



CSC 405

Dynamic Web Pages

Alexandros Kapravelos
akaprav@ncsu.edu

In the beginning...

In the beginning...

HTML was static!





[Yahoo! Deutschland](#) [CLICK HERE TO VISIT THE STARS](#) [YAHOO! LOS ANGELES](#) [Weekly Picks](#)

 [Options](#)

[Yellow Pages](#) - [People Search](#) - [City Maps](#) -- [News Headlines](#) - [Stock Quotes](#) - [Sports Scores](#)

- [Arts](#) - - *Humanities, Photography, Architecture, ...*
- [Business and Economy \[Xtra!\]](#) - - *Directory, Investments, Classifieds, ...*
- [Computers and Internet \[Xtra!\]](#) - - *Internet, WWW, Software, Multimedia, ...*
- [Education](#) - - *Universities, K-12, Courses, ...*
- [Entertainment \[Xtra!\]](#) - - *TV, Movies, Music, Magazines, ...*
- [Government](#) - - *Politics [Xtra!], Agencies, Law, Military, ...*
- [Health \[Xtra!\]](#) - - *Medicine, Drugs, Diseases, Fitness, ...*
- [News \[Xtra!\]](#) - - *World [Xtra!], Daily, Current Events, ...*
- [Recreation and Sports \[Xtra!\]](#) - - *Sports, Games, Travel, Autos, Outdoors, ...*
- [Reference](#) - - *Libraries, Dictionaries, Phone Numbers, ...*
- [Regional](#) - - *Countries, Regions, U.S. States, ...*
- [Science](#) - - *CS, Biology, Astronomy, Engineering, ...*
- [Social Science](#) - - *Anthropology, Sociology, Economics, ...*
- [Society and Culture](#) - - *People, Environment, Religion, ...*

[Yahoo! New York](#) - [Yahoo! Shop](#) - [Yahooligans!](#)

[Yahoo! Japan](#) - [Yahoo! Internet Life](#) - [Yahoo! San Francisco](#)



Welcome to Amazon.com Books!

*One million titles,
consistently low prices.*

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

SPOTLIGHT! -- AUGUST 16TH

These are the books we love, offered at Amazon.com low prices. The spotlight moves **EVERY** day so please come often.

ONE MILLION TITLES

Search Amazon.com's [million title catalog](#) by author, subject, title, keyword, and more... Or take a look at the [books we recommend](#) in over 20 categories... Check out our [customer reviews](#) and the [award winners](#) from the Hugo and Nebula to the Pulitzer and Nobel... and [bestsellers](#) are 30% off the publishers list...

EYES & EDITORS, A PERSONAL NOTIFICATION SERVICE

Like to know when that book you want comes out in paperback or when your favorite author releases a new title? Eyes, our tireless, automated search agent, will send you mail. Meanwhile, our human editors are busy previewing galleys and reading advance reviews. They can let you know when especially wonderful works are published in particular genres or subject areas. Come in, [meet Eyes](#), and have it all explained.

YOUR ACCOUNT

Check the status of your orders or change the email address and password you have on file with us. Please note that you **do not** need an account to use the store. The first time you place an order, you will be given the opportunity to create an account.



Search the web using Google!

Google Search

I'm feeling lucky

Special Searches

[Stanford Search](#)

[Linux Search](#)

[Help!](#)

[About Google!](#)

[Company Info](#)

[Google! Logos](#)

Get Google!
updates monthly:

your e-mail

[Archive](#)

Copyright ©1998 Google Inc.

Source: <https://web.archive.org/web/19981202230410/http://www.google.com/>

Dynamic HTML

- However, people soon realized that we could have websites interact with users based on inputs

JavaScript

- Client-Side scripting language for interacting and manipulating HTML
- Created by Brendan Eich at Netscape Navigator 2.0 in September 1995 as "LiveScript"
- Renamed to "JavaScript" in December 1995
- By August 1996, Microsoft added support for JavaScript to Internet Explorer
 - Microsoft later changed the name to JScript to avoid Sun's Java trademark
- Submitted to ECMA International for standardization on November 1996
- [ECMA-262](#), on June 1997, standardized first version of ECMAScript

JavaScript

- [Lingua franca](#) of the web
 - bridge language allowing users to execute code
- Eventually supported by all browsers
- Language organically evolved along the way

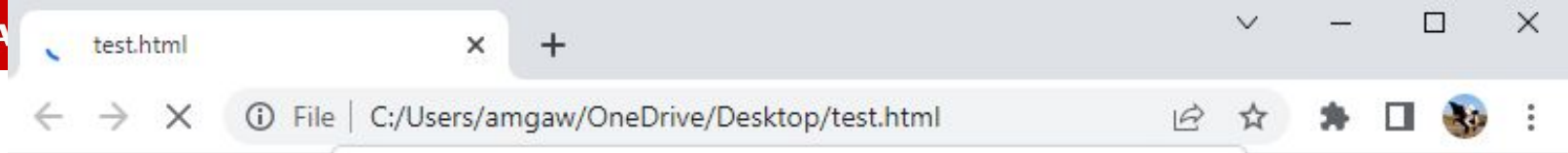
JavaScript

- Code can be embedded into HTML pages using the `script` element and (optionally storing the code in HTML comments)

```
<script>
var name = prompt('Please enter your name below.', '');
if (name == null) {
    document.write('Welcome to my site!');
} else {
    document.write('Welcome to my site ' + name + '!');
}
</script>
```

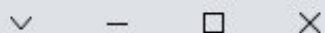
```
<script type="text/javascript">
<script language="javascript">
<script src="js/html2canvas.js">
```

Alternative Implementations



test.html

x +



File | C:/Users/amgaw/OneDrive/Desktop/test.html

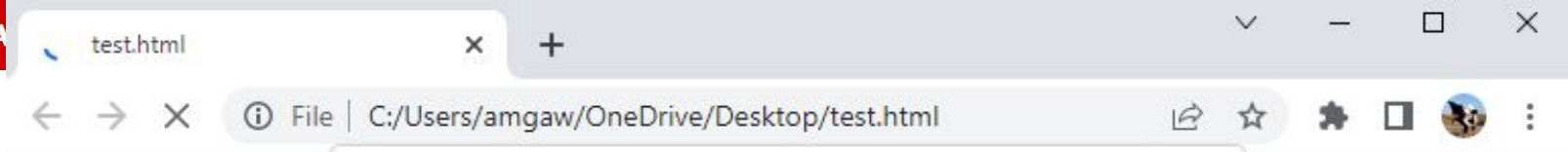


This page says

Please enter your name below.

OK

Cancel



test.html

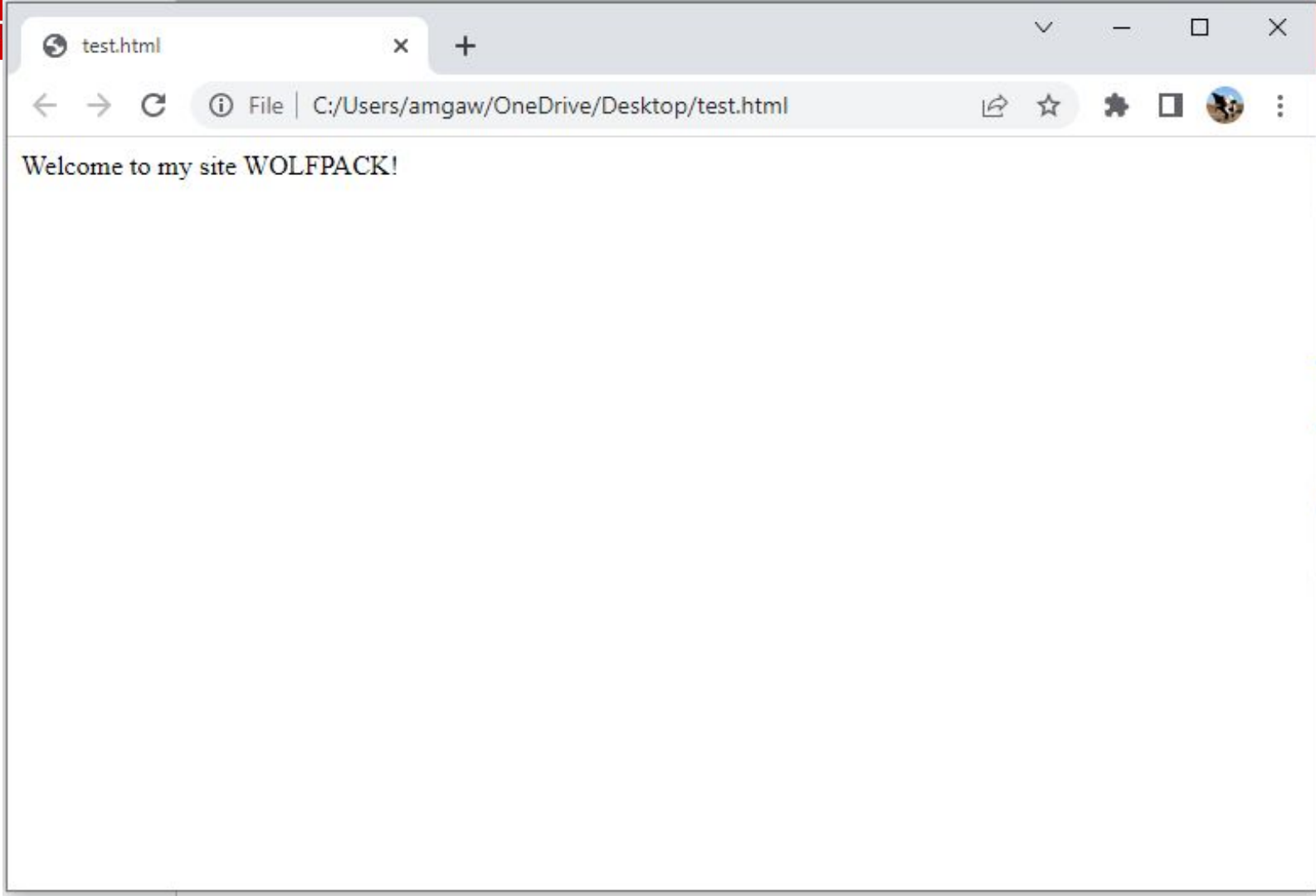


File | C:/Users/amgaw/OneDrive/Desktop/test.html



This page says

Please enter your name below.



Welcome to my site WOLFPACK!

JavaScript

- You can also include external JavaScript files in your HTML
 - As opposed to the inline JavaScript that we saw in the previous example
- `<script src="js/html2canvas.js">`
- When the browser parses this HTML element, it automatically fetches and executes the JavaScript before continuing to parse the rest of the HTML
 - Can also place at the **end** of HTML in case you want to render content, **then** execute off it
 - Placement of JS **matters**

DOM Example

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM Example</title>
  </head>
  <body>
    <h1>DOM Example</h1>
    <hr>
    <div id='insert_here'>Original Text</div>
  </body>
  <script>
    document.getElementById('insert_here').innerText = "New Text";
  </script>
</html>
```

DOM Example

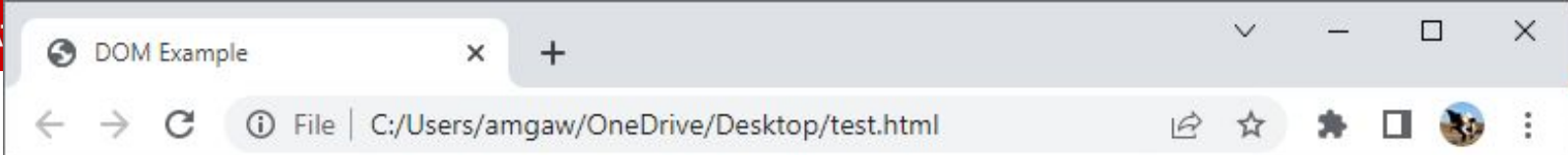
```
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM Example</title>
  </head>
  <body>
    <h1>DOM Example</h1>
    <hr>
    <div id='insert_here'>Original Text</div>
  </body>
  <script>
    document.getElementById('insert_here').innerText = "New Text";
  </script>
</html>
```

HTML is first
rendered...

THEN JavaScript
is executed

DOM Example

New Text



DOM Example

New Text

The `<div>` had "Original Text" when the page loaded.

But as soon as the JavaScript executed, the `innerText` variable changed the value to "New Text"

DOM Example

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM Example</title>
  </head>
  <body>
    <h1>DOM Example</h1>
    <hr>
    <div id='insert_here'>Original Text</div>
  </body>
  <script>
    var x = document.getElementById('insert_here');
    x.innerHTML = "<iframe src='evilsite.html'> </iframe>";
  </script>
</html>
```

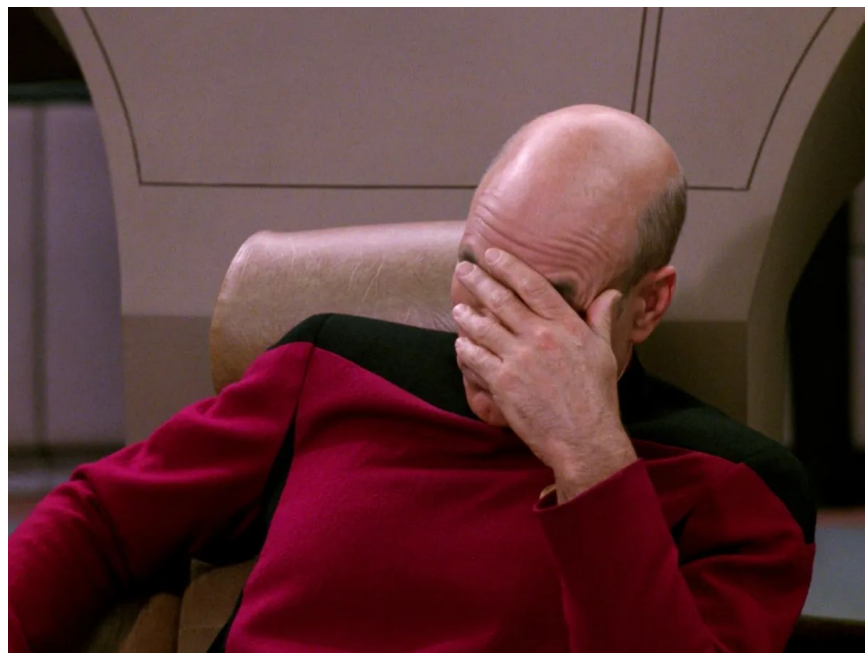
We can place **anything** for the browser to render

DOM Example

```
<iframe src='evilsite.html'> </iframe>
```

DOM Example

```
<iframe src='evilsite.html'> </iframe>
```



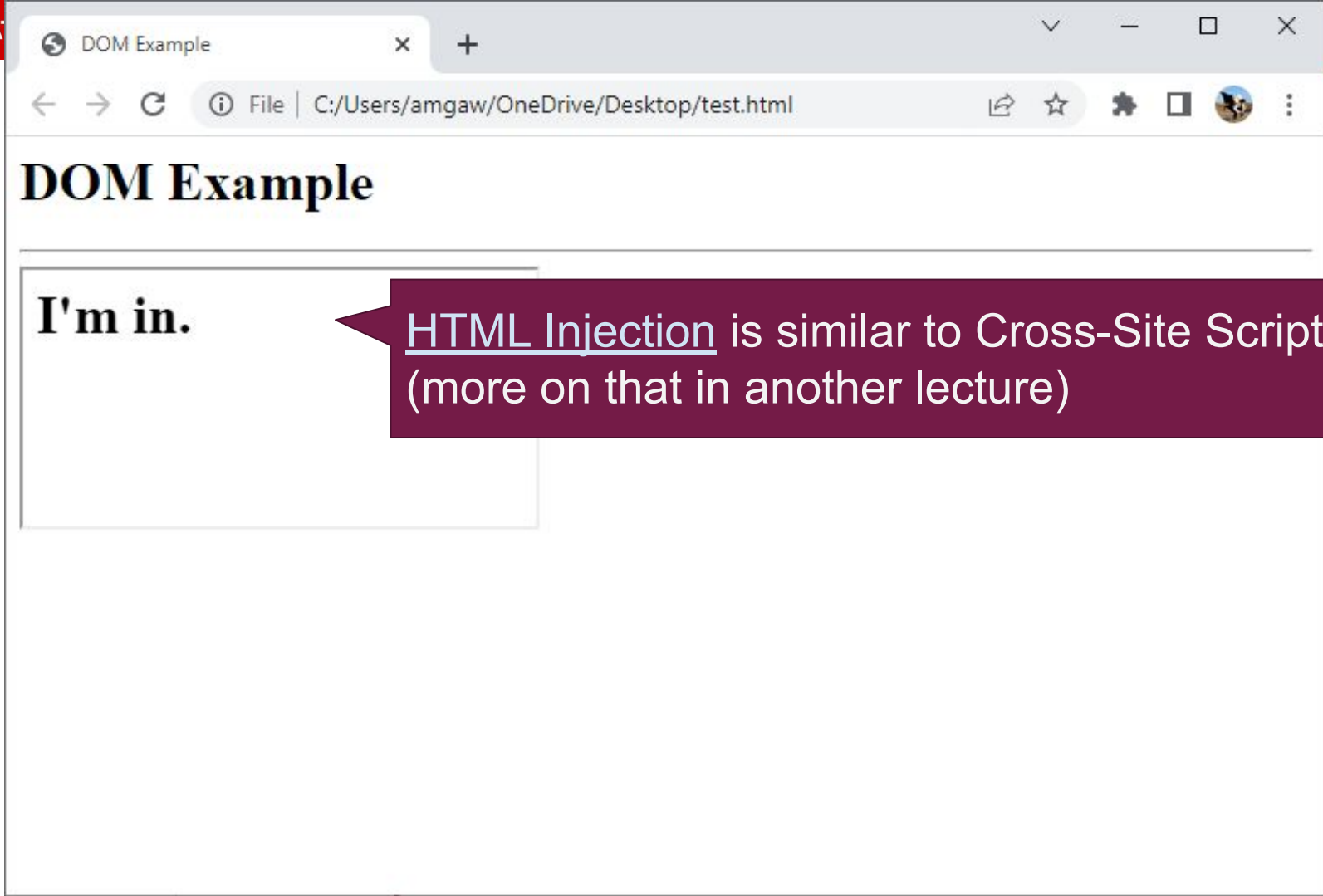
DOM Example

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>DOM Example</title>
  </head>
  <body>
    <h1>DOM Example</h1>
    <hr>
    <div id='insert_here'>Original Text</div>
  </body>
  <script>
    var x = document.getElementById('insert_here');
    x.innerHTML = "<iframe src='evilsite.html'> </iframe>";
  </script>
</html>
```

But it does need to
render...

DOM Example

I'm in.



The image shows a web browser window with a single tab titled "DOM Example". The address bar shows the file path "C:/Users/amgaw/OneDrive/Desktop/test.html". The page content includes a heading "DOM Example" and a text input field containing "I'm in.". A dark purple callout box with a white arrow pointing to the input field contains the text: "HTML Injection is similar to Cross-Site Scripting (more on that in another lecture)".

DOM Example

I'm in.

HTML Injection is similar to Cross-Site Scripting (more on that in another lecture)

Preventing HTML Injection

- **NEVER** let raw user input be rendered
- Escape HTML special characters to **&equivalent;**

```
<?php
$user_input = "<iframe src='evilsite.html'>";
$parsed = htmlspecialchars($user_input);
// converts to &lt;iframe src=&#039;evilsite.html&#039;&gt;
?>
```

PHP

Converts < to **<**; so when it is displayed it renders as text instead of legitimate HTML

```
desc = request.form.get("description")
desc = html.escape(desc, True) # Convert HTML entities to Unicode
```

Python

Using the DOM

- Coding proper DOM access in a cross-browser world is a **nightmare**
- Some highlights from <http://stackoverflow.com/questions/565641/what-cross-browser-issues-have-you-faced>
 - **Internet Explorer** does not replace ` `; or HTML char code `160`, you need to replace it w/ its Unicode equivalent `\u00a0`
 - In **Firefox**, a dynamically created input field inside a form (created using `document.createElement`) does not pass its value on form submission
 - `document.getElementById` in **Internet Explorer** will return an element even if the `name` attribute matches.
 - **Mozilla** only returns element if `id` matches

Browser Object Model (BOM)

- Programmatic interface to everything outside the document (aka the browser)
- No complete standard

- Examples

```
window.name = "New name"
```

```
window.close()
```

```
window.location = "http://example.com"
```

JavaScript vs. DOM and BOM

- JavaScript the language is defined separate from the DOM and BOM
 - DOM has its own specification, and much of the BOM is specified in HTML5 spec
- In the web context, these are often confused, because they are used together so often
- However, with JavaScript appearing everywhere, it's an important distinction
 - Server-side code using Node.js
 - Database queries (MongoDB)
 - Flash (dated has its own DOM-like capabilities)
 - Java applications (javax.script)
 - Windows applications (WinRT)

JavaScript – Object-based

- Almost everything in JavaScript is an object
 - Objects are associative arrays (hash tables), and the properties and values can be added and deleted at run-time

```
var object = {test: "foo", num: 50};  
object['foo'] = object;  
console.log(object[object['test']]);  
object.num = 1000;  
console.log(object['num']);
```

```
> var object = {test: "foo", num: 50};  
< undefined  
> object['foo'] = object;  
< ▼ Object {test: "foo", num: 50, foo: Object} ⓘ  
  ▶ foo: Object  
    num: 1000  
    test: "foo"  
  ▶ __proto__: Object  
> console.log(object[object['test']]);  
  ▶ Object {test: "foo", num: 50, foo: Object}  
< undefined  
> object.num = 1000;  
< 1000  
> console.log(object['num']);  
  1000  
< undefined  
>
```

JavaScript – Anonymous Functions and Closures

```
var createFunction = function() {  
    var count = 0;  
    return function () {  
        return ++count;  
    };  
};  
var inc = createFunction();  
inc();  
inc();  
inc();  
var inc2 = createFunction();  
inc2();  
...
```

```
> var createFunction = function() {  
    var count = 0;  
    return function () {  
        return ++count;  
    };  
};  
< undefined  
> var inc = createFunction();  
< undefined  
> inc();  
< 1  
> inc();  
< 2  
> inc();  
< 3  
> var inc2 = createFunction();  
< undefined  
> inc2();  
< 1  
>
```


JavaScript – Runtime Evaluation

- JavaScript contains features to interpret a string as code and execute it
 - eval
 - Function
 - setTimeout
 - setInterval
 - execScript (deprecated since IE11)

```
var foo = "bar";  
eval("foo = 'admin';");  
console.log(foo);  
var x = "console.log('hello');";  
var test = new Function(x);  
test();
```

```
> var foo = "bar";
```

```
< undefined
```

```
> eval("foo = 'admin';");
```

```
< "admin"
```

```
> console.log(foo);
```

```
admin
```

```
VM49:1
```

```
< undefined
```

```
> var x = "console.log('hello');";
```

```
< undefined
```

```
> var test = new Function(x);
```

```
< undefined
```

```
> test()
```

```
hello
```

```
VM54:2
```

```
< undefined
```

```
>
```

JavaScript Uses – Form Validation

- How to validate user input on HTML forms?
- Traditionally requires a round-trip to the server, where the server checks if the input is valid

JavaScript Uses – Form Validation

```
<?php
if ($_GET['submit']) {
    $student = $_GET['student'];
    $class = $_GET['class'];
    $grade = $_GET['grade'];
    if (empty($student) || empty($class) || empty($grade)) {
        echo "<b>Error, did not fill out all the forms</b>";
    }
    else if (!(($grade == 'A' || $grade == 'B' || $grade == 'C' ||
        $grade == 'D' || $grade == 'F')) {
        echo "<b>Error, grade must be one of A, B, C, D, or F</b>";
    }
    else { echo "<b>Grade successfully submitted!</b>";
    }
} ?>
<form>
Student: <input type="text" name="student"><br>
Class: <input type="text" name="class"><br>
Grade: <input type="text" name="grade"><br>
<input type="submit" name="submit">
</form>
```



Student:

Class:

Grade:



Student:

Class:

Grade:

Error, did not fill out all the forms

Student:

Class:

Grade:

localhost/ncsu/csc405/html_injec x +

localhost/ncsu/csc405/html_injection/grades.php?student=student01&class=CSC+405&grade=&submit=Submit

Error, did not fill out all the forms

Student:

Class:

Grade:

Submit

Inputs needed to go to the server to evaluate the user inputs, then render this error message

Error, did not fill out all the forms

Student:

Class:

Grade:

Notice that GET parameters are passed via the address bar

Insecure when dealing with network sniffers

Solution: USE HTTPS!

<https://letsencrypt.org/>

Error, did not fill out all the forms

Student:

Class:

Grade:



Error, grade must be one of A, B, C, D, or F

Student:

Class:

Grade:

Submit



Error, grade must be one of A, B, C, D, or F

Student:

Class:

Grade:



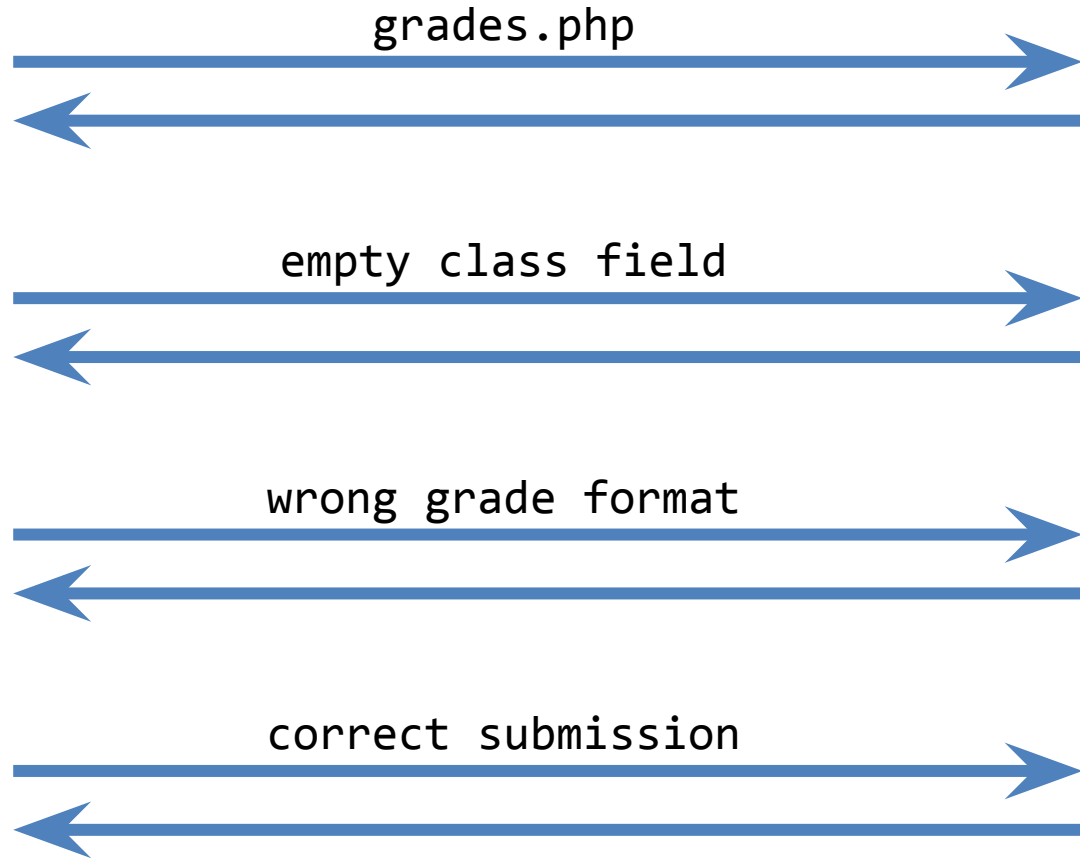
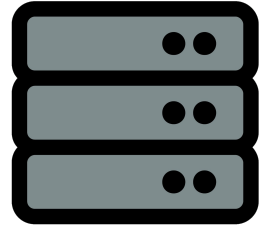
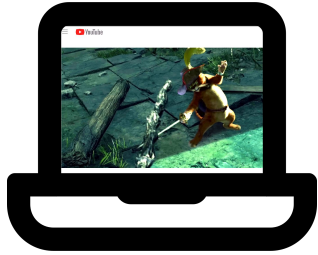
Grade successfully submitted!

Student:

Class:

Grade:

Submit



JavaScript Uses – Form Validation

- How to validate user input on HTML forms?
- Traditionally requires a round-trip to the server, where the server can check the input to make sure that it is valid
 - But we can also do it **client-side**

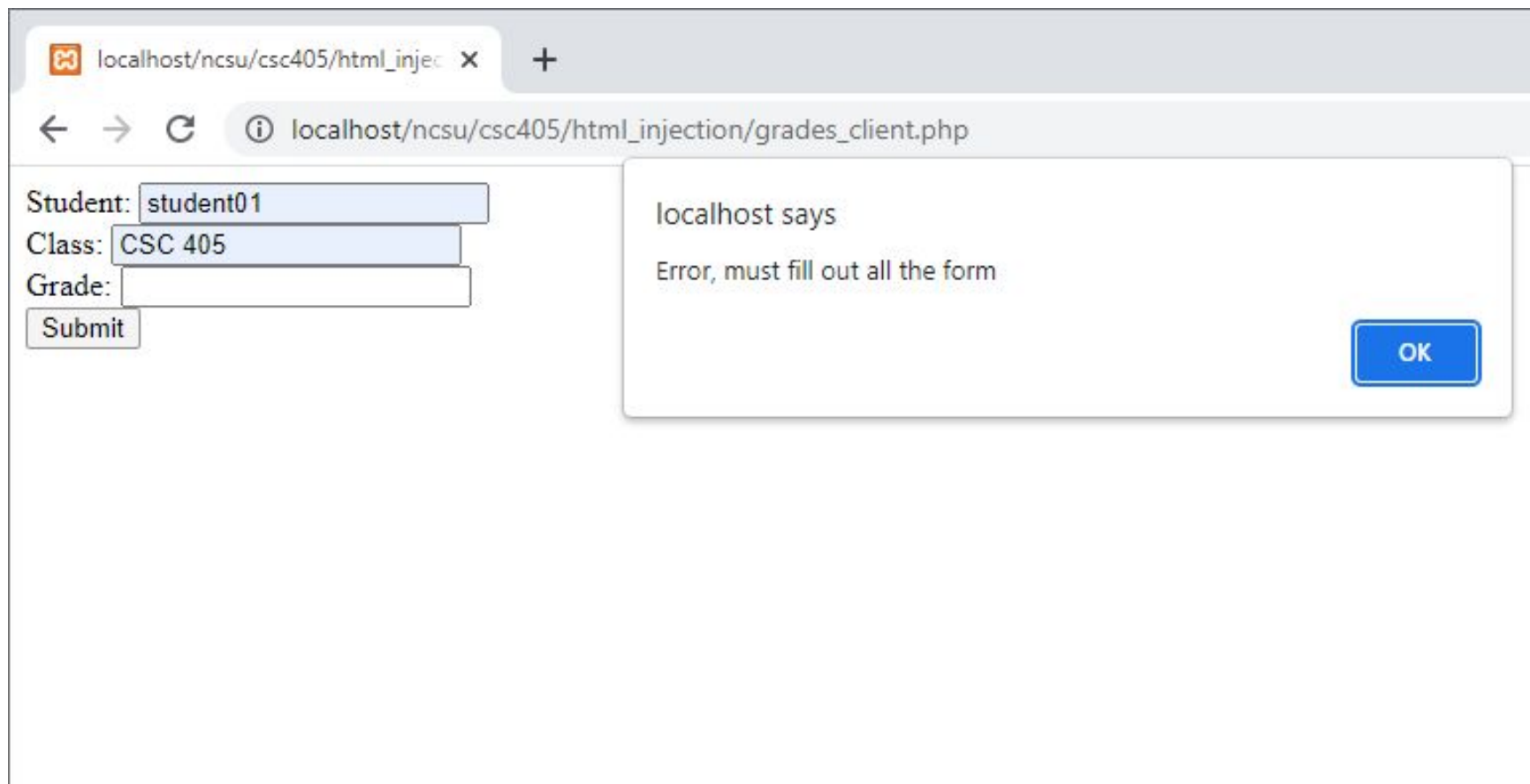
JavaScript Uses – Form Validation

```
<script>
function check_form() {
    var form = document.getElementById("the_form");
    if (form.student.value == "" || form.class.value == "" || form["grade"].value == ""){
        alert("Error, must fill out all the form");
        return false;
    }
    var grade = form["grade"].value;
    if (!(grade == 'A' || grade == 'B' || grade == 'C' ||
        grade == 'D' || grade == 'F')) {
        alert("Error, grade must be one of A, B, C, D, or F");
        return false;
    }
    return true;
}
</script>
<form id="the_form" onsubmit="return check_form()">
    Student: <input type="text" name="student"><br>
    Class: <input type="text" name="class"><br>
    Grade: <input type="text" name="grade"><br>
    <input type="submit" name="submit">
</form>
```


JavaScript Uses – Form Validation

```
<script>
function check_form() {
  var form = document.getElementById("the_form");
  if (form.student.value == "" || form.class.value == "" || form["grade"].value == ""){
    alert("Error, must fill out all the form");
    return false;
  }
  var grade = form["grade"].value;
  if (!(grade == 'A' || grade == 'B' || grade == 'C' ||
    grade == 'D' || grade == 'F')) {
    alert("Error, grade must be one of A, B, C, D, or F");
    return false;
  }
  return true;
}
</script>
<form id="the_form" onsubmit="return check_form()">
  Student: <input type="text" name="student"><br>
  Class: <input type="text" name="class"><br>
  Grade: <input type="text" name="grade"><br>
  <input type="submit" name="submit">
</form>
```

Clicking Submit triggers `check_form` and only if it returns `true` do we send the data to the server



The screenshot shows a web browser window with a single tab titled "localhost/ncsu/csc405/html_injec". The address bar displays "localhost/ncsu/csc405/html_injection/grades_client.php". On the left side of the page, there is a form with three input fields: "Student:" containing "student01", "Class:" containing "CSC 405", and "Grade:" which is empty. Below these fields is a "Submit" button. On the right side, a modal error message box is displayed with the text "localhost says" and "Error, must fill out all the form". An "OK" button is located in the bottom right corner of the error message box.

localhost/ncsu/csc405/html_injec x +

localhost/ncsu/csc405/html_injection/grades_client.php

Student: student01

Class: CSC 405

Grade:

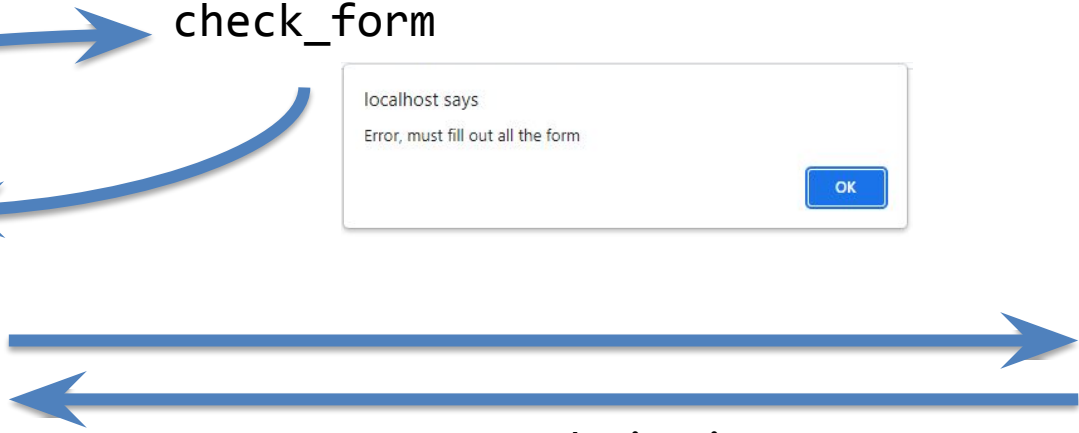
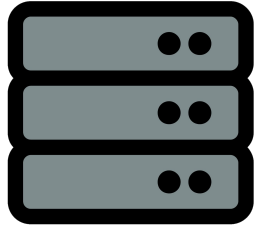
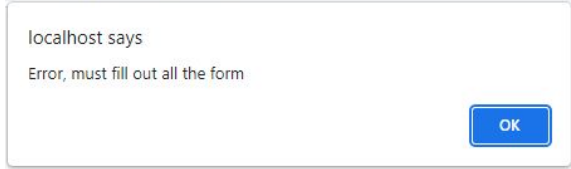
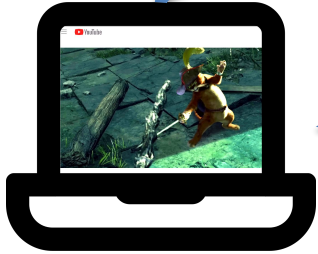
Submit

localhost says
Error, must fill out all the form

OK

grades_client.php

check_form



correct submission

Client-Side Validation

- Now that we're doing validation on the client, can we get rid of all those checks in our server-side code?
 - No!
 - No guarantee that client-side validation is performed
 - User disables JavaScript
 - Command-line clients

Client-Side Validation

- Now that we're doing validation on the client, can we get rid of all those checks in our server-side code?
 - No!
 - No guarantee that client-side validation is performed
 - User disables JavaScript
 - Command-line clients
- Users could enter arbitrary data that does not conform to your validation
 - Could lead to a security compromise or not

Client-Side Validation

- Now that we're doing validation on the client, can we get rid of all those checks in our server-side code?
 - No!
 - No guarantee that client-side validation is performed
 - User disables JavaScript
 - Command-line clients
- Users could enter arbitrary data that does not conform to your validation
 - Could lead to a security compromise or not
- So the validation must remain on the server-side and the client-side
 - Brings up another problem, how to perform consistent validation when server-side and client-side written in different languages