



DevX **HTML/CSS**

Class 11 - Forms and Inputs



Review

We learned about *User Interface Design*:

- Three key areas:
 - Information Architecture
 - Visual Design
 - Interaction Design
- Information Architecture
 - Organizing, simplifying, and managing user interaction/information processing
- Visual Design
 - Balance: Symmetry, asymmetry, & radial
 - Consistency, Whitespace, Scale, Proportion, Golden Ratio, Movement
- Interaction Design
 - Target Audience, Psychographics, Understanding the Industry
- Information Density
- Layouts and Hierarchy
- Fonts & Typography
 - Serif, sans-serif, script fonts.
 - Primary and secondary typefaces
- Iconography
 - Menu icons
 - Icon choices

Questions?

The image features a dark gray background. In the top-left corner, there are two overlapping triangular shapes, one cyan and one magenta, pointing towards the center. In the bottom-right corner, there is a cluster of several overlapping triangular shapes in cyan and magenta, also pointing towards the center. In the center of the image, the word "Forms" is written in a bold, black, sans-serif font, enclosed within a solid orange rectangular box.

Forms

Forms

Forms collect data on the web.
You see them everywhere.

Any website that can receive input
data has a **form**.

You will use them more when
learning **JavaScript**. For now, we
will learn their structure and
syntax.

Sign Up Log In

Sign Up for Free

First Name* Last Name*

Email Address*

Set A Password*

GET STARTED

Personal Details

Salutation
--None--

First name:

Last name:

Gender : ☐ Male ☐ Female

Email:

Date of Birth: dd / mm / yyyy

Address :

Submit

Forms Syntax

(1) A `<form>` tag on the outside

```
<form>
```

Visible Text



(2) `<label>` and `<input>` tags inside.

(3) (Optional) Other tag types inside.

```
</form>
```

`<label>` provide context to the form fields.

First name:

Forms Syntax

First name:

(1) The `<label>` links with the `<input>` by linking the `"for="` attribute with the `"id="` attribute.

(2) The `<input>`'s `"name="` attribute labels the input data for storage. (More on this later)

(3) The `<input>`'s `"type="` attribute decides whether the input is a text box, radio button, etc.

```
<form>
  for
  {
    <label for="fName">First Name:</label>
    <input type="text" id="fName" name="firstName" />
  }
</form>
```

Type of Input id data name

Do a quick demonstration.

Form Submit

Form id

```
<form id="theForm">
```

```
  <label for="fname">First name:</label>
```

```
  <input type="text" id="fname" name="fname" />
```

Submit
Button



```
  <button type="submit" form="theForm">Submit</button>
```

```
</form>
```

Button can
"submit"

Submits
"theForm"

Visible Text

We've added a button to the form from the last page. The button allows us to "submit" the form. That is to say, it collects the data and sends it somewhere.

By default, this redirects the page. But there are ways to prevent that, and observe the data.

Do a quick demonstration.

Form Data

```
<script src="https://tinyurl.com/devx-form-reader"></script>
```

First name:

Last name:

The screenshot shows a web browser's developer console. At the top, there's a toolbar with icons for back, forward, search, and other functions. Below the toolbar, there's a 'Filter' input field and a 'Default levels' dropdown. The console shows '1 Issue: 1'. Below that, a message says 'Console was cleared VM5332:1'. Then, there's an 'undefined' message. Finally, there's a 'Form Data' object from 'devx-form-reader:8' with the following structure:

```
Submitted:
  NAME: devx-form-reader:13
  fname VALUE: Albert
  NAME: devx-form-reader:13
  lname VALUE: Einstein
```

For this class I made a little form-reading script. It uses **Javascript**, which we aren't covering in this class. But that's OK, we can still use it to learn about forms.

When this **<script>** is added to a page, submitting a form will display the data in the inspector's console.

You can get to the inspector by right-click on the page and picking "inspect".

Or CTRL + SHIFT + "i".

Or CMD + SHIFT + "i" for Mac.

Do a quick demonstration.



Activity: Make a Form

Activity: Make a Form

We're going to make a simple form page!
And use a tiny bit of Javascript to read the
form data.



- (1) Open a boilerplate HTML file
- (2) Put this content in the boilerplate HTML file
- (3) Open with live server
- (4) Add some data to the form,
- (5) Click Submit.

```
<form action="" method="" id="myForm">
  <label for="First Name">First Name</label>
  <input type="text" name="First Name" />
  <label for="Last name">Last Name</label>
  <input type="text" name="Last Name" />
  <button type="submit" form="myForm" value="Submit">Submit</button>
</form>

<script src="https://tinyurl.com/devx-form-reader"></script>
```

Activity: Make a Form (2)

Let's add some additional fields to the form. Radio buttons!

```
<br />
<input type="radio" id="html" name="favoriteLanguage " value="HTML" />
<label for="html">HTML</label><br />
<input type="radio" id="css" name="favoriteLanguage " value="CSS" />
<label for="css">CSS</label><br />
<input
  type="radio"
  id="javascript"
  name="favoriteLanguage "
  value="JavaScript"
/>
<label for="javascript">JavaScript</label>
<br />
```

Notice they have type="radio".

Also the labels are after the button.

Activity: Make a Form (3)

Now let's try a phone field!

Don't worry about the "pattern" attribute. You can google various ways to handle a phone "pattern". Also, you'll learn more about how to write them when you learn Javascript.

```
<br />
<label for="phone">Enter your phone number (### ### ####) </label>
<input
  type="tel"
  id="phone"
  name="phone"
  pattern="[0-9]{3}[\s-]+[0-9]{3}[\s-]+[0-9]{4}"
/>
<br />
```

This is called Regex. We'll learn it later, in Javascript. For now, just copy/paste it.



Activity: Make a Form (4)

Now try it yourself! Go online and find another type of input field to use in the form.

Try this resource on W3Schools:

https://www.w3schools.com/html/html_forms.asp

<https://www.javatpoint.com/html-form-input-types>

Some suggestions:

- Color
- Date
- Email
- Number
- Url
- Password
- Reset
- Checkbox
- Search
- Tel

File and Image input types might be too complicated for now. But feel free!



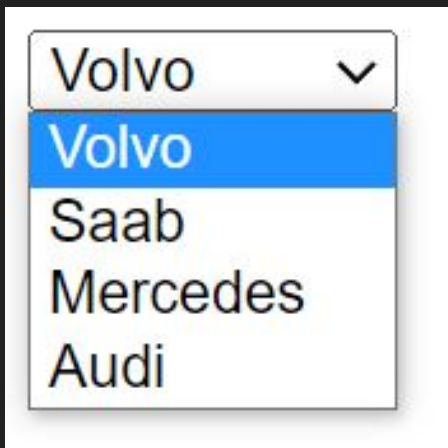
Other Form Elements

Select

Select is used when you want to make a dropdown menu.

The tag requires both a specific structure, and the value attribute on the inner items.

The `<option>` tags need to be inside the `<select>` tag, and there can be multiple.



```
<select name="cars" id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```



Activity: Select

Activity: Select

Let's make a select dropdown!

Copy this format, and change the values!

```
<select name="cars" id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

Show how to do vsCode boilerplate HTML document. Try out some tags.

Tables

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Amazon</td>
    <td>Maria Anders</td>
    <td>USA</td>
  </tr>
  <tr>
    <td>Tencent</td>
    <td>Li Chang</td>
    <td>China</td>
  </tr>
</table>
```

Tables are... tables. They have rows and columns. They are not used that often, and you can look up the structure when you need to.

They have a complex structure of tags:

- **<table>** The outer table.
- **<tr>** Table Row
- **<th>** Table Header
- **<td>** Table Data

Company	Contact	Country
Amazon	Maria Anders	USA
Tencent	Li Chang	China

Forms

Forms can be very complex. They combine groups of `<input>` tags into a dataset. They are everywhere on the internet.

We'll have a class on them later, as they are a bigger topic. They usually involve Javascript.

```
<form action="/">
  <label for="firstName">First name:</label><br>
  <input type="text" id="firstName" name="firstName" value="John"><br>
  <label for="lastName">Last name:</label><br>
  <input type="text" id="lastName" name="lastName" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

First name:

Last name:



Google Forms





Google Forms

Google forms are another way to add some functionality to your webpages without too much code.


Google has created an online tool that allows you to use, share, and add forms to webpages with just a little bit of HTML code.

I would consider using Google forms for small projects, but not for larger ones. In those cases you probably want to use a custom form with custom styling.

[https://www.google.com › forms › about](https://www.google.com/forms/about) ⋮

Google Forms: Online Form Creator | Google Workspace 

Use **Google Forms** to create online forms and surveys with multiple question types. Analyze results in real-time and from any device.





Google Forms

From there, “Go to Forms”



Google Forms

Get insights quickly, with Google Forms

Easily create and share online forms and surveys, and analyze responses in real-time.

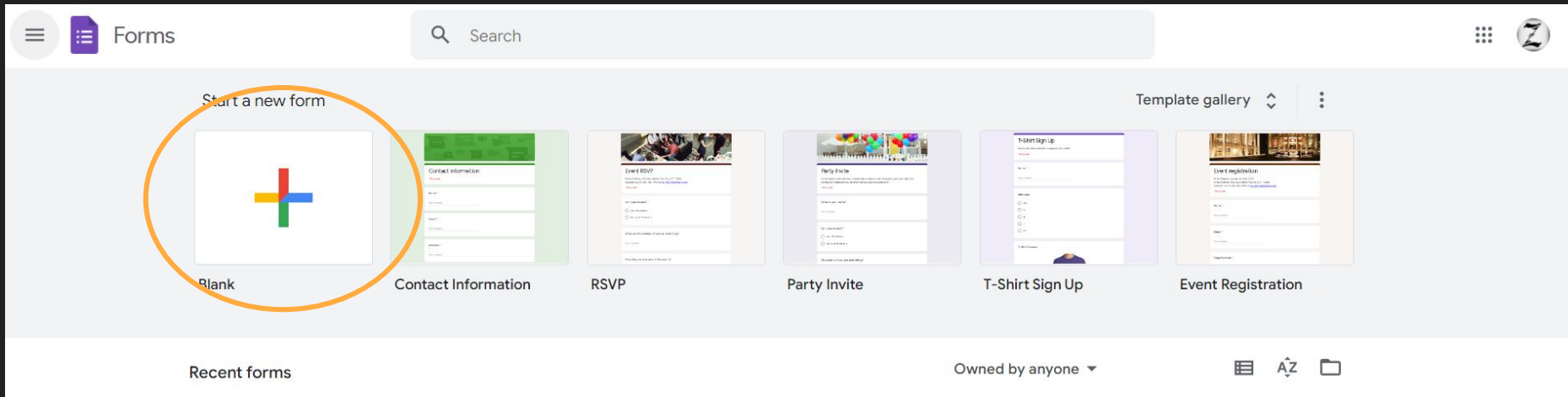
Try Forms for Work

Go to Forms



Google Forms

Choose a form type that fits your needs. Try “Blank” for now.



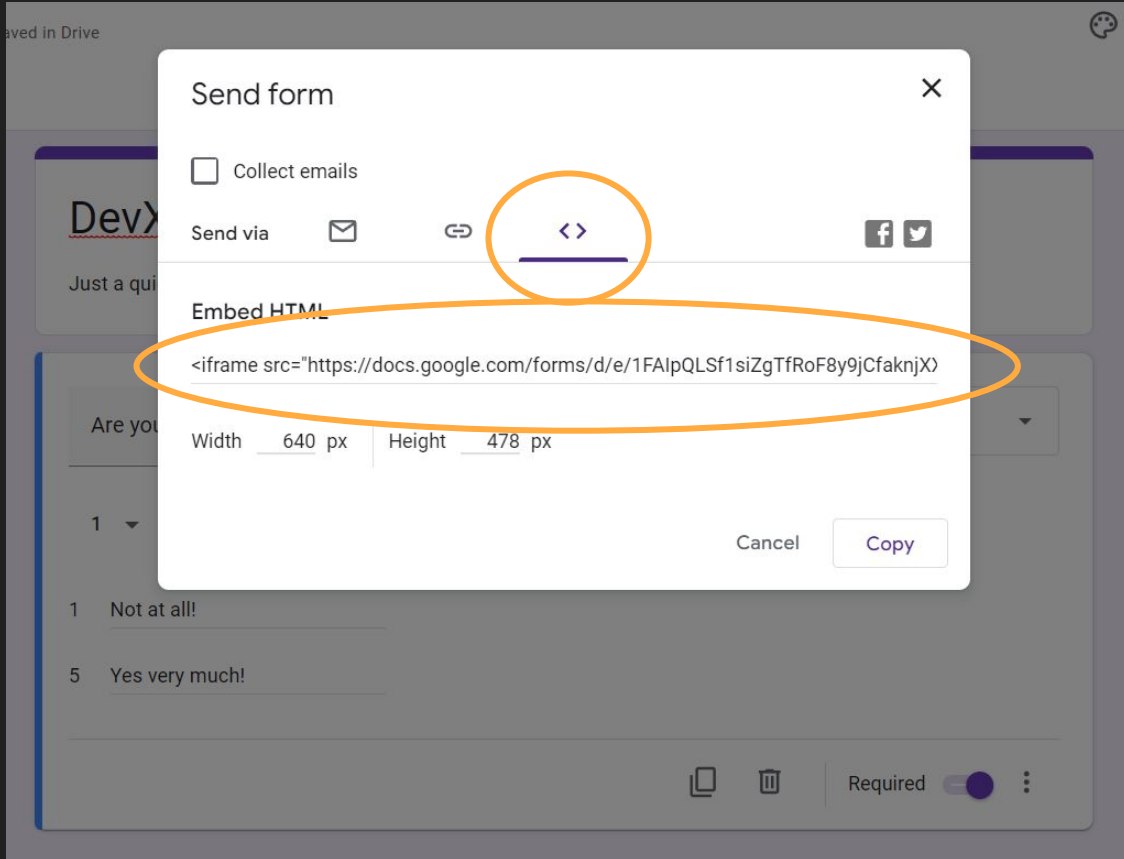
Google Forms

Here you can build the questions. It takes a little time to get used to all the possible settings.

When you are done, the eye to “preview” or click “Send” top open up a modal.

The screenshot displays the Google Forms editor interface. At the top, the header bar includes the text 'Untitled form', a folder icon, a star icon, and the status 'All changes saved in Drive'. On the right side of the header, there are icons for a question, a preview (eye icon), a share icon, and a 'Send' button, all of which are circled in orange. Below the header, there are tabs for 'Questions', 'Responses', and 'Settings', with 'Questions' being the active tab. The main content area shows a form titled 'DevX Test' with the subtitle 'Just a quick trial form.' Below this, there is a question: 'Are you enjoying learning about web development?'. The question is set to a 'Linear scale' type. The scale is configured from 1 to 5, with '1' labeled 'Not at all!' and '5' labeled 'Yes very much!'. At the bottom right of the question card, there is a 'Required' toggle switch which is currently turned on. On the right side of the form, there is a vertical toolbar with icons for adding new questions, duplicating, deleting, and other editing functions.

Google Forms



In this modal, you can click on the "Embed" icon, then copy the `<iframe>` tag provided.

Google Forms

```
<iframe
  src="https://docs.google.com/forms/d/e/1FAIpQLSf1s
  iZgTfRoF8y9jCfaknjXXOX4BJGz
  WIVqPKCiy_PI570low/viewform
  ?embedded=true"
  width="640"
  height="478"
  frameborder="0"
  marginheight="0"
  marginwidth="0"
>
  Loading...
</iframe>
```



Add the `<iframe>` to your HTML file and boom! You have a form.

A screenshot of a Google Form titled "DevX Test" with the subtitle "Just a quick trial form." The form is displayed within a white frame with a purple header bar. It features a required text input field with a red asterisk and the text "* Required" below it. To the right of the input field, it says "(not shared) Switch account" and has a cloud icon. Below the input field is a question: "Are you enjoying learning about web development? *" with a red asterisk. The question has five radio button options labeled 1, 2, 3, 4, and 5. Below the options, it says "Not at all!" on the left and "Yes very much!" on the right. At the bottom of the form, there is a purple "Submit" button and a "Clear form" link. A vertical scrollbar is visible on the right side of the form.



Activity: Make a Google Form

Activity: Make a Google Form

This time we're going to use a google form.

Follow the steps from the previous slides.

1. Sign into google (if you haven't)
2. Search for Google Forms
3. Go to the Google Forms site
4. Go to Forms
5. Create a blank form
6. Create a few questions
 - a. Multiple Choice
 - b. Short Answer
 - c. Linear Scale
7. "Preview" the form (to check)
8. "Send" the form
9. Find the "embed" tab
10. Copy the "embed" HTML tag
11. Create an HTML file
12. Add the HTML tag to the file
13. Live-Server and fill out/submit the form
14. Back on the form page, check your "Responses"



Q&A