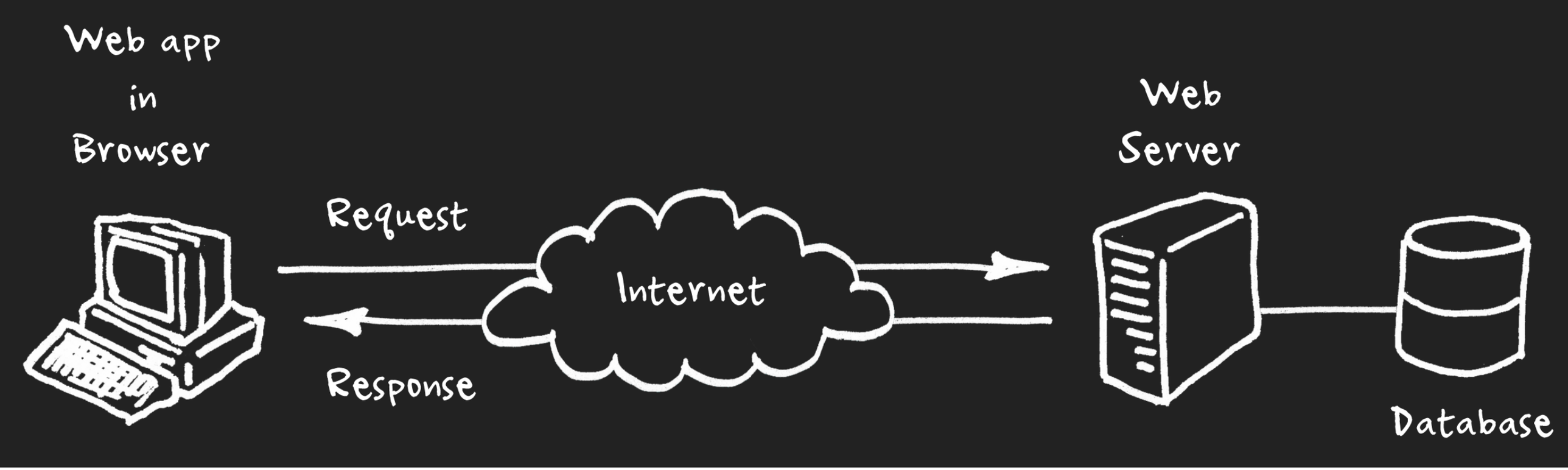
```
1 <button onclick="console.log('OK')">OK</button>
```

callback

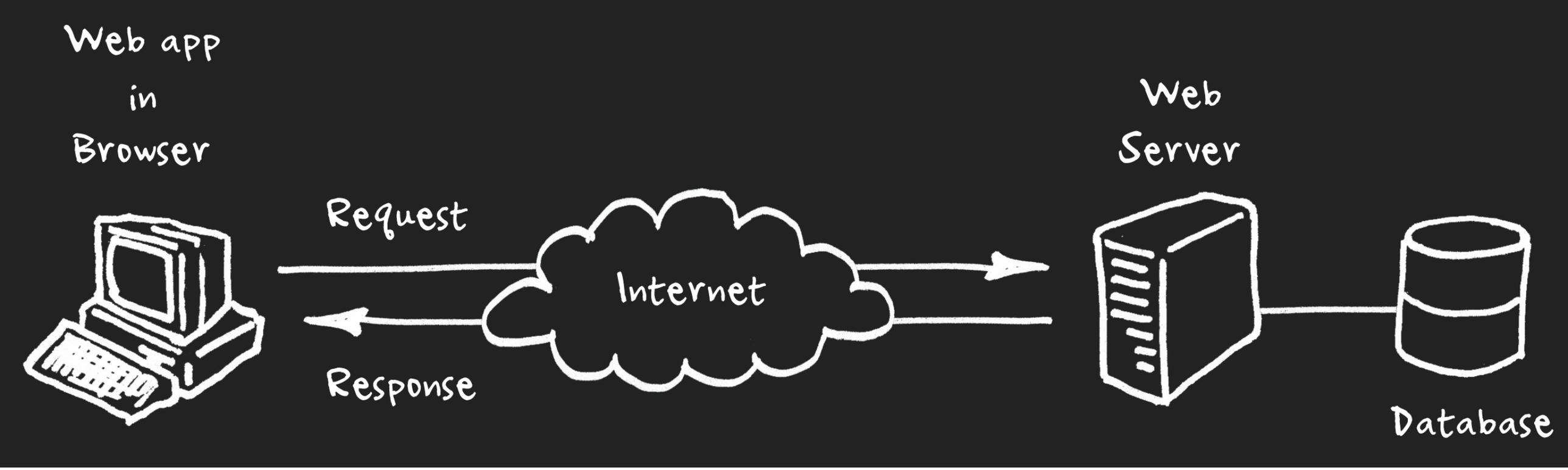
Forms (file upload)

```
1 <button onclick="console.log('OK')">OK</button>
2
3
4
5
6
7
```





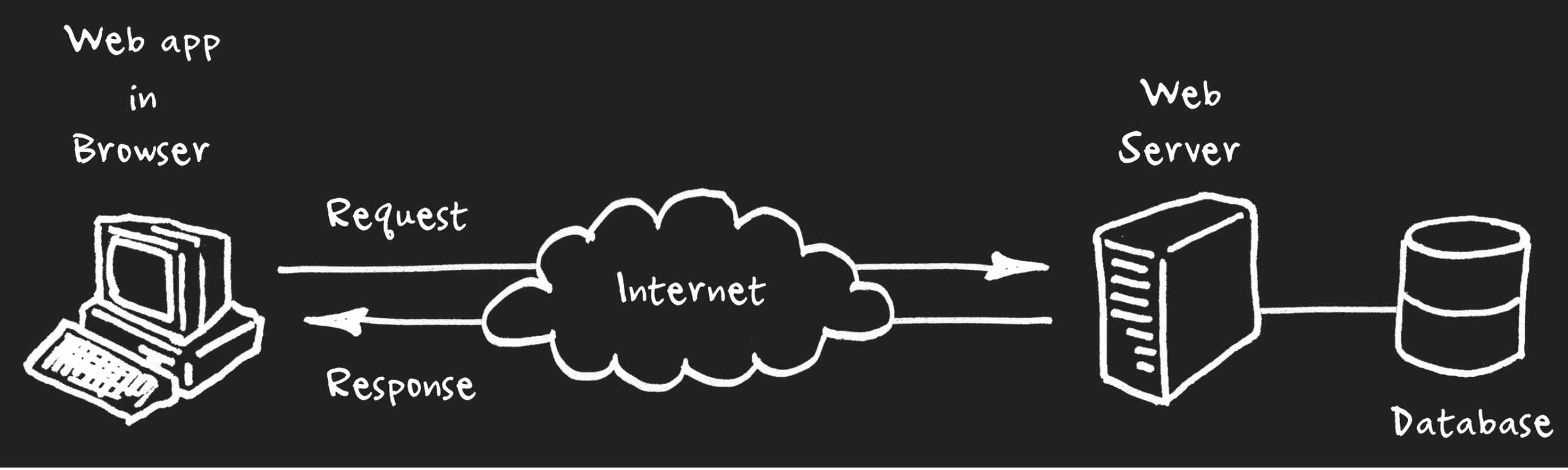
1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)



1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)

validation

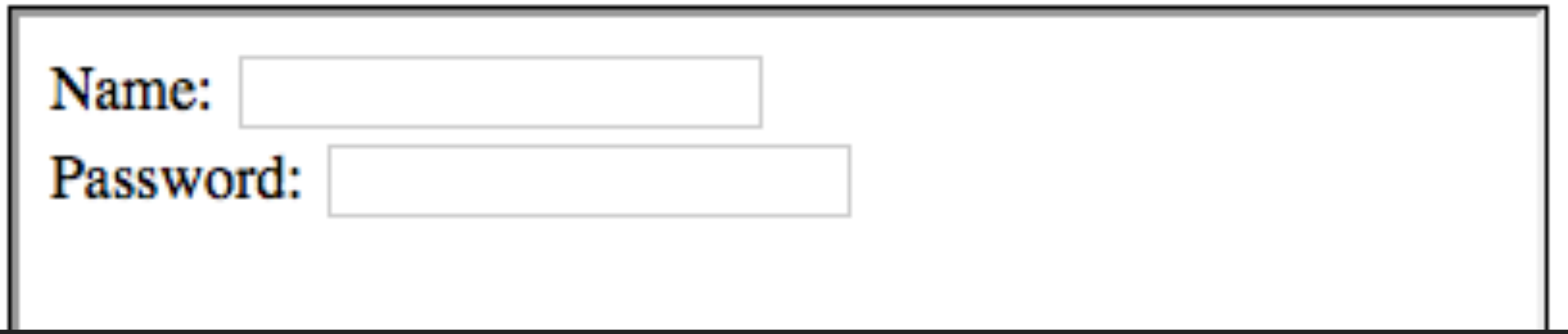
1. client-side
   - part of responsive design
   - limit denial-of-service (DOS) attacks
   - libraries: Parsley.js, validate.js, Verify.js, gvalidator
2. server-side: limit data corruption attacks



1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store, package, and manipulate data (objects)

Submitting forms (client-side)

```
1  <form>
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4  </form>
```



Submitting forms (client-side)

```
1  <form>
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4  </form>
```

Name:

Password:

name (server) vs. id (client)

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4     <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```html
<form action="submit.php" method="POST">
  Name: <input type="text" name="name"> <br>
  Password: <input type="password" name="password"> <br>
  <button type="submit">Log in</button>
</form>
```

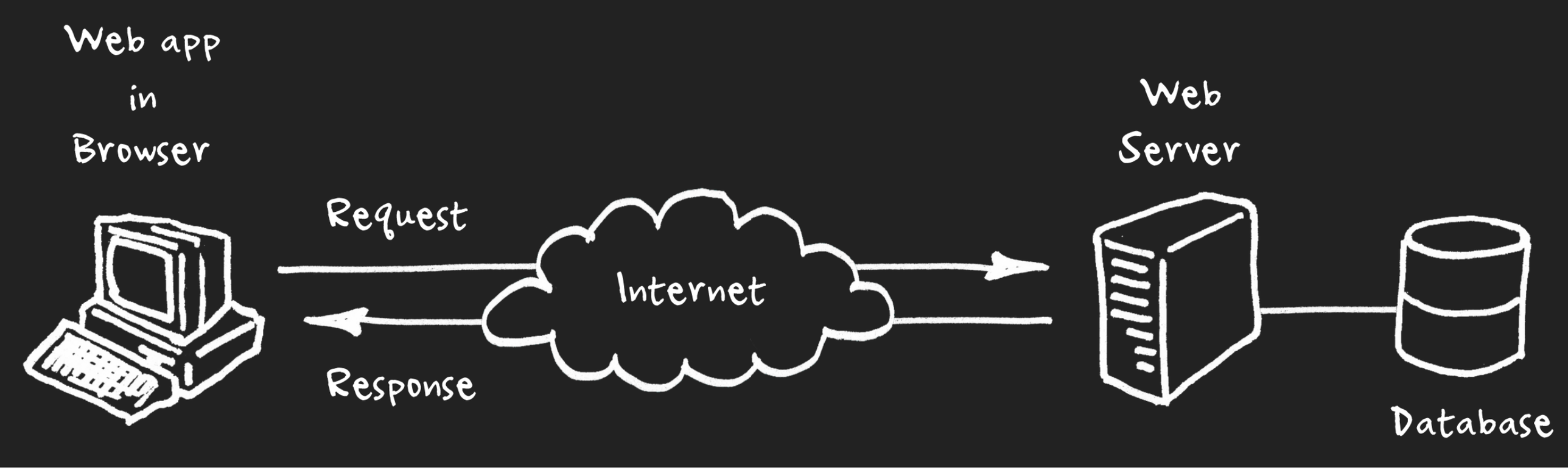
Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Submitting forms (client-side)

```
1  <form action="submit.php" method="POST">
2    Name: <input type="text" name="name"> <br>
3    Password: <input type="password" name="password"> <br>
4    <button type="submit">Log in</button>
5  </form>
```

Welcome hanyfarid, your password has 12 characters



1. Display info (HTML / css)
2. Collect input (HTML forms)
3. Make use of input on client (javascript, callbacks)
   A. responsiveness (modify HTML with js, DOM)
   B. validation (modify input with js)
4. Send input to server (HTML forms + POST)
5. Store and manipulate data server-side (PHP, etc)