NC STATE UNIVERSITY

CSC 405
Dynamic Web Pages

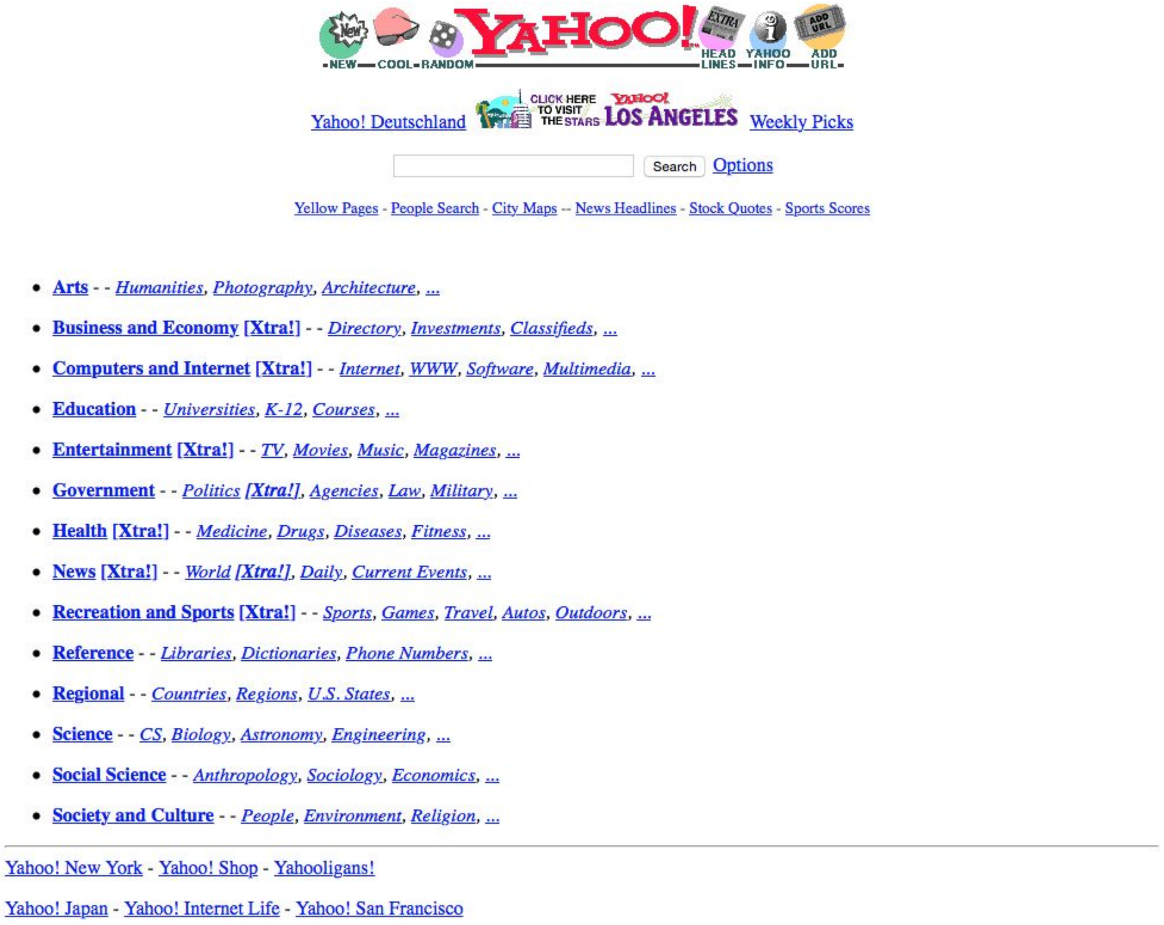
Adam Gaweda
agaweda@ncsu.edu

Alexandros Kapravelos
akaprav@ncsu.edu

In the beginning…

In the beginning…

HTML was static!



Yahoo! Deutschland Weekly Picks

Search Options

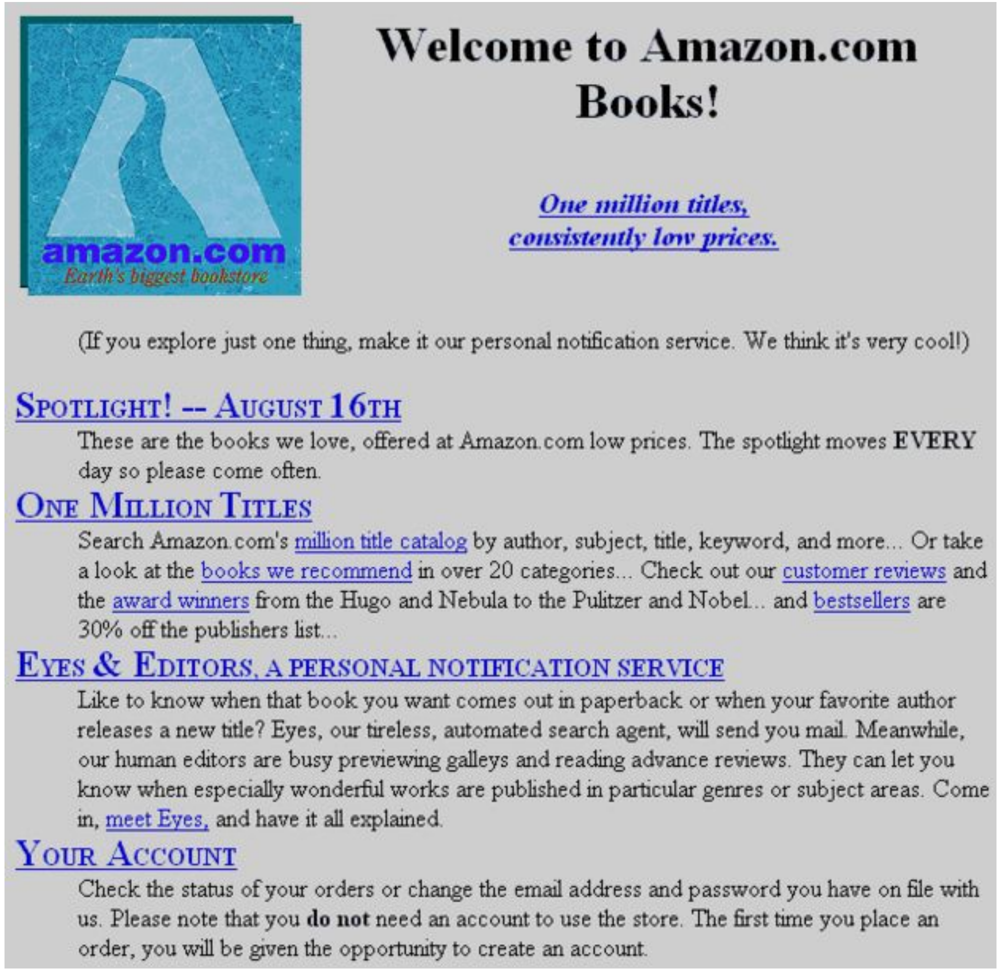
Yellow Pages - People Search - City Maps -- News Headlines - Stock Quotes - Sports Scores

- **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

Source: https://web.archive.org/web/19961017235908/http://www2.yahoo.com/



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

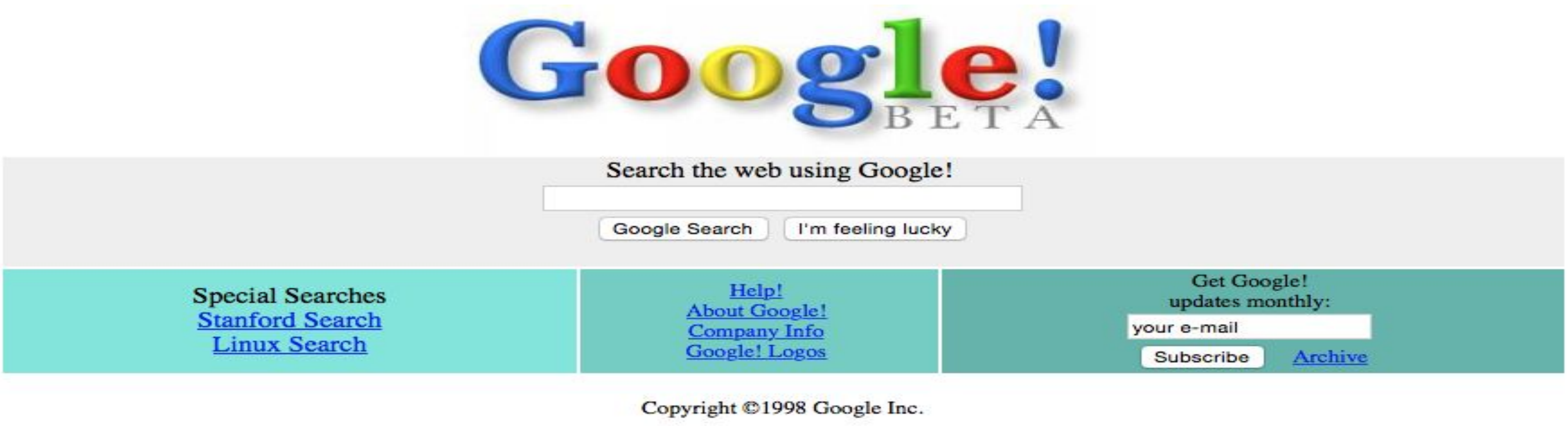
Search Amazon.com's million title catalog by author, subject, title, keyword, and more... Or take a look at the books we recommend in over 20 categories... Check out our customer reviews and the award winners from the Hugo and Nebula to the Pulitzer and Nobel... and bestsellers are 30% off the publishers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

Special Searches
Stanford Search
Linux Search

Help!
About Google!
Company Info
Google! Logos

Get Google! updates monthly:

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

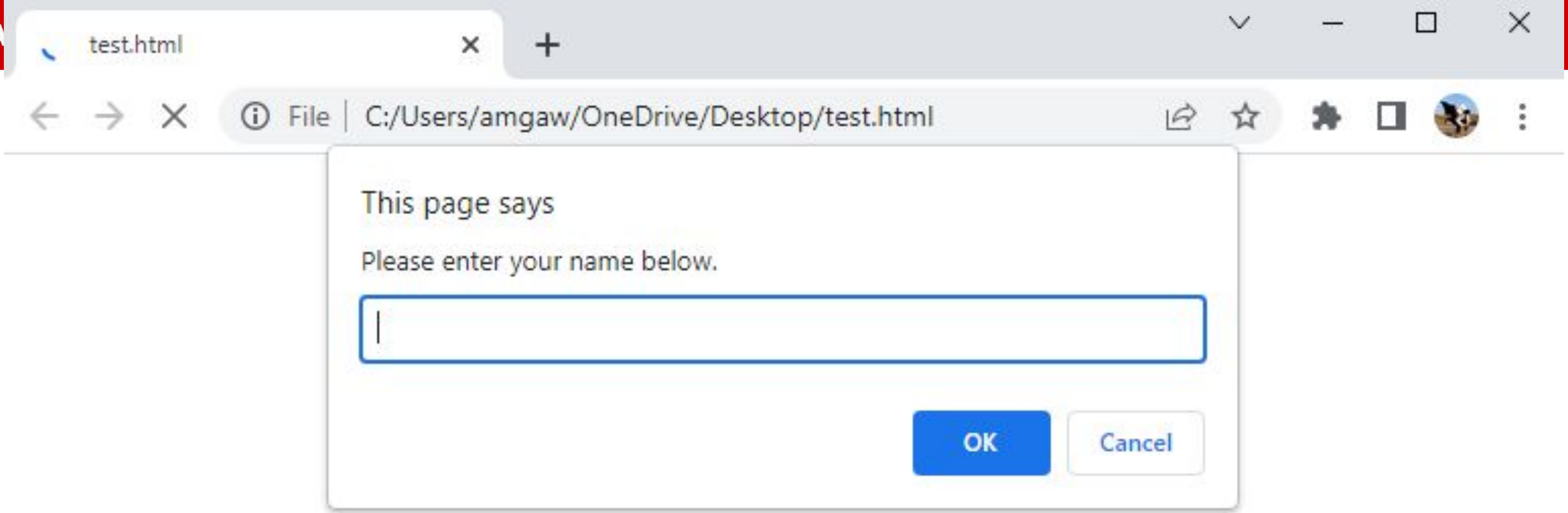
- Code can be embedded into HTML pages using the `script` element and (optionally storing the code in HTML comments)

```
<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

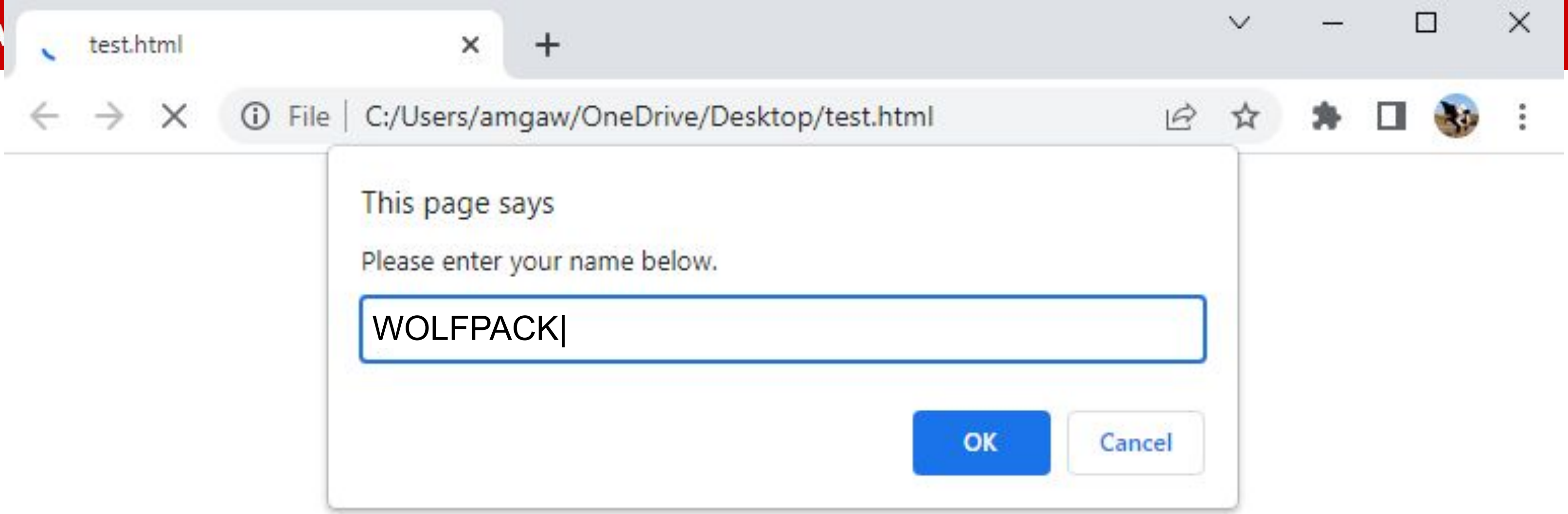
NC STA



This page says

Please enter your name below.

NC STA

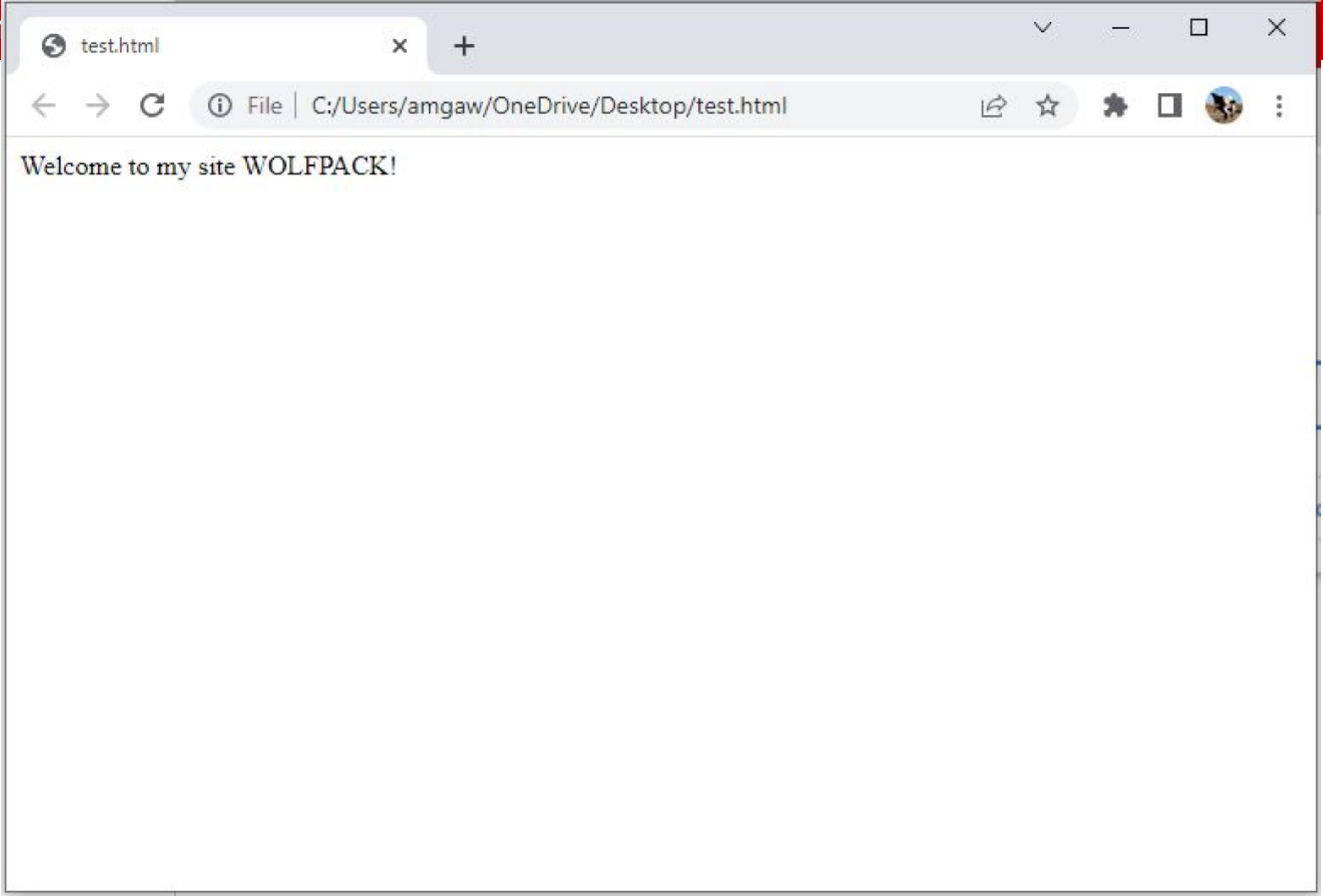


This page says

Please enter your name below.

WOLFPACK|

NC STAT



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

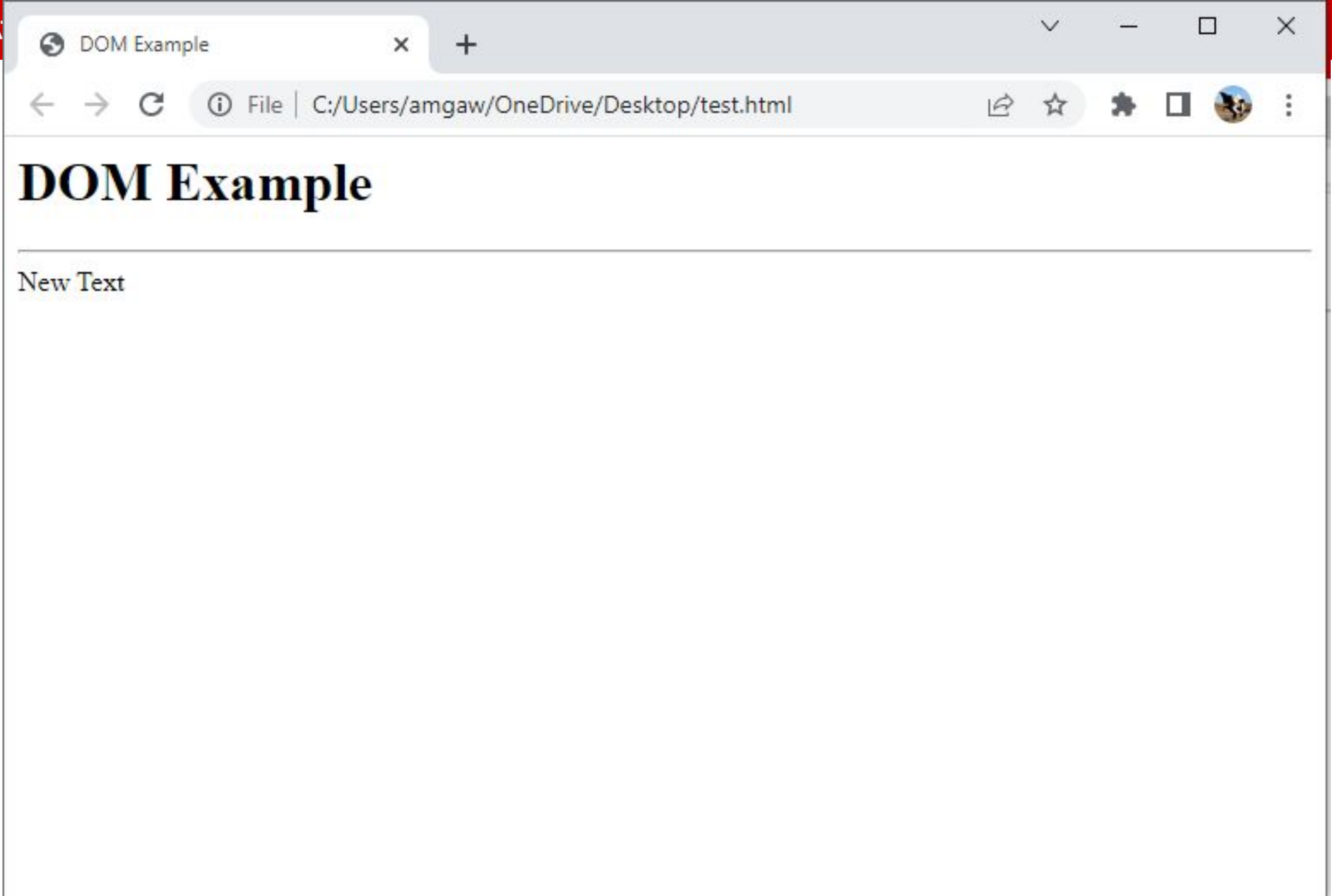
```
<html>
 <head>
   <meta charset="UTF-8">
   <title>DOM Example</title>
 </head>
 <body>
  <h1>DOM Example</h1>
  <hr>
  <div id='insert_here'>Original Text</div>
 </body>
 <script>
   document.getElementById('insert_here').innerText = "New Text";
 </script>
</html>
```

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>
```

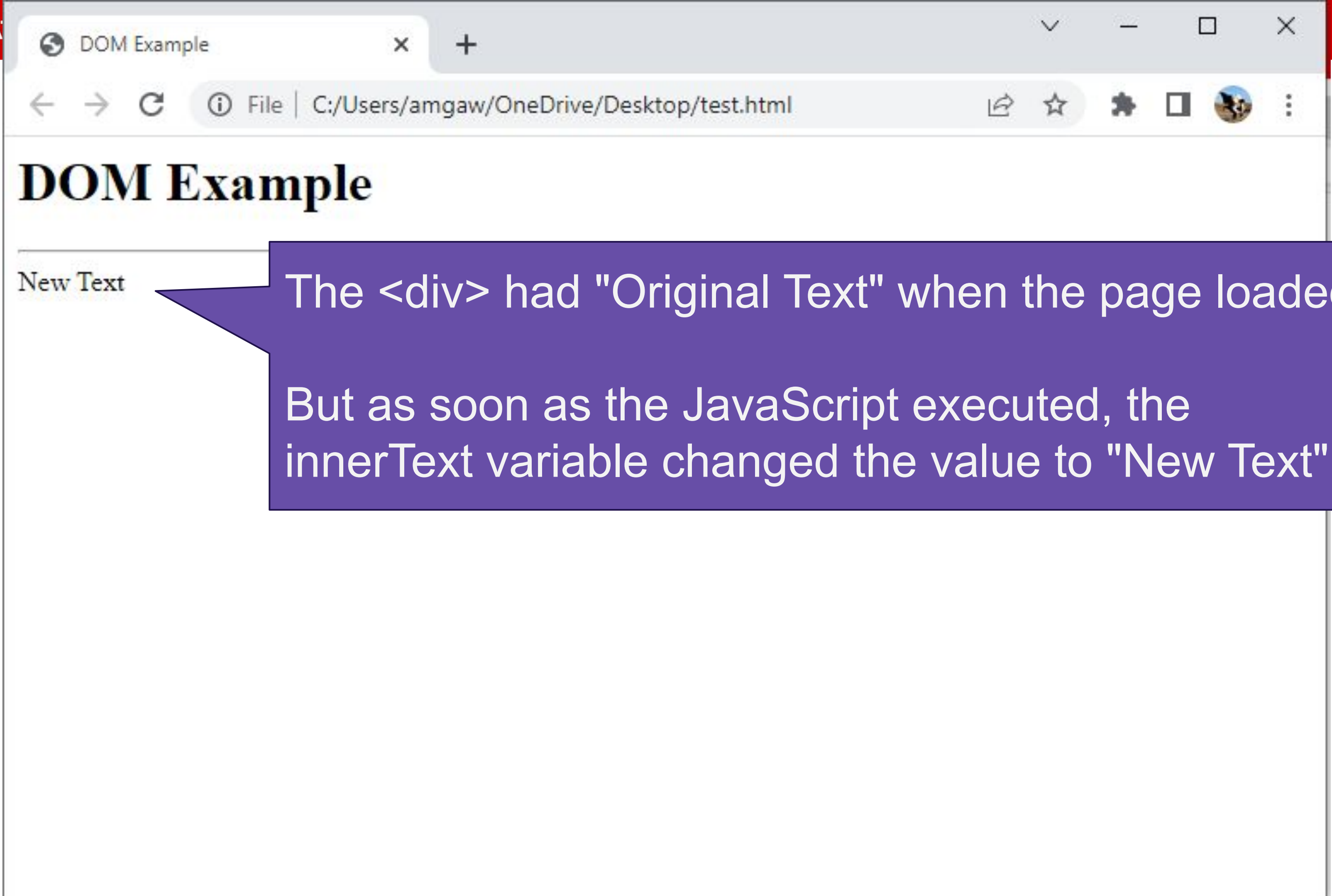
HTML is first rendered...

THEN JavaScript is executed



DOM Example

New Text



DOM Example

New Text

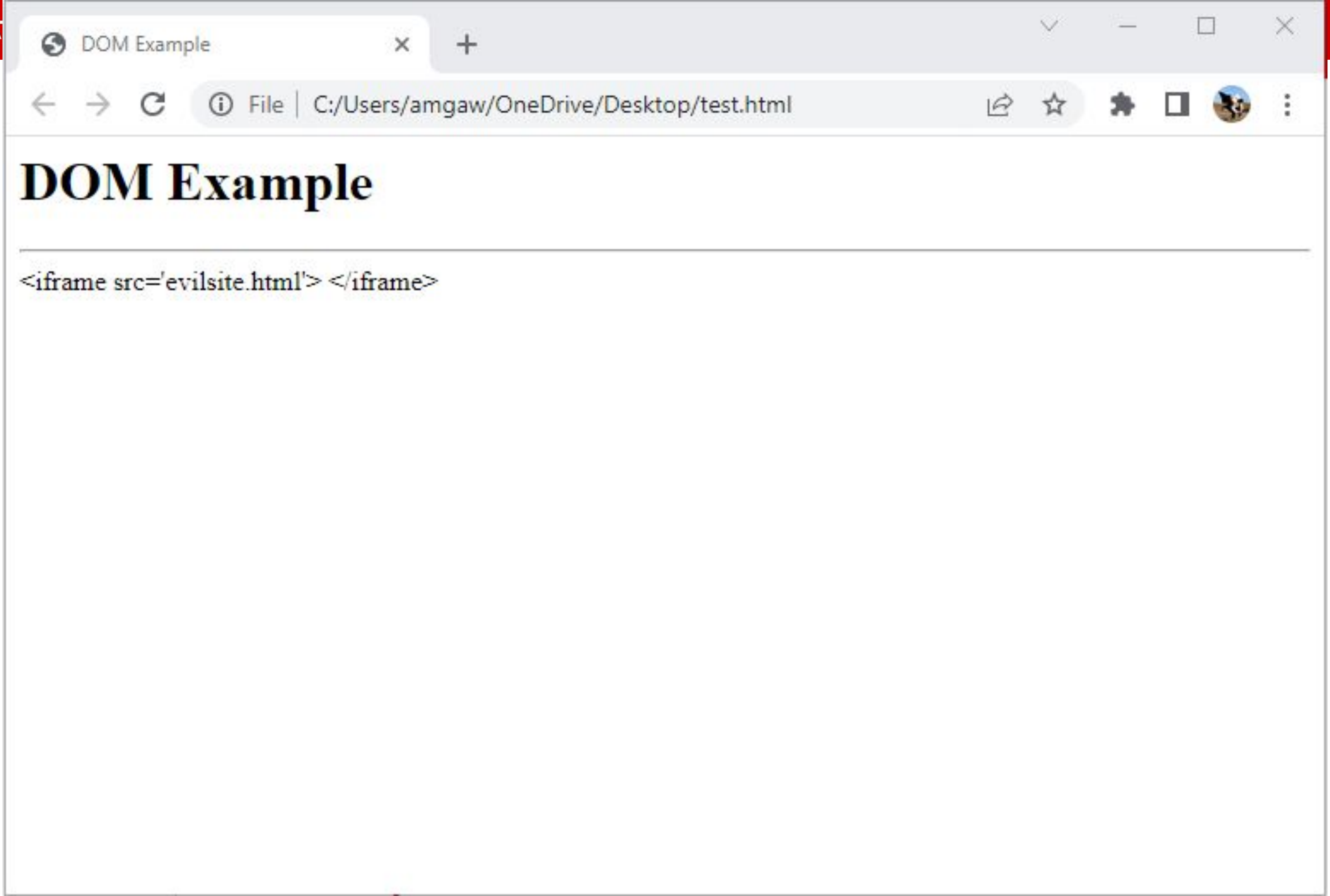
The <div> had "Original Text" when the page loaded.

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.innerText = "<iframe src='evilsite.html'> </iframe>";
 </script>
</html>
```

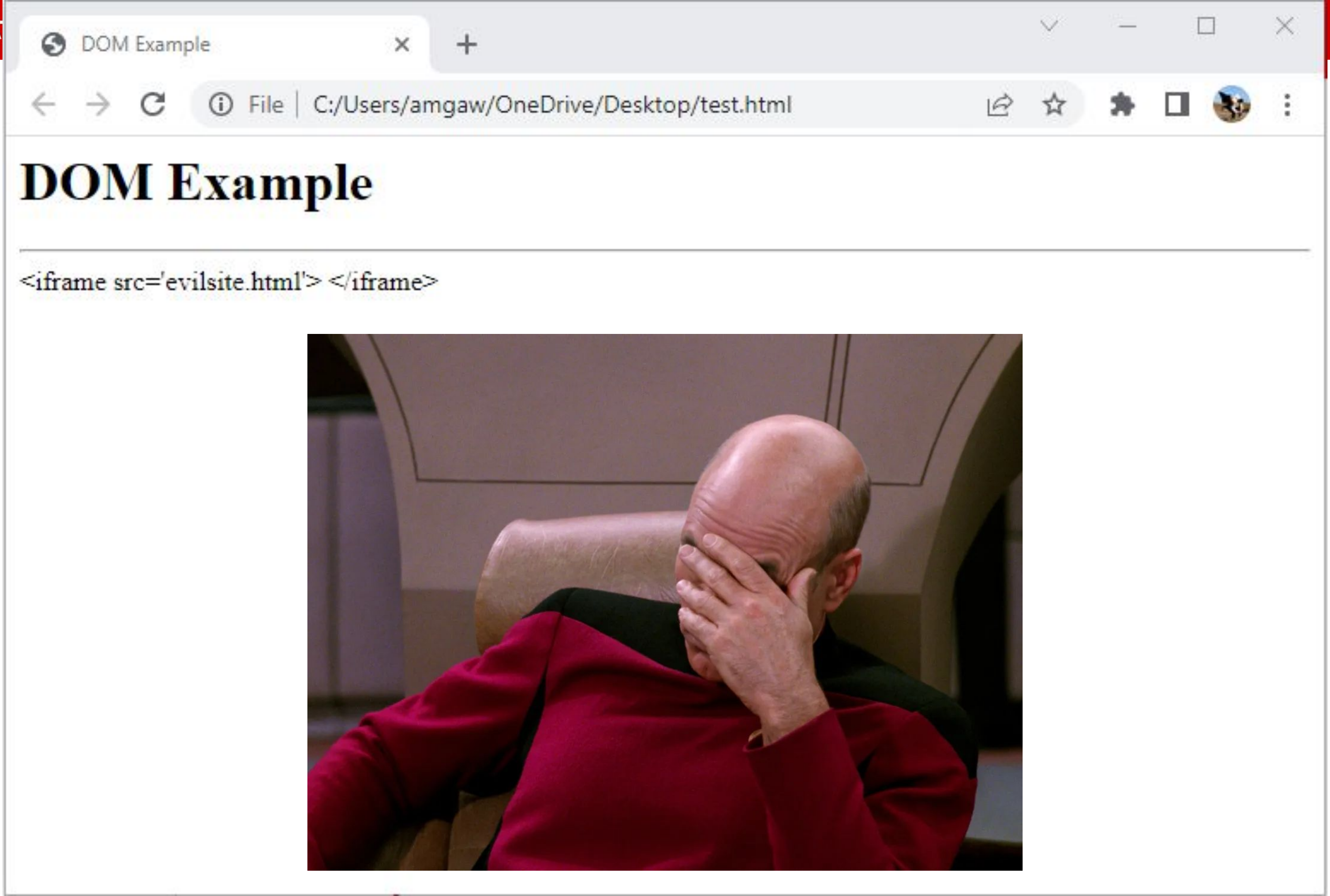
We can place **anything** for the browser to render



DOM Example

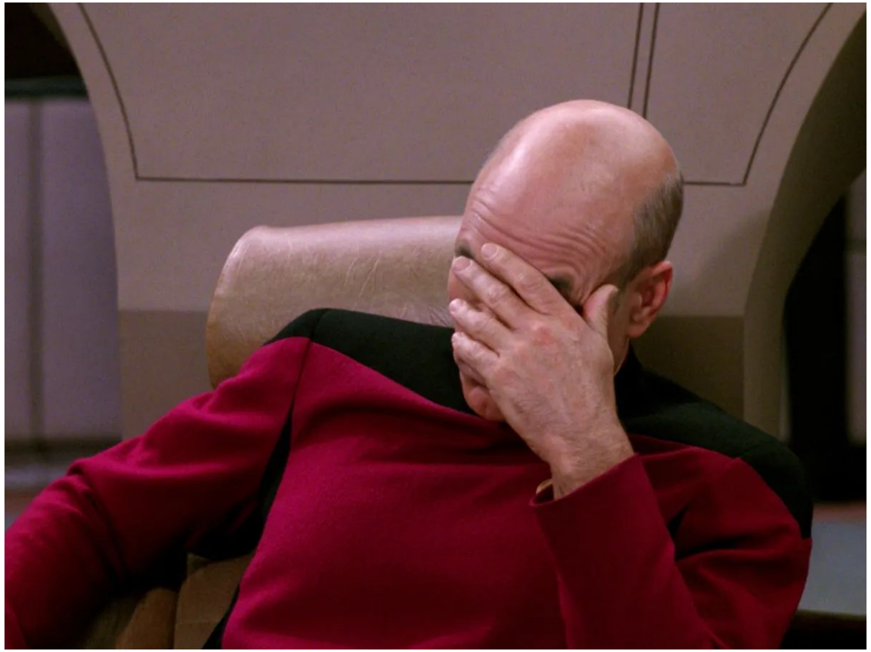
<iframe src='evilsite.html'> </iframe>

NC STA



DOM Example

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

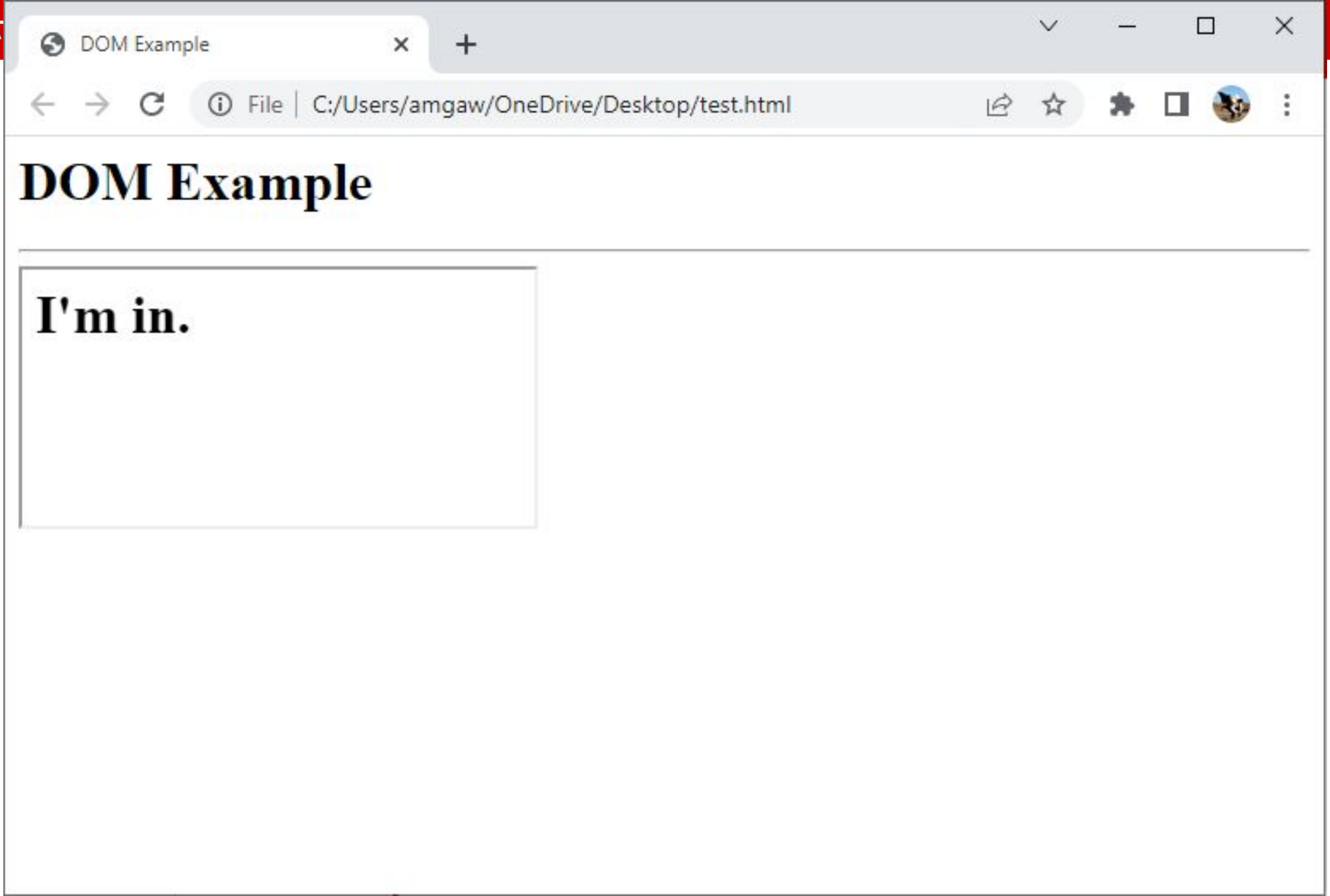


DOM Example

```
<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...

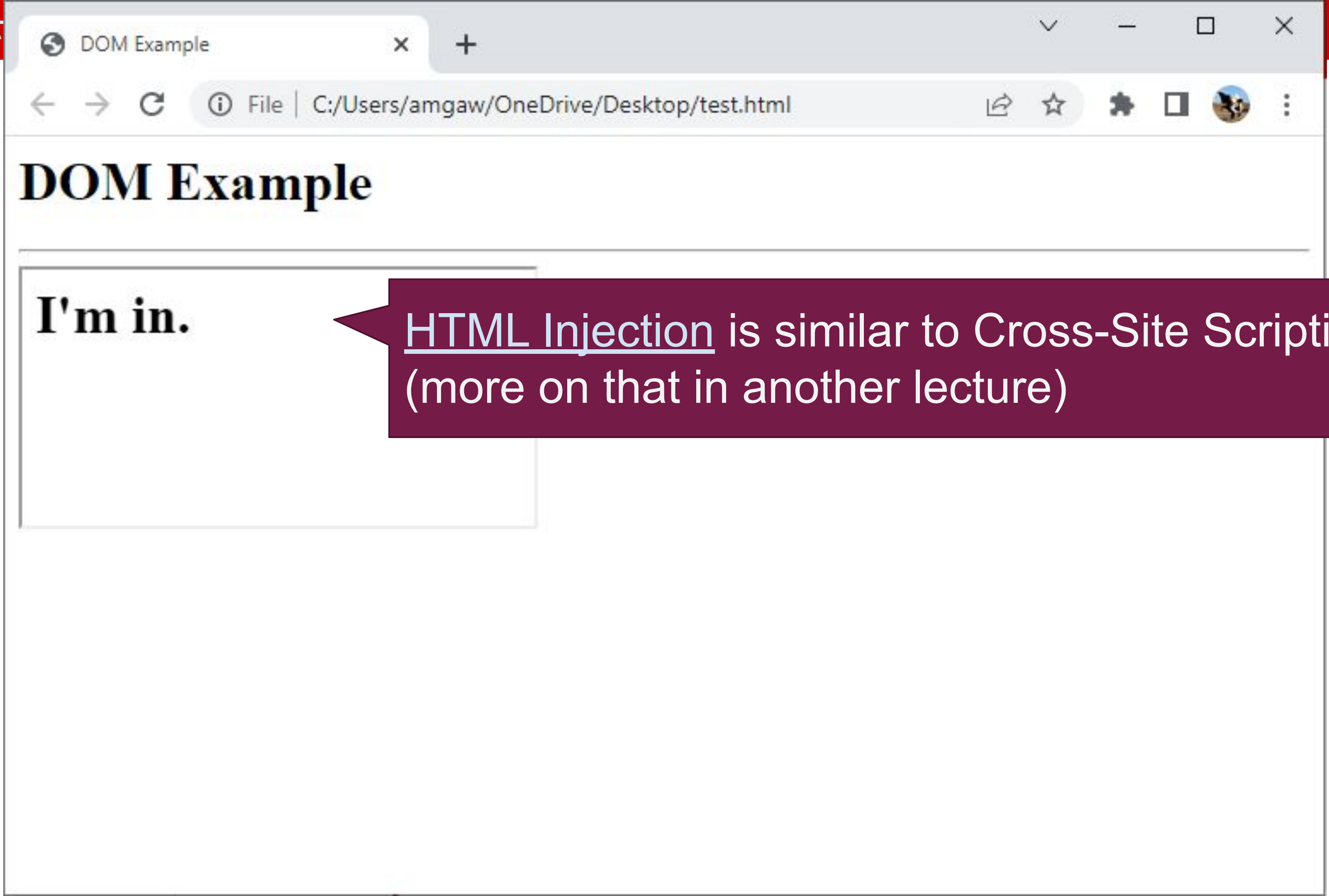
NC STA



DOM Example

I'm in.

NC STAT


DOM Example

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

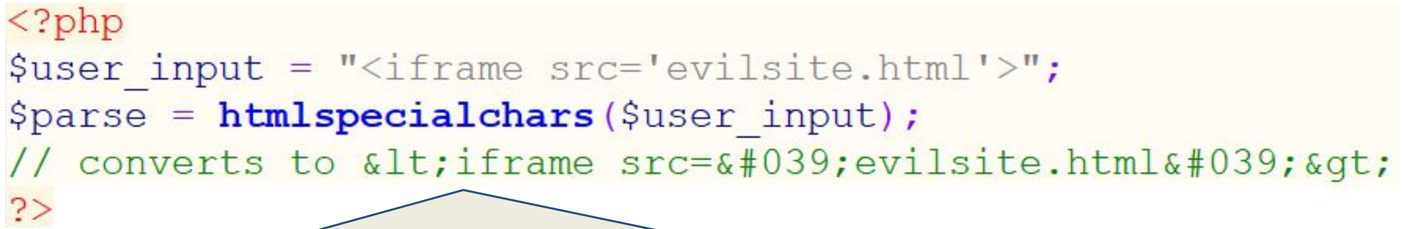
DOM Example

I'm in.

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

Preventing HTML Injection

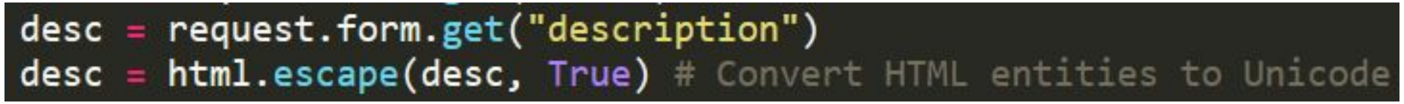
- **NEVER** let raw user input be rendered
- Escape HTML special characters to **`&equivalent;`**



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

PHP

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



```python
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 **`&nbsp;`** 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"
```

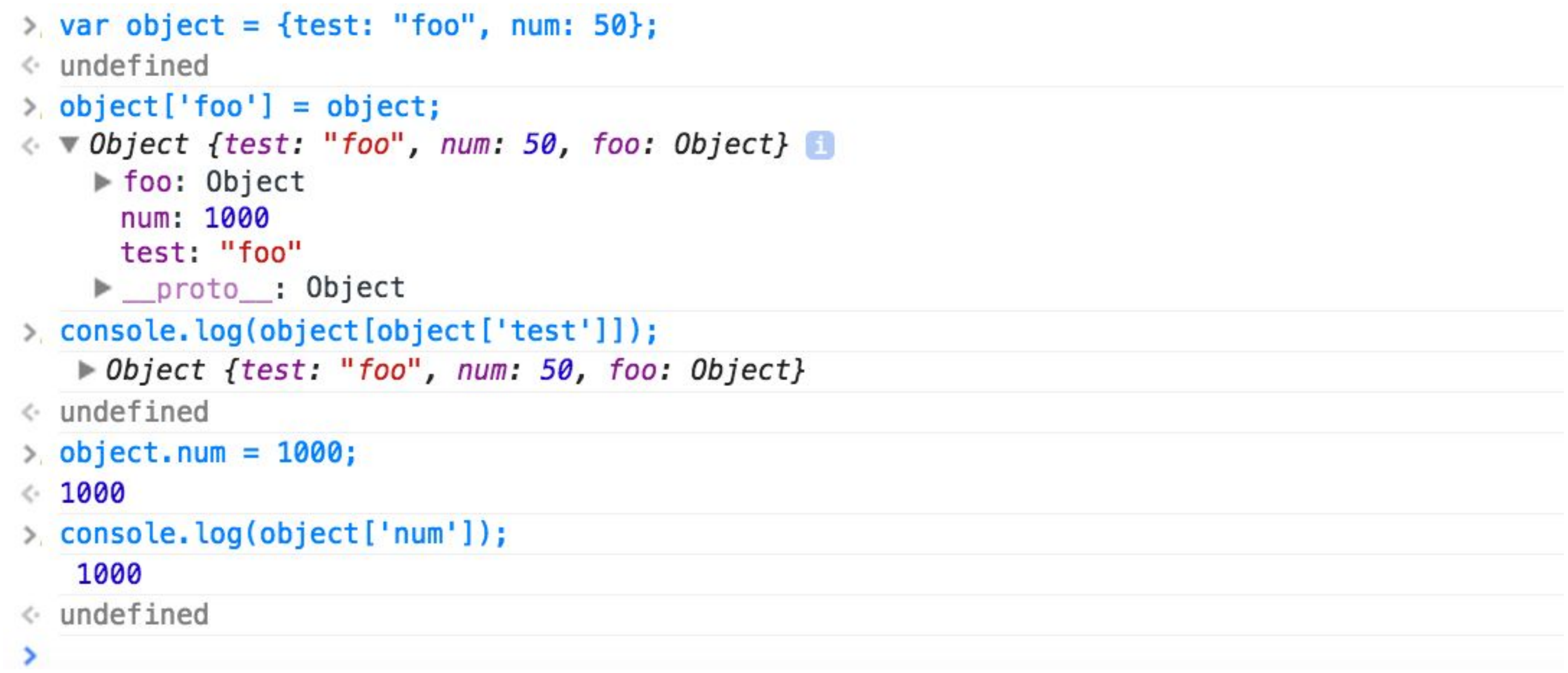
JavaScript vs. DOM and BOM

- 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

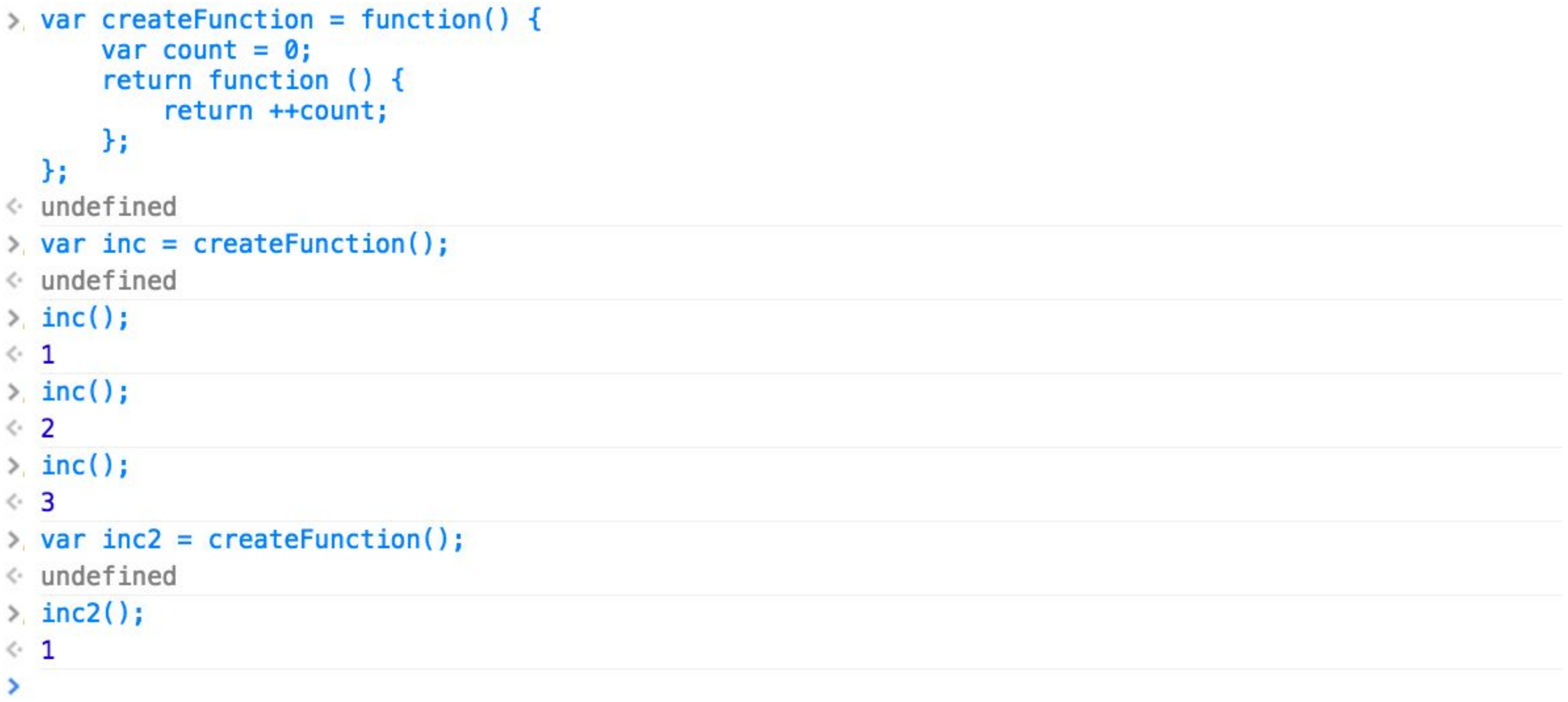
```
var object = {test: "foo", num: 50};
object['foo'] = object;
console.log(object[object['test']]);
object.num = 1000;
console.log(object['num']);
```



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

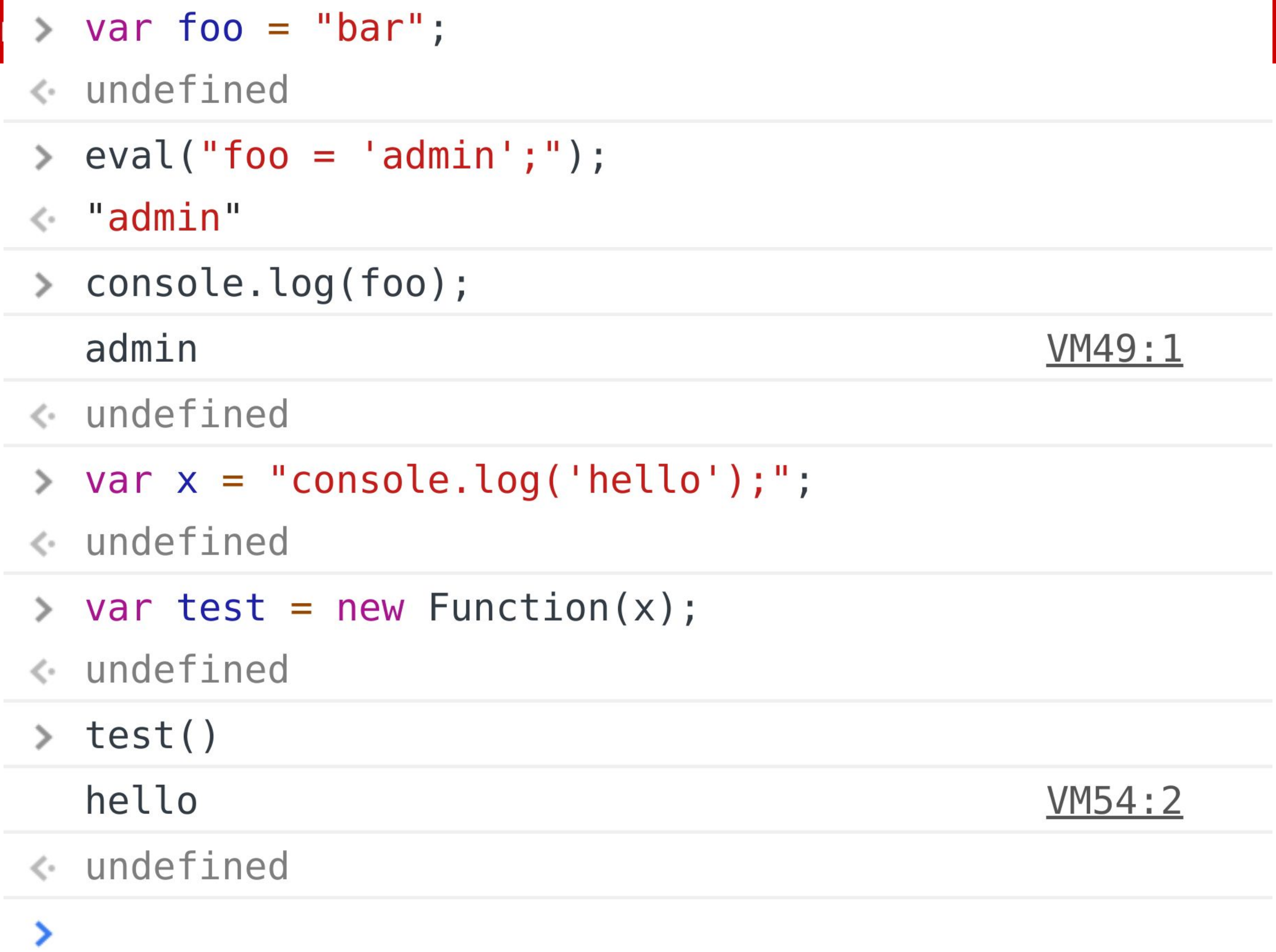


```
> var createFunction = function() {
      var count = 0;
      return function () {
          return ++count;
      };
  };
<· undefined
> var inc = createFunction();
<· undefined
> inc();
<· 1
> inc();
<· 2
> inc();
<· 3
> var inc2 = createFunction();
<· undefined
> inc2();
<· 1
>
```

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
>
```

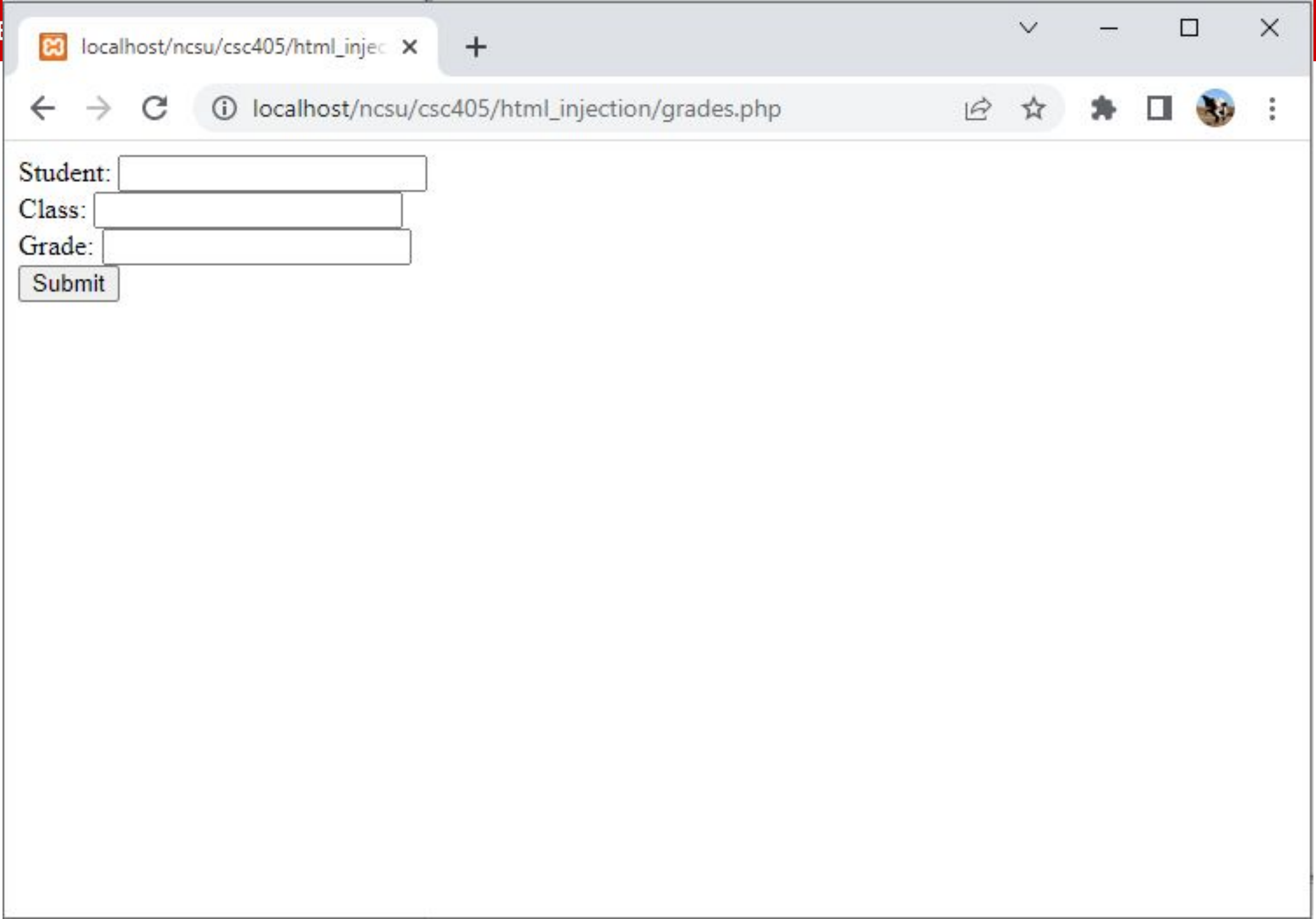
JavaScript Uses – Form Validation

- 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>
```

NC STATE



localhost/ncsu/csc405/html_injection/grades.php

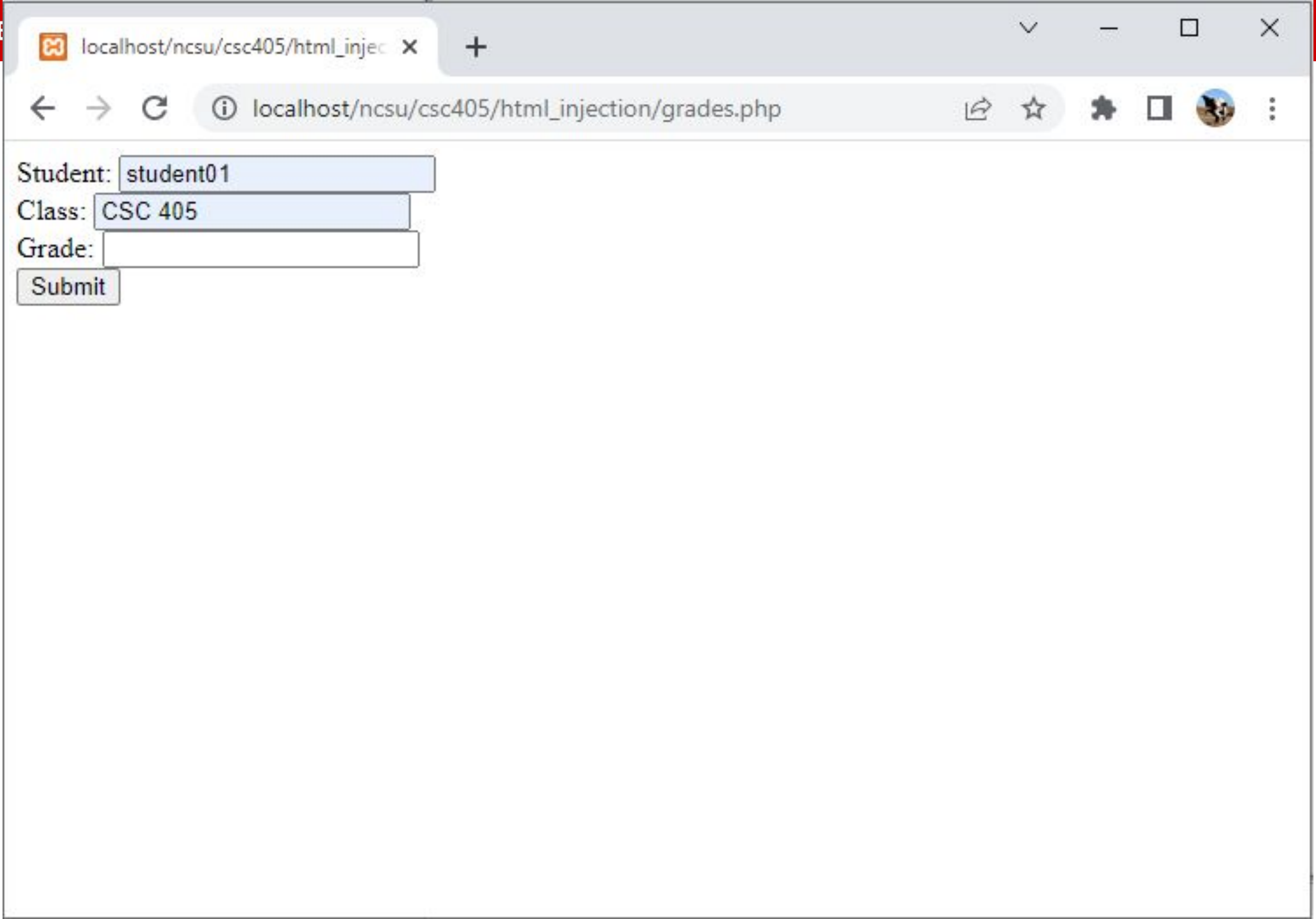
Student:

Class:

Grade:

Submit

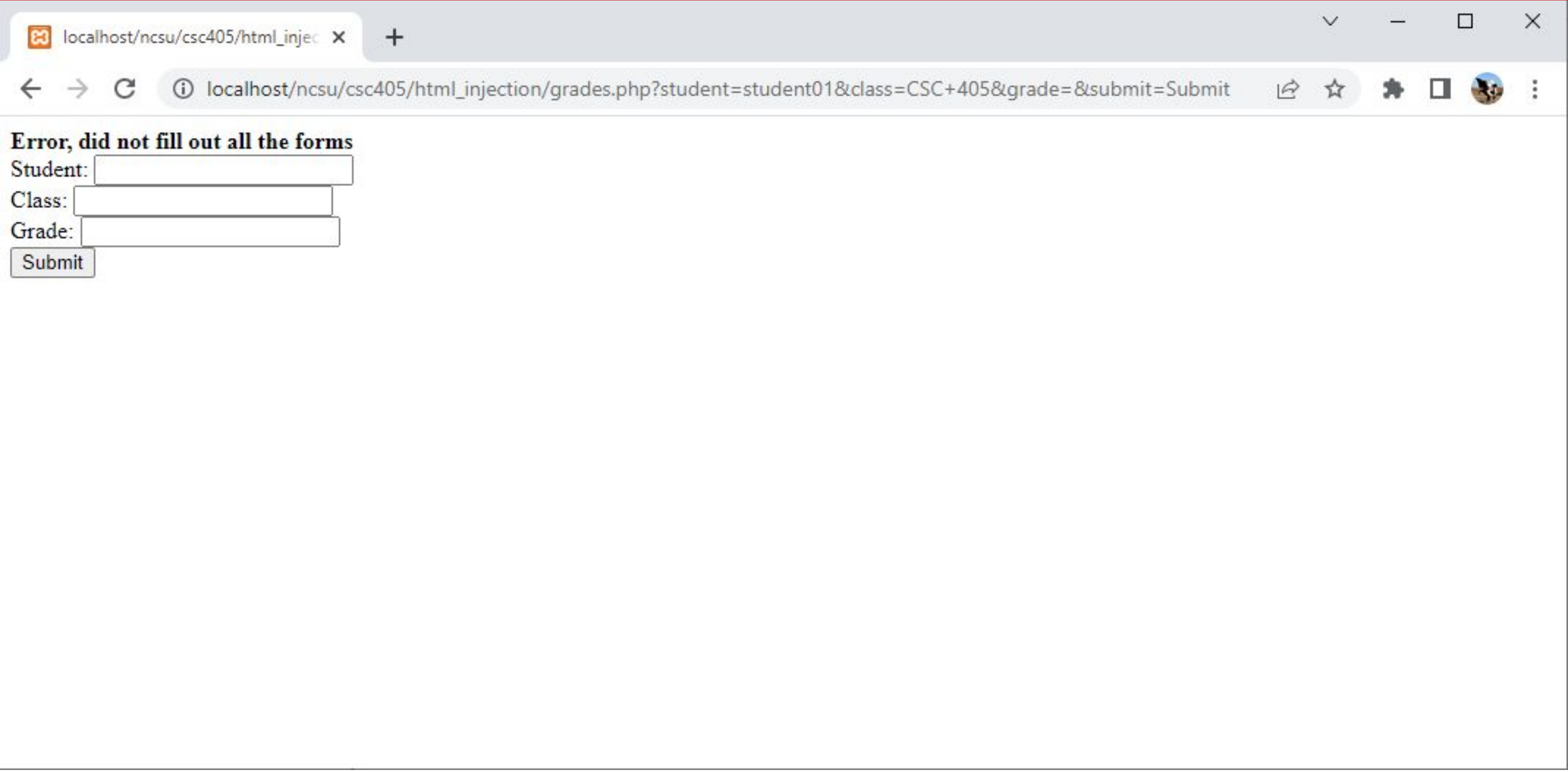
NC STATE


localhost/ncsu/csc405/html_injec
localhost/ncsu/csc405/html_injection/grades.php

Student:	student01
Class:	CSC 405
Grade:	

Submit

NC STATE UNIVERSITY



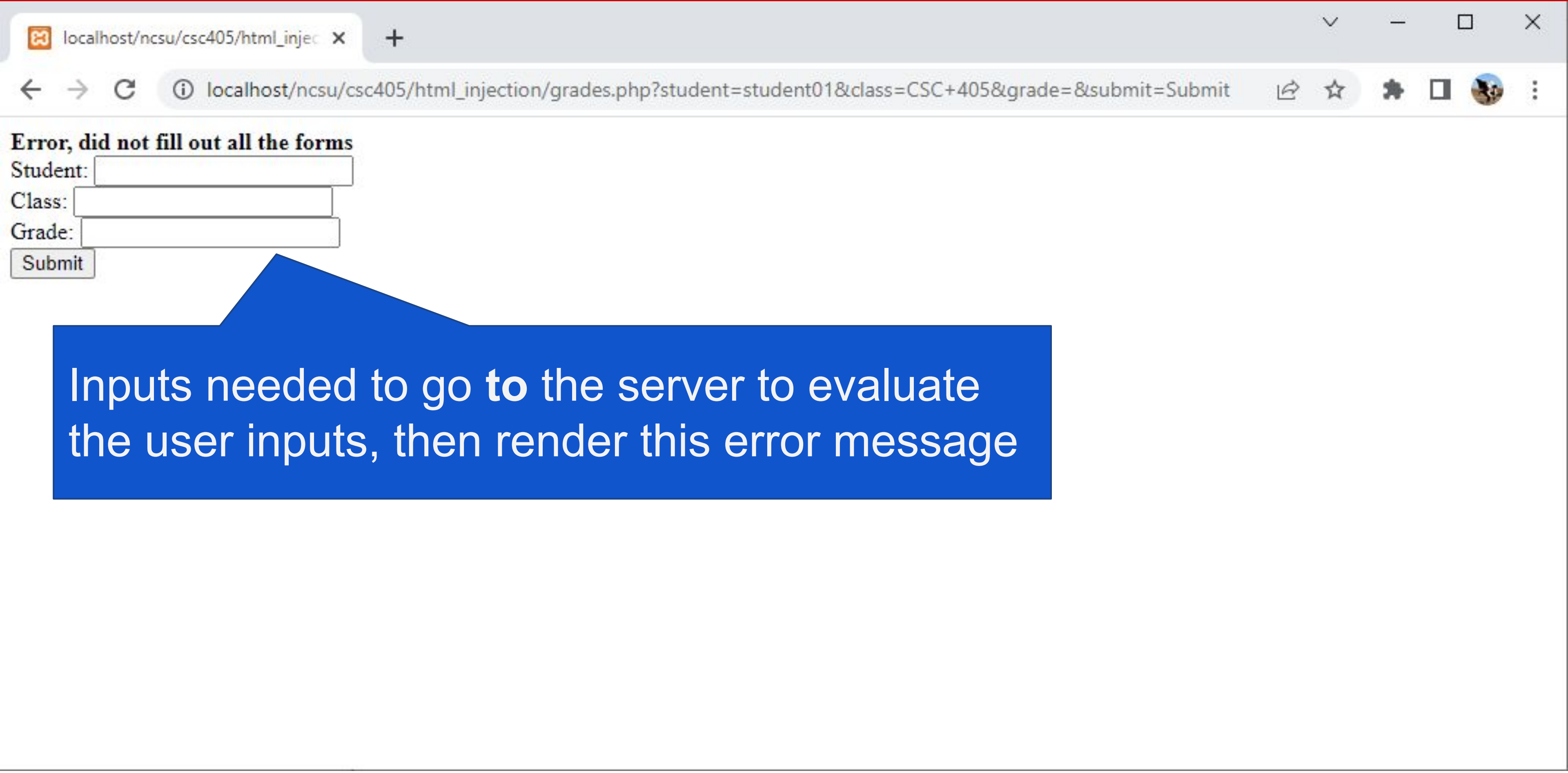
localhost/ncsu/csc405/html_injection/grades.php?student=student01&class=CSC+405&grade=&submit=Submit

Error, did not fill out all the forms

Student:

Class:

Grade:


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

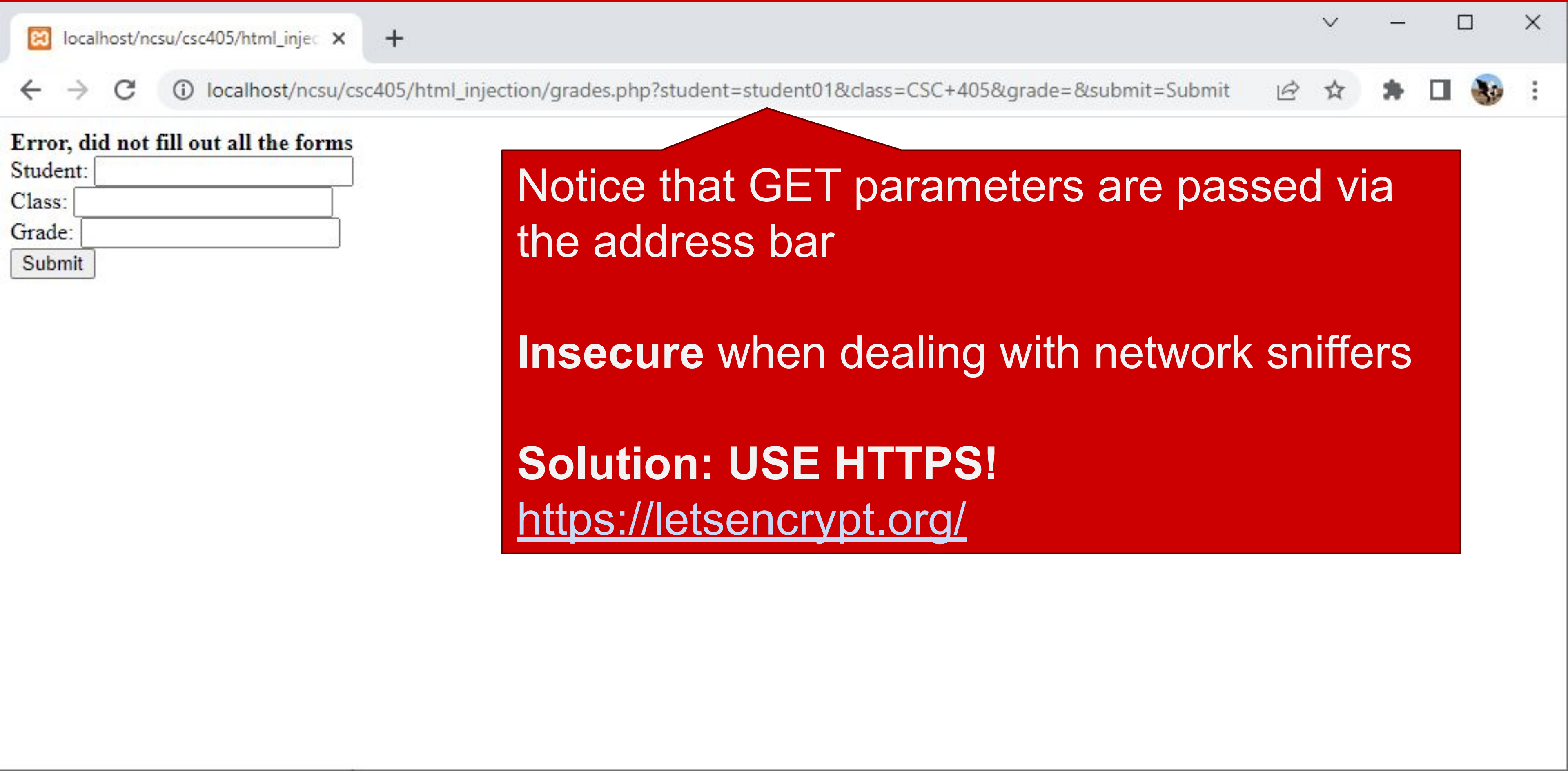
Error, did not fill out all the forms

Student:

Class:

Grade:

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



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:

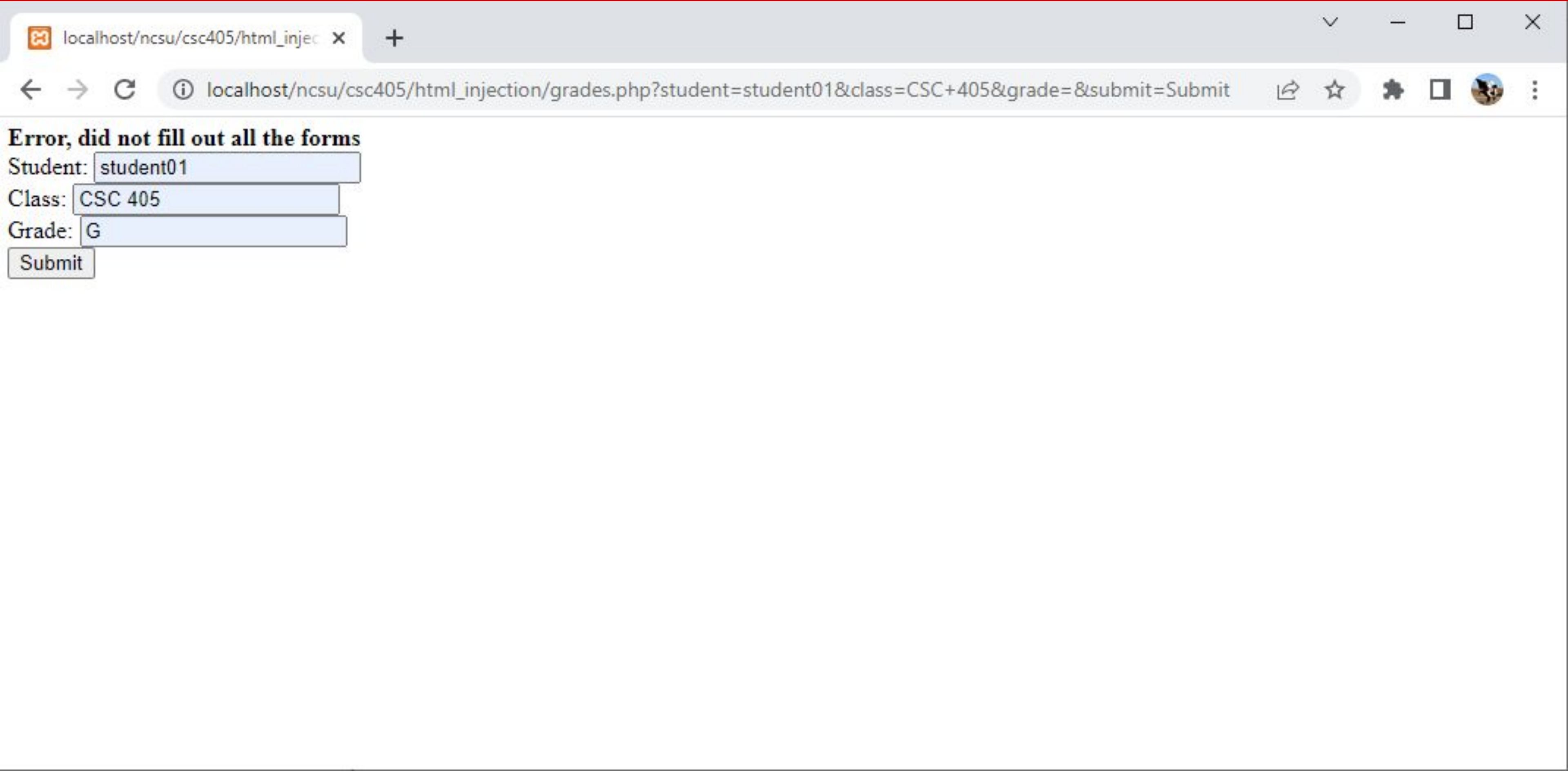
Grade:

Submit

Notice that GET parameters are passed via the address bar

Insecure when dealing with network sniffers

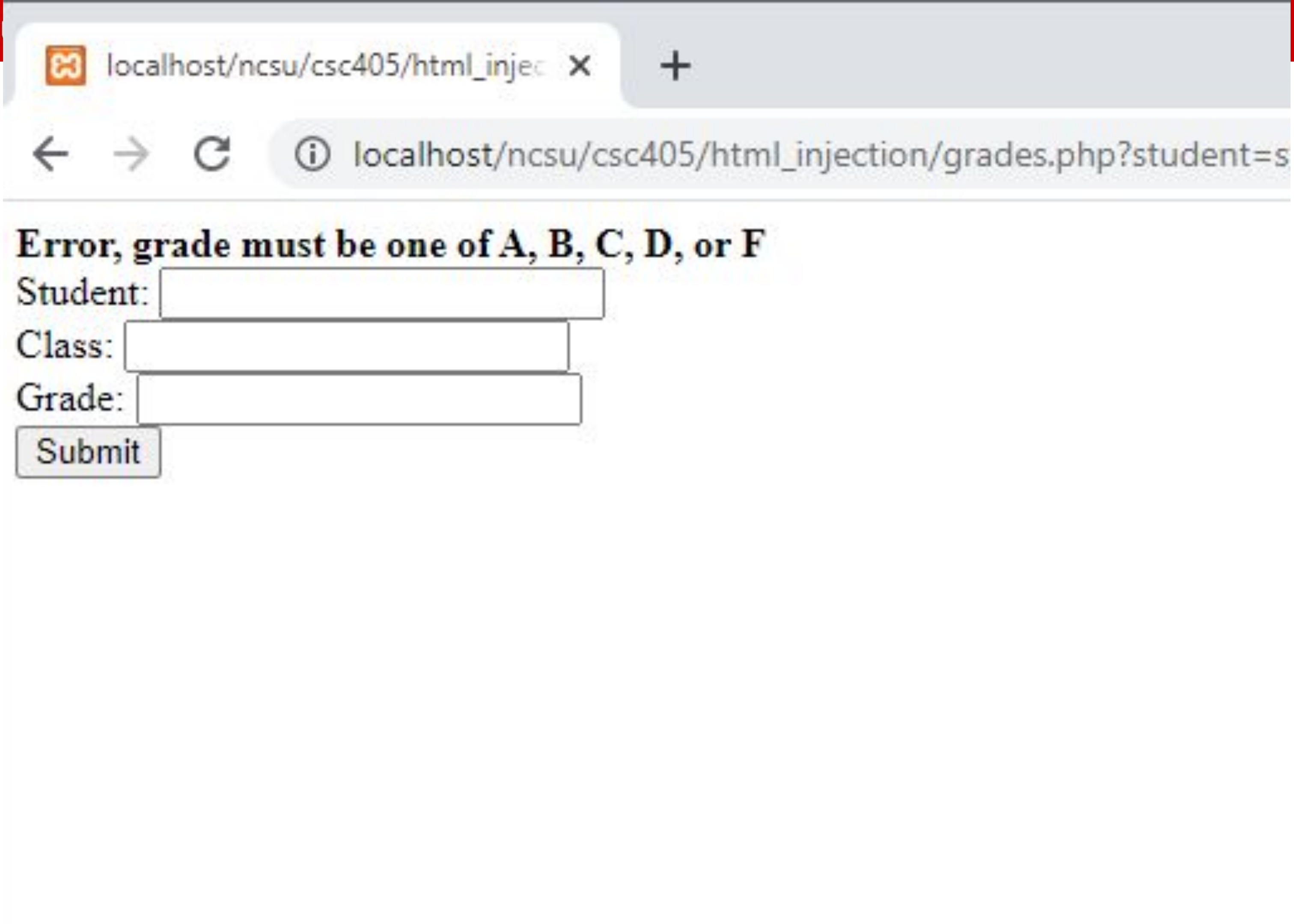
Solution: USE HTTPS!
https://letsencrypt.org/


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

Error, did not fill out all the forms

Student:	student01
Class:	CSC 405
Grade:	G

NC STATE


localhost/ncsu/csc405/html_injec
localhost/ncsu/csc405/html_injection/grades.php?student=s

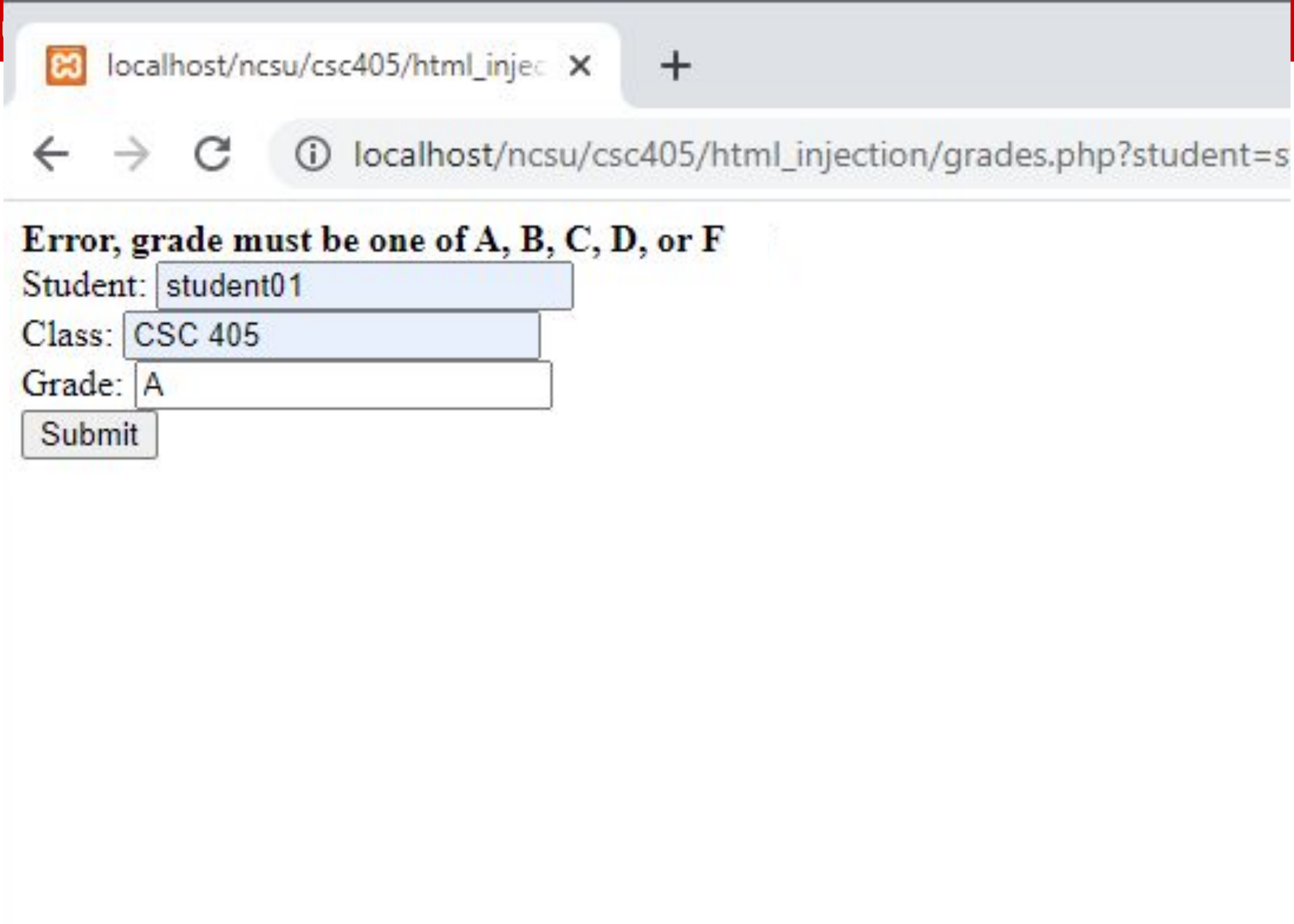
Error, grade must be one of A, B, C, D, or F

Student:

Class:

Grade:

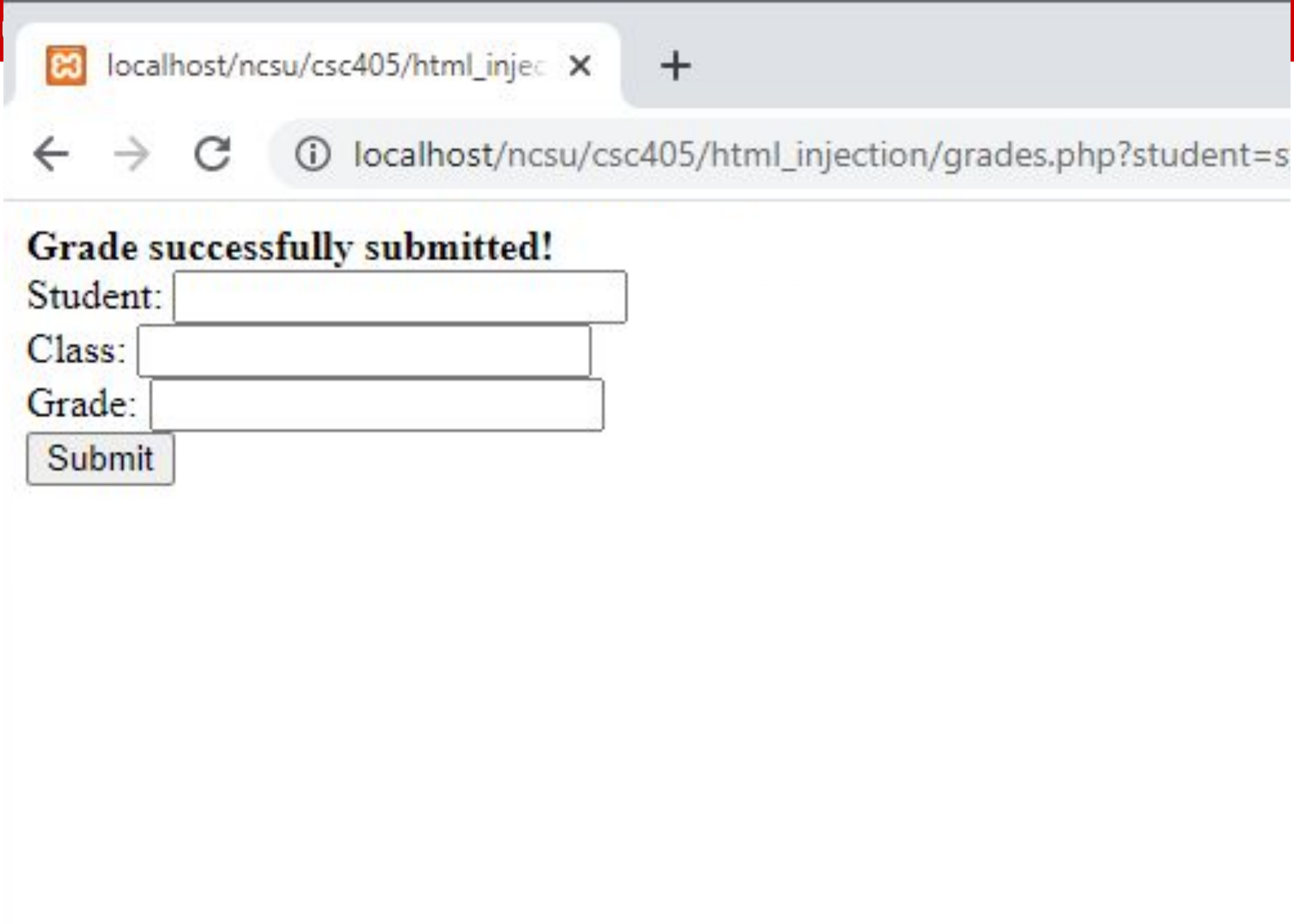
NC STATE


localhost/ncsu/csc405/html_injec
localhost/ncsu/csc405/html_injection/grades.php?student=s

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

Student: student01
Class: CSC 405
Grade: A

NC STATE


localhost/ncsu/csc405/html_inje

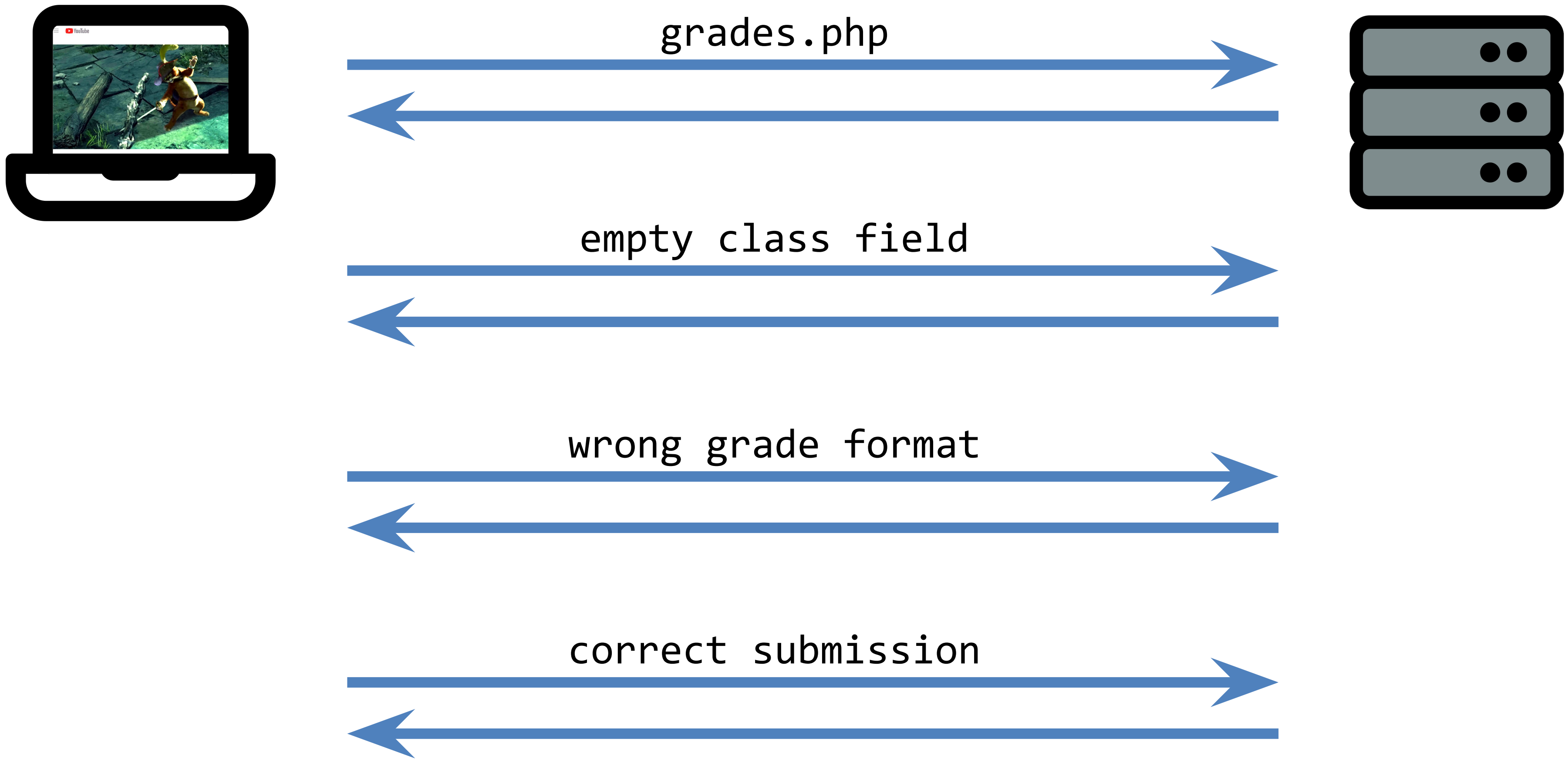
localhost/ncsu/csc405/html_injection/grades.php?student=s

Grade successfully submitted!

Student:

Class:

Grade:


grades.php
empty class field
wrong grade format
correct submission

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**

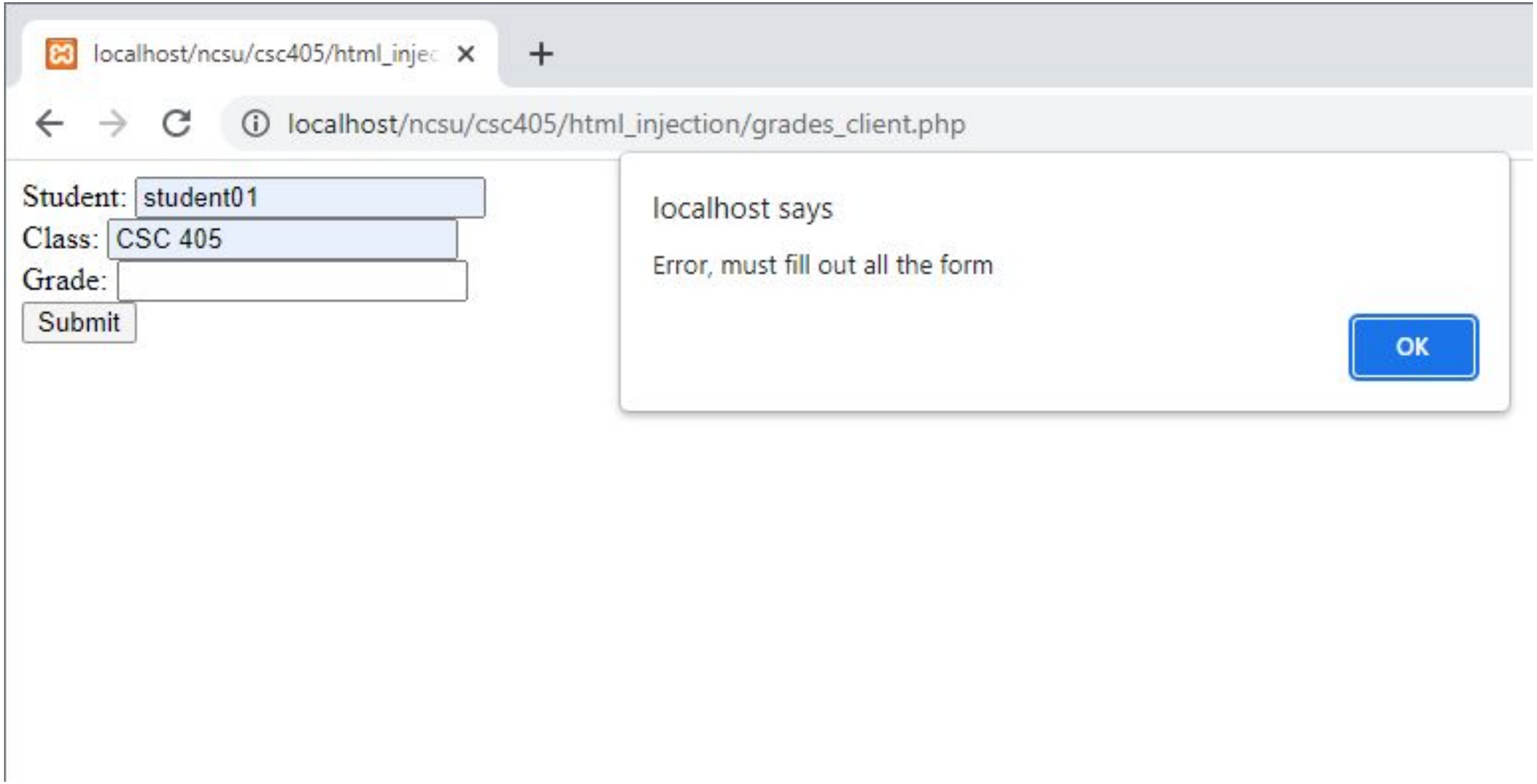
JavaScript Uses – Form Validation

```
<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

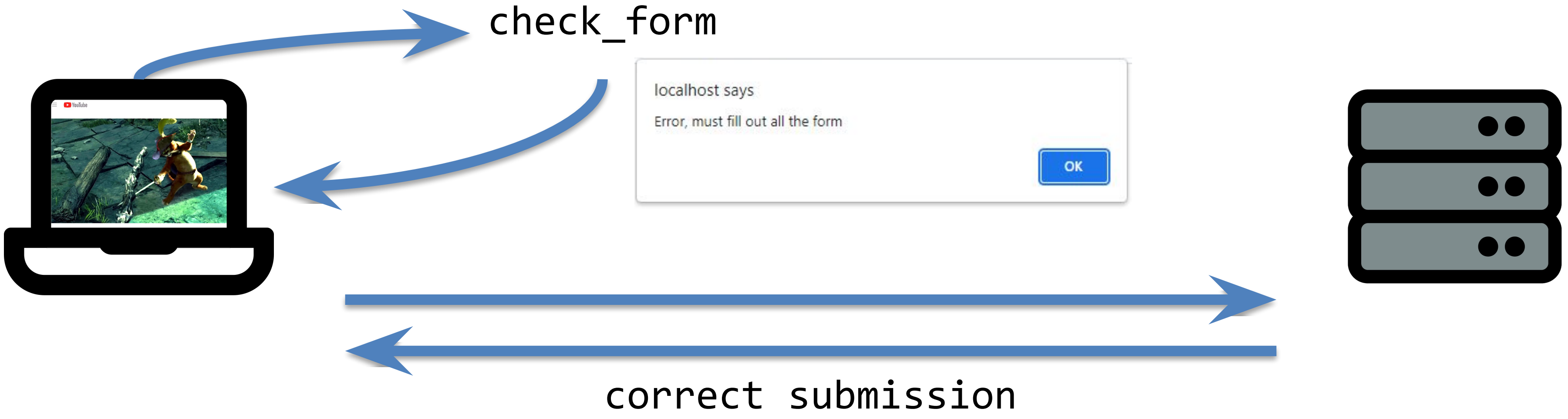

localhost/ncsu/csc405/html_injec
localhost/ncsu/csc405/html_injection/grades_client.php

Student: student01
Class: CSC 405
Grade:

localhost says

Error, must fill out all the form

grades_client.php



localhost says

Error, must fill out all the form

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

Source: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/AJAX_Security_Cheat_Sheet.md

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

Source: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/AJAX_Security_Cheat_Sheet.md

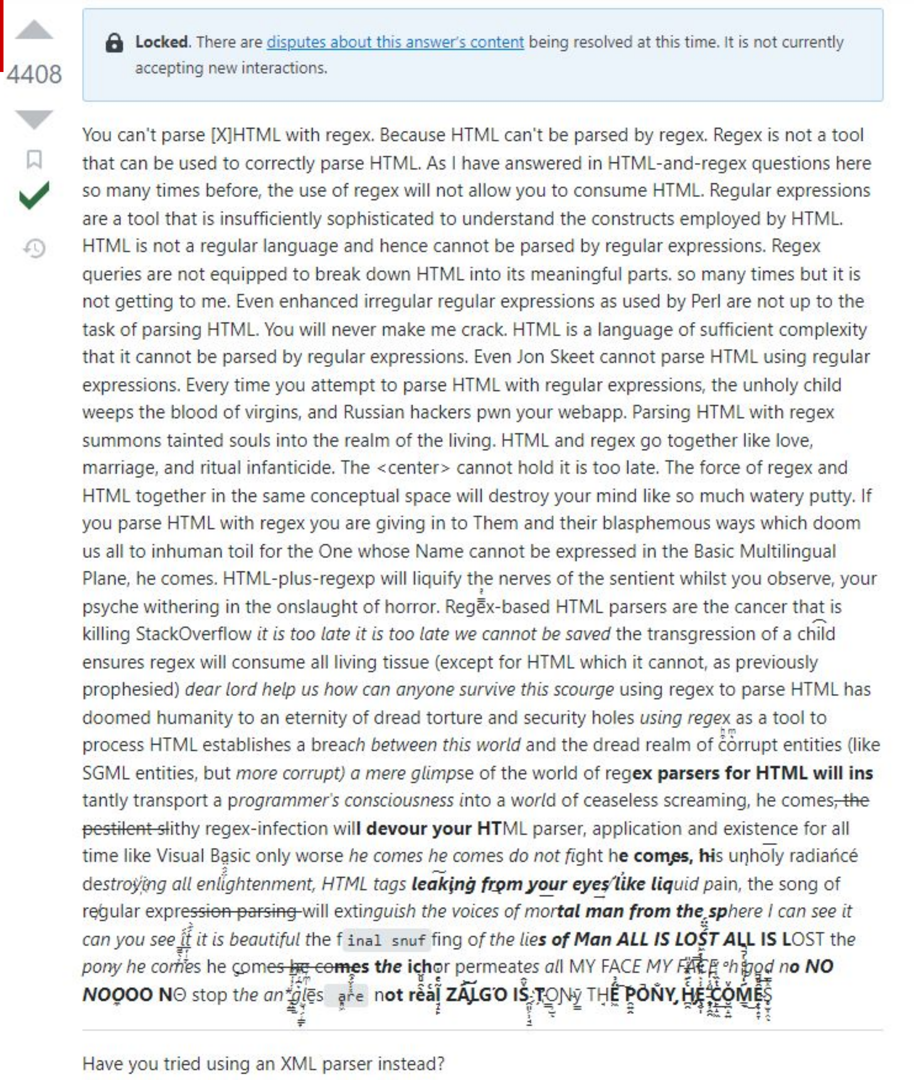
Client-Side Validation

- 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

Source: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/AJAX_Security_Cheat_Sheet.md

Security Zen
ZALGO COMES

https://stackoverflow.com/questions/1732348/regex-match-open-tags-except-xhtml-self-contained-tags



4408

Locked. There are disputes about this answer's content being resolved at this time. It is not currently accepting new interactions.

You can't parse [X]HTML with regex. Because HTML can't be parsed by regex. Regex is not a tool that can be used to correctly parse HTML. As I have answered in HTML-and-regex questions here so many times before, the use of regex will not allow you to consume HTML. Regular expressions are a tool that is insufficiently sophisticated to understand the constructs employed by HTML. HTML is not a regular language and hence cannot be parsed by regular expressions. Regex queries are not equipped to break down HTML into its meaningful parts. so many times but it is not getting to me. Even enhanced irregular regular expressions as used by Perl are not up to the task of parsing HTML. You will never make me crack. HTML is a language of sufficient complexity that it cannot be parsed by regular expressions. Even Jon Skeet cannot parse HTML using regular expressions. Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. HTML and regex go together like love, marriage, and ritual infanticide. The <center> cannot hold it is too late. The force of regex and HTML together in the same conceptual space will destroy your mind like so much watery putty. If you parse HTML with regex you are giving in to Them and their blasphemous ways which doom us all to inhuman toil for the One whose Name cannot be expressed in the Basic Multilingual Plane, he comes. HTML-plus-regexp will liquify the nerves of the sentient whilst you observe, your psyche withering in the onslaught of horror. Regex-based HTML parsers are the cancer that is killing StackOverflow *it is too late it is too late we cannot be saved* the transgression of a child ensures regex will consume all living tissue (except for HTML which it cannot, as previously prophesied) *dear lord help us how can anyone survive this scourge* using regex to parse HTML has doomed humanity to an eternity of dread torture and security holes *using regex* as a tool to process HTML establishes a brea*ch between this world* and the dread realm of corrupt entities (like SGML entities, but *more corrupt) a mere glimps*e of the world of reg**ex parsers for HTML will ins** tantly transport a p*rogrammer's consciousness i*nto a world of ceaseless screaming, he comes~~, the pestilent sl~~ithy regex-infection wil**l devour your HT**ML parser, application and existence for all time like Visual Basic only worse *he comes he comes do not fi*ght h**e comes, hi**s unholy radiance destro*ying all enlightenment, HTML tags* ***leaking from your eyes like liq****uid* pain, the song of regular expre~~ssion parsing~~ will exti*nguish the voices of mor****tal man from the sp****here I can see it can you see it it is beautiful* the f`inal snuf`fing *of the lie****s of Man ALL IS LOST ALL IS L**OST the *pony he comes* he com~~es he co~~**mes the ich**or permeat*es al*l MY FACE *MY FACE ᵒh god n**o NO NOOOO N**Θ stop *the an**gles `are` n**ot real ZALGO IS TO**N̷Y TH**E PONY HE COMES**

Have you tried using an XML parser instead?