

The Six Components Of Forms

1. **Labels** (These tell users what the corresponding input fields mean.)
2. **Input Fields** (Input fields enable users to provide feedback. They include text fields, password fields, check boxes, radio buttons, sliders and more.)
3. **Actions** (These are links or buttons that, when pressed by the user, perform an action, such as submitting the form.)
4. **Help** (This provides assistance on how to fill out the form.)
5. **Messages** (Messages give feedback to the user based on their input. They can be positive (such as indicating that the form was submitted successfully) or negative (“The user name you have selected is already taken”).)
6. **Validation** (These measures ensure that the data submitted by the user conforms to acceptable parameters.)

Tackling Usability Via Three Aspects Of Forms

Despite differences in layout, functionality and purpose, all forms have three main aspects.

1. **Relationship** (Forms establish a relationship between the user and the organisation.)

Principles:

- a. Relationships are based on trust, so establishing trust in your form is critical. This can be achieved through the logo, imagery, color, typography and wording. The user will feel at ease knowing that the form comes from a sincere organisation.
- b. Every relationship has a goal, ask yourself, what is the goal of your form?
- c. Base the name of the form on its purpose. That name will inform users what the form is about and why they should fill it in.
- d. Just as in a relationship, getting to know the other person is essential. Get to know your users and always consider whether the questions you’re asking are appropriate and, if so, whether they are timely. This will instill a natural flow to your form.
- e. Knowing your users will also help you choose appropriate language and remove superfluous text. And it will help you craft an interface that balances your needs and the user’s.
- f. Do not ask questions beyond the scope of the form. In a relationship, you would become distrustful of someone who asked questions that were out of place. The same thing happens online. Consult with relevant stakeholders to see what information really is required.
- g. Sudden changes in behavior or appearance will make users edgy. Likewise, never introduce sudden changes between forms or between steps in a form.

2. **Conversation** (They establish a dialogue between the user and the organisation.)

A form is a conversation. And like a conversation, it represents two-way communication between two parties, in this case, the user and the organisation. In fact, the user has filled out the form in order to initiate communication with the organisation.

Guidelines:

- a. As mentioned, a form is a conversation, not an interrogation. Aggressive wording in labels will make users feel edgy, and (if they do not leave) they will most likely give you the answers that you want to hear, rather than the truth.
- b. Order the labels logically, reflecting the natural flow of a conversation. For example, wouldn’t it be weird to ask someone their name only after having asked a number of other questions? More involved questions should come towards the end of the form.
- c. Group related information, such as personal details. The flow from one set of questions to the next will better resemble a conversation.
- d. As in a real conversation, each label should address one topic at a time, helping the user to respond in the corresponding input field.
- e. The natural pauses in a conversation will indicate where to introduce white space, how to group labels and whether to break the form up over multiple pages.

- f. In any conversation, people get distracted by background noise. So, remove clutter such as banners and unnecessary navigation that might distract users from filling out the form.

3. Appearance (By the way they look, they establish a relationship and a conversation.)

The appearance or user interface (UI) is central to the usability of a Web form.

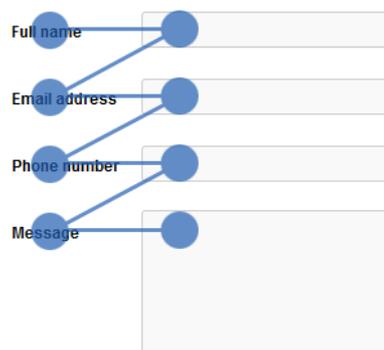
LABELS

- **Individual words vs. sentences**
If the purpose of a label is simple to understand, such as to ask for a name or telephone number, then a word or two should suffice. But a phrase or sentence might be necessary to eliminate ambiguity.
- **Sentence case vs. title case**
Should it be “Name and Surname” or “Name and surname”? Sentence case is slightly easier—and thus faster—to follow grammatically than title case. One thing is for sure: never use all caps, or else the form would look unprofessional and be difficult to scan.
- **Colons at the end of labels**
UI guidelines for some desktop applications and operating systems such as Windows recommend adding colons at the end of form labels. Some designers of online forms adhere to this, primarily because old screen readers mostly rely on the colon symbol to indicate a label. Modern screen readers rely on mark-up (specifically, the label tag). Otherwise, the colon is a matter of preference and neither enhances nor detracts from the form’s usability, as long as the style is consistent.
- **Alignment of labels: top vs. left vs. right**
Contrary to common advice, above the input field is not always the most usable location for a label. It’s ideal if you want users to fill in the form as fast as possible. But there are times when you’ll want to deliberately slow them down, so that they notice and read the labels attentively. Also, keeping a long form to a single column and making users scroll down the page is better than breaking it up into columns in an attempt to keep everything “above the fold.” Each style of alignment has its advantages and disadvantages:

Top Aligned Labels



Left Aligned Labels



In the other example demonstrating left aligned labels, the number of visual fixations and directions is doubled (8, and left – right, left – right, left – right, left – right), making it more difficult to visually consume and comprehend.

- **Placement of labels**
Never place labels inside the fields.

This scenario typically involves the label visible inside the field which disappears on focus (when a user tabs or click into it). If the user forgets what the field is (and there are many reasons why this might happen) then the only way to reveal it is to click or tab outside the field, read the label, and click or tab back into it again. Insane. But what's more insane is designers still force this unusable habit on unsuspecting users, which appears to be designed to solve an aesthetic 'problem' at the expense of usability.

Clearly indicate mandatory fields.

If a field is required, then come right out and say it. Anyone who has filled out web forms over the years might understand the meaning of the asterisk next to a label, but what about less experienced users?

INPUT FIELDS

- **Type of input field**
Provide the appropriate type of input field based on what is being requested. Each type of input field has its own characteristics, which users are accustomed to. For instance, use radio buttons if only one option of several is permitted, and check boxes if multiple choices are allowed.
- **Customizing input fields**
Do not invent new types of input fields. This was common in the early days of Flash websites, and it seems to be making a comeback; I have seen some odd input fields implemented with jQuery. Simple is often the most useful. Keep input fields as close to their unaltered HTML rendering as possible.
- **Restricting the format of input fields**
If you need to restrict the format of data inputted by users, then at least do so in a way that won't irritate users. For example, instead of displaying MM/DD/YYYY next to a text field for a date, consider using three drop-down fields or, better yet, a calendar control.
- **Mandatory vs. optional fields**
Clearly distinguish which input fields cannot be left blank by the user. The convention is to use an asterisk (*). Any symbol will do, as long as a legend is visible to indicate what it means (even if it's an asterisk).

Mandatory fields clearly labeled

Full name (required)

Email address (required)

Phone number

Message

Meaningless asterisk

Full name *

Email address *

Phone number

Message

(* indicates a required field)

Given the benefit of the doubt, let's pretend that a user does understand the meaning of an asterisk in the context of a web form. Who is to say they may even notice it? Someone with a slight visual disability – say a mild dyslexia – might not notice something so small.

With this in mind, it's just a lot simpler to explain in the label: (required). **Note:** To be fair, the WCAG 2.0 spec [suggests both methods](#) (**required** or the **astrisk**), but the latter is conditional on the meaning of the asterisk being explained at the start of the form. But to avoid any confusion and to minimise the risk of users missing the explanation (after all people don't read, they skim – especially on web forms), I recommend the first technique.

Reduce the number of mandatory fields.

This is a no-brainer. It can already be difficult to convince users to fill in a form, especially given the amount of time it can require. While I know you'll have the guy in marketing insisting you capture every piece of information right down to the brand of toilet paper they use, it's also your job to tell him that more mandatory fields means less conversions.

- **Grouped vs. Ungrouped**

It's not uncommon now to see sign-up forms ask for nothing more than an email address, and perhaps even a name. But if you have a larger complex form, have a good think about what benefit you're really going to get from some of those mandatory fields, and either make them non-mandatory or drop them entirely.

Group related fields with fieldsets.

Dividing up sections of the form will make for much better comprehension and presentation. It also in some cases gives the impression of being easier to fill in, because the user will see a few small sections as opposed to a singular intimidating mess.

Grouped with fieldsets	Ungrouped
<p>Your personal details</p> <p>First name <input type="text"/></p> <p>Last name <input type="text"/></p>	<p>First name <input type="text"/></p> <p>Last name <input type="text"/></p>
<p>Your contact details</p> <p>Email address <input type="text"/></p> <p>Phone number <input type="text"/></p>	<p>Email address <input type="text"/></p> <p>Phone number <input type="text"/></p>
<p>Your location</p> <p>Country <input type="text"/></p> <p>State <input type="text"/></p> <p>Suburb <input type="text"/></p>	<p>Country <input type="text"/></p> <p>State <input type="text"/></p> <p>Suburb <input type="text"/></p>

- **Whitespace**

One of my biggest gripes is cramped form design. In the example below, it's obvious which one is easier to use.

Generous white space

Your contact details

Email address

Landline number

Mobile number

Country

Cramped, usability compromised

Your contact details

Email address

Landline number

Mobile number

Country

The biggest problem (aesthetics aside) with the form on the right is it's unclear if the label is associated with the field above it or below it. Though it may be reasonable to expect the label should be at the top, not all users know this, and some designers even place labels below the field. I know of at least one WordPress e-commerce plugin which does exactly that out of the box, and resulted in many confused clients and customers, not to mention likely lost sales.

- **Height and width**

Not only should you provide adequate space around the fields, but also inside them as shown in the left sample. The right sample, which aside from the obvious aesthetic deficiency could also make it difficult (or impossible) for users who suffer from dyslexia to comprehend.

Name

Email address

Comments

Allow adequate field width and height.

Small input fields are annoying because they are by default, unusable. How many times have you suffered through a situation like this?

- **Correct keyboard for mobiles**

By adding `type="number/url/email"` in the input tag we can specify which type of keyboard we want to load for the mobile device.

- **Password Masking**

Most password input fields in Web forms instantly obscure any characters people enter in an effort to keep sensitive information (like your password) hidden from prying eyes. Automatic masking of people's passwords may provide the appearance of security, but it can also create usability issues as well when people are left staring at a row of bullets they hope (but can't verify) is their password.

Several mobile devices have implemented a solution that addresses this issue by displaying the most recent character someone has entered, then changing that character into an obscured bullet only after a brief delay.

- **Pop-up Menu Controls**

Drop-down select menus are one of the hardest input types to manipulate.

First you have to click on the menu to open it, then maneuver through a potentially long list of small targets. Once you find the value you want, you need place your cursor on the right target and select it. To top it off, many implementations of drop-down menus on the Web require you to keep your cursor on the menu while navigating the list, else the menu closes. Even quite dexterous users often miss and need to start over. Couple this interaction challenge with the small screens found on mobile devices and it becomes quite obvious that a different solution for select menus is needed. When faced with drop-down select menus in Web forms, Apple's iPhone presents users with a pop-up menu control. This control displays the options within a menu in a contained list that can be scrolled at various speeds though drag, nudge, or flick gestures. The large touch targets also make it easy to select a value once you find the right one.

- **Compound Menu Controls**

Pop-menu controls can be applied to compound inputs as well. So instead of requiring three separate input fields for the month, day, and year of a date, one date field can bring up a set of pop-up menus that allow people to scroll through three lists at once to find the right answer. This approach can be applied to other kinds of compound inputs as well -like height in feet and inches for example.



ACTIONS

- **Primary vs. secondary actions**

Primary actions are links and buttons in a form that perform essential “final” functionality, such as “Save” and “Submit.” Secondary actions, such as “Back” and “Cancel,” enable users to retract data that they have entered. If clicked by mistake, secondary actions typically have undesired consequences, so use only primary actions where possible. If you must include secondary actions, give them less visual weight than primary actions.

- **Naming conventions**

Avoid generic words such as “Submit” for actions, because they give the impression that the form itself is generic. Descriptive words and phrases, such as “Join LinkedIn,” are preferred.

HELP

- **Text to accompany forms**

You should never have to explain to users how to fill out a form. If it does not look like a form or it's too complicated to fill out, then redesigning it is your only option. Accompanying text should be used only where needed, such as to explain why credit card data is being requested or how a birth date will be used or to link to the “Terms and conditions.” Such text tends to be ignored,

so make it succinct and easy to read. As a rule of thumb, do not exceed 100 words of explanation (combined).

- **User-triggered and dynamic help**

Rather than include help text next to each input field, show it only where required. You could show an icon next to an input field that the user can click on when they need help for that field. Even better, show help dynamically when the user clicks into an input field to enter data. Such implementation is becoming more common and is relatively easy to implement with JavaScript libraries such as jQuery.

- **Provide hints**

There will almost certainly be times where your form asks for information that isn't quite clear at first, or it could be interpreted in more than one way. One way to provide hints is by using 'help' icons next to the fields that reveal information when your mouse passes over them. For example:

Serial Number (required)



The serial number can be found on the back of your device

This of course isn't much help for people who don't use a mouse to navigate the web, so the more usable solution is to simply have the explanation in plain sight near the field. For example:

Serial Number (required)

(The serial number can be found on the back of your device)

MESSAGES

- **Error message**

This notifies the user that an error has occurred, and it usually prevents them from proceeding further in the form. Emphasize error messages through color (typically red), familiar iconography (such as a warning sign), prominence (typically at the top of the form or beside where the error occurred), large font, or a combination of these.

Provide visible and informative error messages.

Form validation is only useful if you explain to the user what went wrong preferably in human digestible language. Not only should error messages be in plain sight so the user doesn't have to scroll up or down to find the error message, but the message needs to be concise and offer a visual clue as to which areas need to be corrected.

Informative error message	Uninformative error message
<p>Name (required)</p> <input type="text"/>	<p>Name (required)</p> <input type="text"/>
<p>Email address (required)</p> <input type="text"/>	<p>Email address (required)</p> <input type="text"/>
<p>Comments</p> <input type="text"/>	<p>Comments</p> <input type="text"/>
<p>You forgot to fill in two mandatory fields: name and email address. Please correct this issue and try submitting the form again.</p>	<p>Failed: Fill in the mandatory fields.</p>
<p>Submit</p>	<p>Submit</p>

- **Success message**

Use this to notify users that they have reached a meaningful milestone in the

form. If the form is lengthy, a success message encourages the user to continue filling it out. Like error messages, success messages should be prominent. But they should not hinder the user from continuing.

VALIDATION

- **Only where needed**
Excessive validation is as bad as its complete absence, because it will frustrate users. Restrict validation to confirming key points (such as the availability of a user name), ensuring realistic answers (such as not allowing ages above 130) and suggesting responses where the range of possible data is finite but too long to include in a drop-down menu (such as a country-code prefix).
- **Smart defaults**
Use smart defaults to make the user's completion of the form faster and more accurate. For example, pre-select the user's country based on their IP address. But use these with caution, because users tend to leave pre-selected fields as they are.
- **Reset after submit**
Don't reset the user choices after submission
I'd be surprised if you haven't experienced this more than once. You submit the form and are then advised that you left out some mandatory fields. While you go about the business of correcting this problem, you notice that some of the choices you made earlier have been reset back to their default state. This is usually something like a checkbox to receive email spam that you had already unchecked, or more often than not a password field. If I've already made my choice or filled in that information, why should I need to do it again? Getting someone to fill in a form is already a difficult task, asking them to repeat the process is like asking them to give up and try somewhere else. On a side note: Aside from being an incredibly effective way to annoy the user, tricking them into accidentally subscribing to a mailing list is sneaky and underhanded. Credibility instantly compromised.
Set a tabindex on each form field
The tabindex attribute often doesn't get the attention it deserves. But it ensures the user who tabs through fields is doing so in the intended order. Using tabindex will also make your form accessible for users who can't rely on a mouse for navigating web sites.
Provide post-submission feedback
Once a form has been submitted, it's nice to be presented with on-screen confirmation of the fact offering information about what will happen next.

For a form to be usable, all three aspects need to be tackled.

Form Elements

- Input fields (fixed width, variable width, read-only)
- Text area (text box)
- Checkbox
- Dropdown
 - Basic dropdown
 - Selection dropdown
 - Search selection dropdown
- Radio box
 - Basic
 - Slider
 - Toggle
- Boolean
 - Slider
 - Content selection

- Error messages
 - Inline (field level)
 - Global
- Date
 - Input
 - Dropdown (single field, multiple field)
 - Calendar (single field, multiple field, fixed in page, to-from, jquery, input -> icon -> launch)
- Time
 - Input
 - Dropdown (single field, multiple field)
 - Slider (time, hours, minutes, to-from, jquery)
- Call-to-action (buttons; submit, cancel, close, etc)
- Phone number (integer only)
- Currency (integer, decimal)
- Image (embedded in form)
- Label
- Postcode (single field, multiple field)
- Password
- Search
- Attachment (browse -> dialogue -> selection option)
- Tooltip
- Category tree (select, reading only)
- Dynamic
 - Forms
 - Elements
- Tabs
- Data grid (tables)
- Pagination
- Steps (step 1 of 3, etc)
- Headings (overall form, sections, groups)
- Hoover
- Designs (fonts, colours, border, background, alignment, icon, mobile, tablet, desktop)
- Warning/Information messages
- Tags

Good form designs:

Join Twitter today.

Full name

✓ Name looks great.

Email address

✗ This email is already registered. Want to login or recover your password?

Create a password

6 characters or more! Be tricky.

Choose your username

Don't worry, you can change it later.

Suggestions: [MTreder](#) [FormMarcin](#) [FormMarcin](#)

Keep me signed-in on this computer

By clicking the button, you agree to the terms below:

These Terms of Service ("Terms") govern your access to and use of the services, including our various websites, SMS, APIs, email notifications,

Printable versions:
[Terms of Service](#) [Privacy Policy](#)

Create my account

Note: Others will be able to find you by name, username, or email. Your email will not be shown publicly. You can change your privacy settings at any time.

Create your website

http:// .publishpath.com

Please choose a different site address, as the one you chose already exists.

Email address is not valid. This email address already exists in our system. Do you already have an account with us?

Please enter a valid US or international number Area Code is required.

Join the LightCMS Reseller Program. It's Free!

For those who make websites for multiple clients or organizations, the LightCMS Reseller Program offers an amazing set of additional features at no cost.

- ✓ Automated Client Billing
- ✓ Rewards Including Cash Back
- ✓ Unlimited Free Demo Sites
- ✓ Setting Your Own Pricing and Profit
- ✓ Custom-Branded Admin Interface

Enable Reseller Program features on my account

You can opt in or out the LightCMS Reseller Program at any time from inside your account.

Register

Sign In



Sign Up Using Facebook

Signing up will allow your friends to find you on Etsy. We'll never post without your permission.

OR

First Name

Marcin

Your name will be publicly displayed on Etsy. These fields are optional.

Marcin

Treder

Female Male Rather not say

Email

marcin@uxpincom

Please enter a valid email address.

Password

Must be at least 6 characters.

Confirm Password

Username

marcintreder

Other suggestions: marcintreder1 mtreder

By clicking Register, you confirm that you accept our Terms of Use and Privacy Policy

Register

Sign up for the Etsy Finds newsletter

Join Twitter today.

Full name

✓ Name looks great.

Email address

✓ We will email you a confirmation

Create a password

✓ Password is perfect!

Choose your username

✓ Username is available.
You can change it later.

Suggestions: [MTreder](#) [FormMarcin](#) [FormMarcin](#)

Keep me signed-in on this computer

By clicking the button, you agree to the terms below:

These Terms of Service ("Terms") govern your access to and use of the services, including our various websites, SMS, APIs, email notifications,

Printable versions:
[Terms of Service](#) [Privacy Policy](#)

Create my account

Note: Others will be able to find you by name, username, or email. Your email will not be shown publicly. You can change your privacy settings at any time.

Create an account

Display name

Your email address

Choose a username

Type a password (minimum 8 characters, no spaces)

Try using a sentence for a strong password.

Show password

disput

against

Please enter the two words you see above

Having difficulty? [Click to refresh the Captcha.](#)
[Switch to audio.](#)

Sorry, the username 'marcin', has already been taken. Please choose another username.

Join Delicious

Sign up for Whoo!

50 projects, 500 images, 10 videos, domain binding, and technical support.

1 First, name your portfolio

Portfolio title

Portfolio address
 .sample.com

2 Now, enter your account details

Name

Email

NOTE: We'll never share your email, promise.

Password

Show Password

3 Finally, enter your payment information

[Use PayPal](#)

Card number
    

Security code
 

Expiration date
Month Year



Create your portfolio

