Why HTML Forms Still Rule the Web

```
<form
  action="/compose/tickets/purchase"
  method="post">
```

About Me

- Developer Advocate at Pomerium
- From Montreal, Quebec, Canada
- I'm @nickytonline everywhere
- All the places you can find are on nickyt.online
- Not a big fan of spiders

Feedback Form

nickyt.online

What We’ll Cover

What We'll Cover

- Hello Internet
- Hello Forms
- AJAX
- CSS for Forms
- Constraint Validation for Forms
- Demo Time
- Forms & Frameworks

Hello Internet

nickyt.online

The first computer I paid for with my own cash

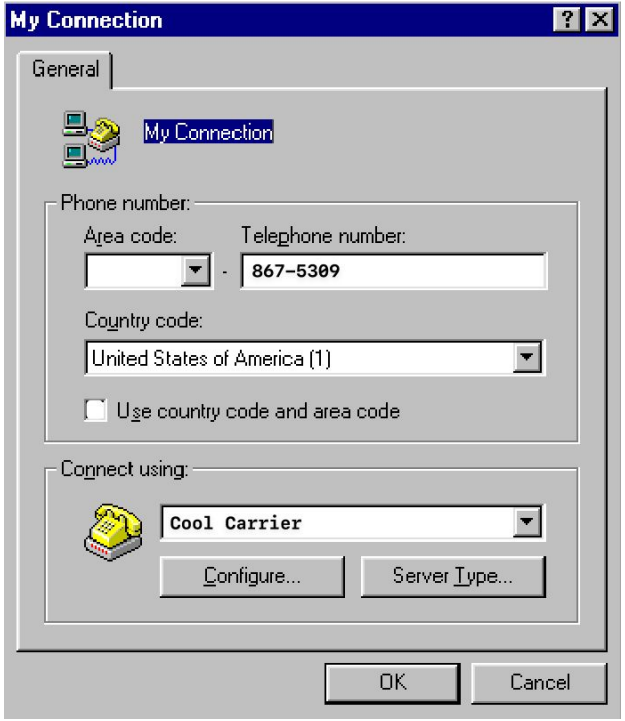
- i486DX-33 processor
- 14.4K baud modem
- Windows 3.1
- chonky Logitech speakers
- 3.5-inch disk drive
- CD-ROM drive

Cost me 3600$ CAD in 1993

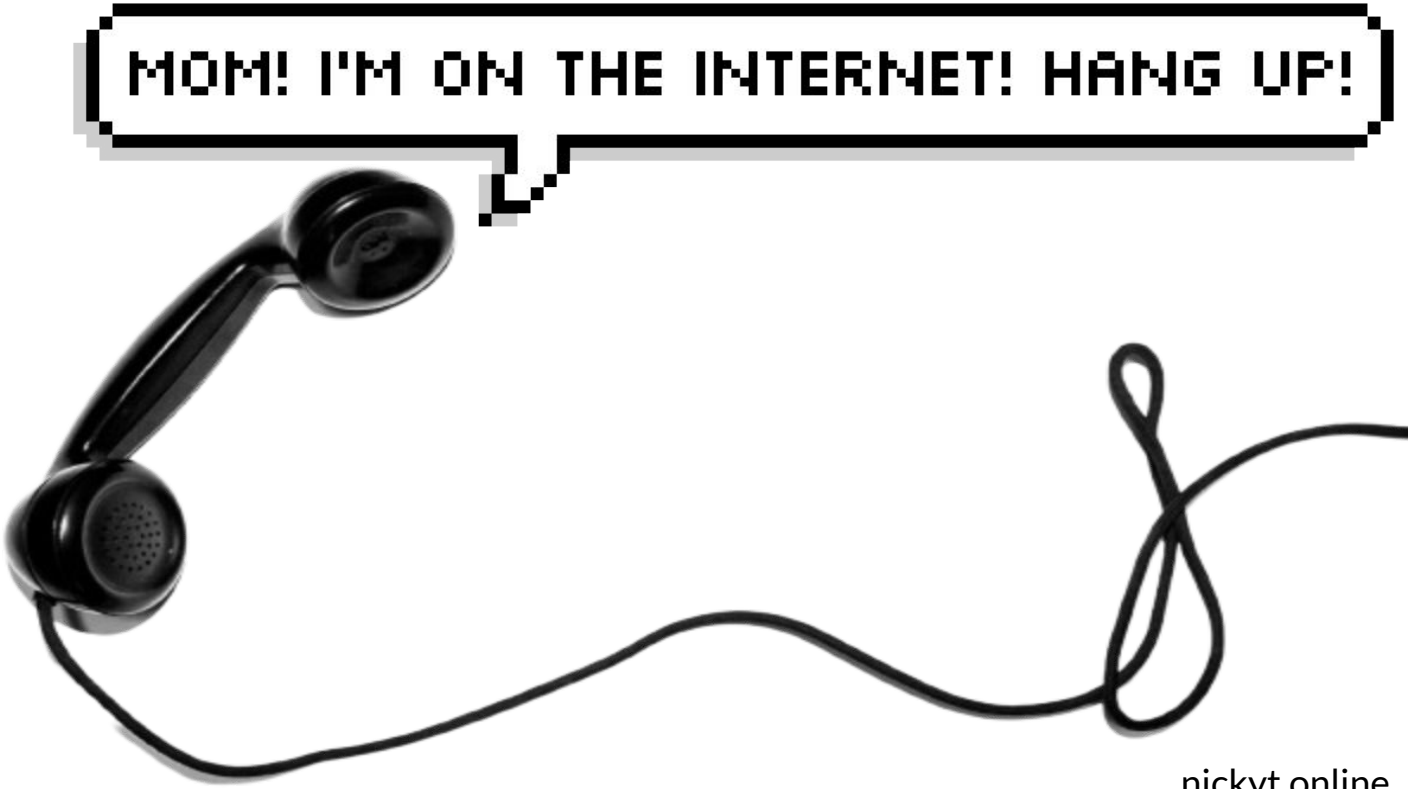
I Got Online

My journey to the Internet

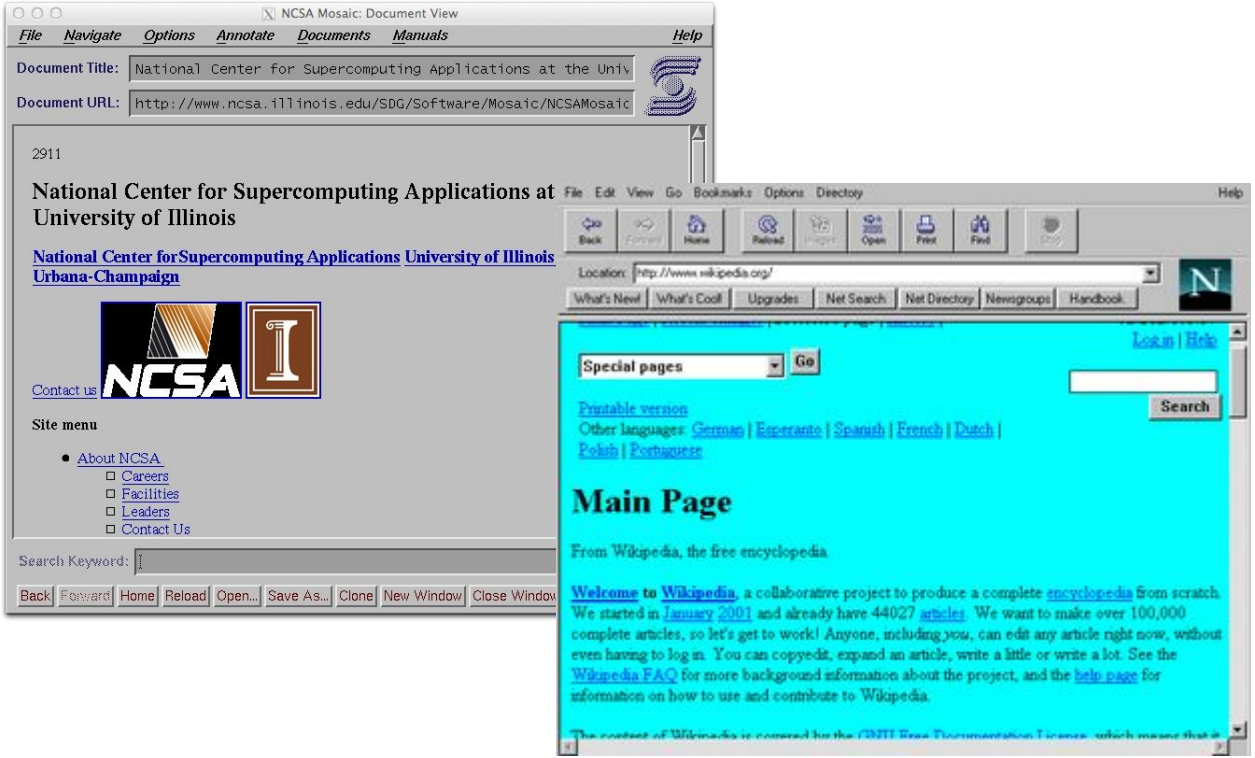
- Bulletin Board System (BBS)
- America Online and Compuserve
- My Local Internet Service Provider



Sometimes you got interrupted 😅



Hello web browser



Google

geocities

All | Images | News | Videos | Shopping | Maps | More | Tools

GeoCities was a once very popular web hosting service founded in 1994 and purchased by Yahoo in 1999.

Archiveteam
https://wiki.archiveteam.org › index.php › GeoCities

GeoCities - Archiveteam

Wikipedia
https://en.wikipedia.org › wiki › GeoCities

GeoCities

GeoCities, later Yahoo! GeoCities, was a web hosting service that allowed users to create and publish websites for free and to browse user-created websites ...

nickyt.online

Hello Forms

Hello Forms

Guestbooks were my first encounter with forms

Hello Forms

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $name = isset($_POST['name']) ? $_POST['name'] : '';
    $email = isset($_POST['email']) ? $_POST['email'] : '';

    echo "<p>Submitted Name: $name</p><p>Submitted Email: $email</p>";
}
?>
<form method="post" action="<?php echo $PHP_SELF; ?>">
    Name: <input type="text" name="name"><br><br>
    Email: <input type="text" name="email"><br><br>
    <input type="submit" value="Submit">
</form>
```

Hello Forms

```
<%@ Page Language="JScript" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ASP.NET WebForm with Server Side JScript</title>
</head>
<body>
    <form id="form1" runat="server"
        <div>
            <h1>Welcome to ASP.NET
            <asp:TextBox ID="txtNam
            <asp:Button ID="btnGree
            <br /><br />
            <asp:Label ID="lblGreet
        </div>
    </form>
</body>
</html>
```

```
<script runat="server" language="JScript">
function btnGreet_Click(sender, e) {
    var name = txtName.Text;
    lblGreeting.Text = "Hello, " + name + "!";
}

function Page_Load(sender, e) {
    if (!IsPostBack) {
        lblGreeting.Text = "Enter your name and click 'Greet'.";
    }
}
</script>
```

Hello Forms

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="SomeForm.aspx.cs" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>
  </head>
  <body>
    <form id="form1" runat="server">
      <h1>Welcome to Web Forms!</h1>
      <!— more awesome markup —>
    </form>
  </body>
</html>
```

Hello Forms

Other server-side frameworks came along eventually like Rails, Django etc. but before all that something happened in 2005 that changed how we built digital experiences.

Hello AJAX

In a 2005 paper,[3] Garrett coined the term Ajax to describe the asynchronous technology behind emerging services like Google Maps and Google Suggest, as well as the resulting user experience which made it possible to browse without interruption by eliminating the reloading of the whole page.[4]

– Wikipedia, https://en.wikipedia.org/wiki/Jesse_James_Garrett

Hello AJAX

```
function getXMLHttpRequest() {
  if (document.all) {
    // Internet Explorer
    return new ActiveXObject("Microsoft.XMLHTTP");
  } else {
    // Netscape and other browsers
    return new XMLHttpRequest();
  }
}
```


Relax at the Spa
AJAX
AJAX

Hello AJAX

jQuery
write less, do more.

Download | API Documentation | Blog | Plugins | Browser Support

- Ajax
  - Global Ajax Event Handlers
  - Helper Functions
  - Low-Level Interface
  - Shorthand Methods
- Attributes
- Callbacks Object
- Core

jQuery.ajax()

Categories: Ajax > Low-Level Interface

jQuery.ajax(url [, settings])

Description: *Perform an a*

jQuery.ajax(url [, s

Hello AJAX

We eventually moved away for the most part from XML to JavaScript Object Notation (JSON), although I'm sure there are still a lot of legacy systems using XML. 😅

CSS for Forms and Form Elements

CSS for Forms and Form Elements

- :focus, when an element receives focus
- :focus-within, when an element or any of its descendants receives focus
- :focus-visible, applies when the focus is visible, e.g. keyboard navigation

CSS for Forms and Form Elements

- :valid, applies to form elements as well as forms that are in a valid state
- :invalid, applies to form elements as well as forms that are in an invalid state

CSS for Forms and Form Elements

- :user-valid, applies to form elements that are valid after the user has interacted with it
- :user-invalid, applies to form elements that are invalid after the user has interacted with it

CSS for Forms and Form Elements

- :required
- :optional
- :in-range
- :out-of-range
- :read-only
- :read-write

CSS for Forms and Form Elements

- :disabled
- :enabled
- :checked
- :indeterminate
- :default
- :placeholder-shown
- :empty

Constraint Validation for Forms

Constraint Validation for Forms

When HTML5 landed, it gave us some new types for the good old input HTML element.

For example:

- `<input type="email" />`
- `<input type="url" />`
- `<input type="number" />`
- etc.

Constraint Validation for Forms

Along with these new input types came some new attributes as well:

- pattern attribute
- min attribute
- max attribute
- required attribute
- and several other validation attributes

Constraint Validation for Forms

Constraint Validation API:

checkValidity()

Reports the validity of an input based element. Returns true if the field is valid, false otherwise. If used on a form element, it will report the validity of all fields.

Constraint Validation for Forms

reportValidity()

Reports the validity of an input based element. Returns true if the field is valid, false otherwise. On top that, if the element is invalid it displays the issue to the user in the browser so long as the event isn't canceled. If used on a form element, it will report the validity of all fields and fire an invalid event on each invalid element.

Constraint Validation for Forms

setCustomValidity()

Sets a custom error message and marks the field as invalid.

validationMessage

Returns a string with the validation message for the element.

Constraint Validation for Forms

willValidate

Returns a boolean indicating if the element is a candidate for constraint validation.

Constraint Validation for Forms

[validity](#)

Returns a ValidityState object with the validity states of the element.

```
> $0.validity
<· ▼ ValidityState {valueMissing: true, …} i
    badInput: false
    customError: false
    patternMismatch: false
    rangeOverflow: false
    rangeUnderflow: false
    stepMismatch: false
    tooLong: false
    tooShort: false
    typeMismatch: false
    valid: false
    valueMissing: true
  ▶ [[Prototype]]: ValidityState
```

Constraint Validation for Forms

The invalid event

Fires when a submittable element has been checked and doesn't satisfy its constraints

e.g in JSX

```
<input type="text" onInvalid={(e) => console.log("oh nos")} />
```

Demo time

demo deployed at
formFUN.dev

Forms & Frameworks

Forms & Frameworks

- web standards in frameworks: Request, Response, FormData
  - Fresh
  - Remix
- Frameworks with dedicated <Form /> components:
  - Next.js Form component
  - Remix Form component
  - Redwood.js Form components_hooks
  - Nuxt Form component

Forms & Frameworks

- React 19 showing forms some love:
  - Server actions
  - Form hooks
- Actions in general for frameworks like:
  - Next.js
  - Astro
  - SolidStart
- Some frameworks allowing you to POST or GET to the same page like the good ol’ days

Resources

Resources

- NCSA Mosaic
- Netscape Navigator
- formFUN.dev
- <form>: The Form element
- Constraint validation
- ValidityState
- HTML5
- MSXML
- Jesse James Garrett

Resources

- Very small CSS tweaks for better forms
- A deep dive on forms with modern React
- next/form
- <Form> – Remix
- Nuxt - Form
- Accepting form data from an action – Astro
- Tailwind - Handling Hover, Focus, and Other States
- Fresh - Forms

Resources

- React 19 Features: Actions - Explained in 15 mins | React Tutorials
- Fresh - Form Submission
- Archive Team - Geocities
- Dancing Baby
- "(Ab)use the Platform!" by Jon Jensen at #RemixConf 2023 💿

Slide Deck

nickyt.online

Feedback Form

nickyt.online

That’s all folks!

Thank you!

Stay in touch!

- @nickytonline everywhere
- nickyt.online