Form Builder Quick Reference

Before You Start

- Be comfortable with or brave about HTML
- Understand the Online Form HTML Specification document in the Form Builder Documentation
- Think through and plan your workflow
- Set up your groups in test.forms.northwestern.edu (we assume you have requested access)
  - Only group admins can create and manage forms. Organization admins must make themselves an admin of a group first
  - A good practice to follow is have one group "own" the form and other groups for the approval processes. For example, a two-step workflow for ordering office supplies might have an "Office Supplies Order - Form Owners" group to manage the workflow and an "Office Supplies Order - Approvers" to approve purchase requests

Starter Template Download

Download OnlineForms-Template.html

Customize Template Metadata

- Change "Online Forms" in the <head> <title> to the name of your form
- Change "Northwestern University" (or remove it) and "Online Forms" in the <body> <header> to something representative of your unit
- Change "Online Form" in the <h1> of the main content region to the name of your form
- Optionally customize the <footer> (maybe add another <p>)

Add Sections

Basic Section Code

Just a standard section you can customize

```html
<hr>
<form id="SECTION_IDENTIFIER" class="form-section visiblefromall">
<h2>SECTION_NAME</h2>
<!-- MORE ROWS AND FIELDS HERE -->
<div class="controls">
  <button type="submit" class="btn btn-hover-success" value="approve" name="submit"><i class="icon-ok"></i> Submit</button>
  <button type="submit" class="btn" value="save" name="submit"><i class="icon-disk"></i> Save as Draft</button>
  <button type="submit" class="btn btn-hover-warning" value="return" name="submit"><i class="icon-turn-left"></i> Return Form</button>
  <button type="submit" class="btn btn-hover-danger" value="reject" name="submit" onclick="if(confirm('Are you sure you want to reject this form?')) {return true;} event.returnValue = false; return false;" value="reject" name="submit" onclick="if (confirm('Are you sure you want to reject this form?')) {return true;}} event.returnValue = false; return false;"
<i class="icon-remove"></i> Reject</button>
</div>
</form>
```

Signature Section Code

A section that is really just a signature
Routing Section Code

- If you change the ID of this form section ("Route" in this example) you must change the visiblefrom-class to match ("visiblefrom-Route" in this example)
- This section assumes the ID you are assigning is "Approver." If the section you assign has a different ID, change "Approver" to that ID in both the visiblefrom-class (visiblefrom-Approver) and the name of the NetID input field (formcycle-assign-to-netid-Approver)

Sign or Route Section Code

Sometimes you want to give a user the option to sign themselves or route a form to someone else to sign. E.g., a dispatcher might sign routine requests themselves but escalate an unusual case. To achieve this we create two sections essentially with duplicate fields, hide and skip the second section when the user signs, show and activate the second section (while hiding the first section) when the user routes. This is a pretty slick but advanced technique so it is not covered in detail at this time. This template outlines the basic idea. Doing a find-and-replace of XXXXXX with a valid ID string should work (you may want to clean up the two instances in <h2> tags).

Bootstrap Rows

Each spanX is a column where all the Xs must add up to less than 12
Other Field Types

For all of these field types you will want to customize the name attribute and optionally add the "required" class to required fields (avoid the required attribute at this time)

Radio Button
Wrapping the button and text in the <label> makes the text clickable. A group of radio buttons should all have the same name

```html
<label class="radio">
  <input type="radio" name="Radio_Buttons" value="Yes"> Yes
</label>
<label class="radio">
  <input type="radio" name="Radio_Buttons" value="No"> No
</label>
```

Checkbox
Wrapping the button and text in the <label> makes the text clickable. Each checkbox must have a unique name within the form section

```html
<label class="checkbox">
  <input type="checkbox" name="Check_Box" value="Agree"> I Agree to the Terms and Conditions
</label>
```

Dropdown Menu (select)
If the required class were applied the empty option would not be a valid choice. To add text to the empty option while still making it required use <option value="">Select one...<option>

```html
<select name="Dropdown" class="form-autosize">
  <option></option>
  <option>Bienen School of Music</option>
  <option>McCormick School of Engineering and Applied Science</option>
  <option>Medill School of Journalism, Media, Integrated Marketing Communications</option>
  <option>School of Communication</option>
  <option>School of Education and Social Policy</option>
  <option>Weinberg College of Arts and Sciences</option>
</select>
```

Multiple Select Options
The empty brackets ([]) after the name are required. The size attribute dictates how tall the box will be
<select name="Multiselect[]" multiple="multiple" class="form-autosize" size="6">
  <option>Bienen School of Music</option>
  <option>McCormick School of Engineering and Applied Science</option>
  <option>Medill School of Journalism, Media, Integrated Marketing Communications</option>
  <option>School of Communication</option>
  <option>School of Education and Social Policy</option>
  <option>Weinberg College of Arts and Sciences</option>
</select>

**Text Area (large text box)**

The rows attribute dictates how tall the box will be

```html
<textarea name="Textarea" rows="6" class="form-autosize"></textarea>
```

**Upload Attachment**

```html
<input type="file" name="Attachment" class="form-autosize fancyupload">
```

**Date**

Adds a datepicker interface when clicked. As of this writing this produces more consistent results than using the "date" HTML5 type

```html
<input type="text" name="Date" class="form-autosize datepicker">
```

**Email**

Includes the @ symbol on the keyboard of some mobile devices, limited browser validation

```html
<input type="email" name="Email" class="form-autosize">
```

**URL**

Includes the : and / symbols on the keyboard of some mobile devices, limited browser validation

```html
<input type="url" name="URL" class="form-autosize">
```

**Telephone**

Defaults to the numeric keypad on some mobile devices, limited browser validation

```html
<input type="tel" name="Phone" class="form-autosize">
```

**Number**

Defaults to the numeric keypad on some mobile devices, limited browser validation

```html
<input type="number" name="Number" class="form-autosize">
```

**Assigning Sections Based on User Input**

Assigning sections to a fixed group/s is done on the access control screen. These instructions detail how to assign a section based on user input where "SECTIONID" in each example is the ID of the section you want assigned
To One of a Pre-Determined List of Groups ("magic dropdown")

Leave the `<select>` empty. It will be populated based on selections on the access control screen (below)

```html
<select name="formcycle-assign-to-group-SECTIONID" class="form-autosize required"></select>
```

To Any NetID

```html
<input type="text" name="formcycle-assign-to-netid-SECTIONID" class="form-autosize required">
```

Double-Checking and Testing

- Are all required fields marked as required? (add the "required" class to required fields, avoid the "required" attribute at this time)
- Remove unwanted buttons from sections (e.g., short sections don't need a "Save as Draft" button)

Setting Up Access Controls

(after you submit the form code)

- The initial section should almost always be set to any (unless you know what you're doing)
- Any section inheriting permissions should be set to any (not covered in this document)
- Any section where the user assigns to an individual NetID should be set to any
- Any web service section should be set to any (not covered in this document)
- EVERYTHING ELSE SHOULD BE SET TO SOME GROUP/S or that form section won't be accessible ("black hole")
- For sections where users choose the groups using a magic dropdown (above) select ALL the candidate groups

More Advanced Things?

- Inherit Section Permissions from Previous Sections: for groups or for users
- Make Sections Optional
- Customize Email Alerts (email notice cheat sheet)
- Set Section/Content Visibility
- Web Services/Hooks
- Conditionally Show/Disable Content, and much, much more