

Introduction to Focus



By Meggin Kearney

Meggin is a Tech Writer



By Dave Gash

Dave is a Tech Writer



By Rob Dodson

Rob is a contributor to WebFundamentals

In this lesson we'll talk about *focus* and how you can manage it in your application. Focus refers to which control on the screen (an input item such as a field, checkbox, button, or link) currently receives input from the keyboard, and from the clipboard when you paste content.

This is a great place to start learning about accessibility because we all know how to use a keyboard, it's easy to relate to and test, and it benefits virtually all users.

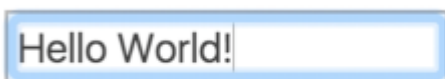
Users with motor impairments, which could be anything from permanent paralysis to a sprained wrist, may rely on a keyboard or switch device to navigate your page, so a good focus strategy is critical to providing a good experience for them.

And for the power users who know every keyboard shortcut on their machine, being able to quickly navigate your site with the keyboard alone will certainly make them more productive.

Thus, a well implemented focus strategy ensures that everyone using your application has a better experience. We'll see in the upcoming lessons that the effort you put into focus is an important basis for supporting assistive technology users and, indeed, all users.

What is focus?

Focus determines where keyboard events go in the page at any given moment. For instance, if you focus a text input field and begin typing, the input field receives the keyboard events and displays the characters you type. While it has focus, it will also receive pasted input from the clipboard.



Hello World!

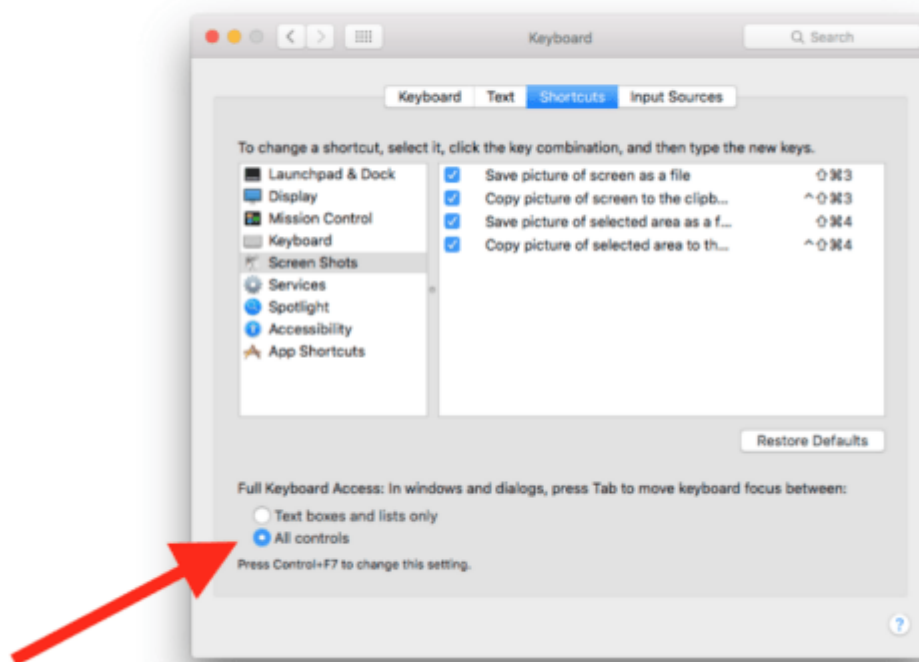
The currently focused item is often indicated by a *focus ring*, the style of which depends on both the browser and on any styling the page author has applied. Chrome, for instance, typically highlights focused elements with a blue border, whereas Firefox uses a dashed border.



Some users operate their computer almost entirely with the keyboard or other input device. For those users, focus is critical; it's their primary means of reaching everything on the screen. For that reason, the Web AIM checklist states in section 2.1.1 that [all page functionality should be available using the keyboard](#) [\[7\]](#), unless it's something you cannot do with a keyboard, such as freehand drawing.

As a user, you can control which element is currently focused using **Tab**, **Shift+Tab**, or the arrow keys. On Mac OSX this works a little differently: while Chrome always lets you navigate with **Tab**, you need to press **Option+Tab** to change focus in other browsers like Safari. (You can change this setting in the Keyboard section of System Preferences.)

System Preferences > Keyboard



The order in which focus proceeds forward and backward through interactive elements via **Tab** is called, not surprisingly, the *tab order*. Ensuring that you design your page with a

logical tab order is an important step that we'll cover later.

What is focusable?

Built-in interactive HTML elements like text fields, buttons, and select lists are *implicitly focusable*, meaning they are automatically inserted into the tab order and have built-in keyboard event handling without developer intervention.

Implicitly focusable

automatically in the
tab order + built-in
keyboard event handling

But not all elements are focusable; paragraphs, divs, and various other page elements are not focused as you tab through the page, and that's by design. There's generally no need to focus something if the user can't interact with it.


Not all elements are focusable

Study shows 9 out of 10 cats quietly judging their owners as they sleep

GRAND RAPIDS, MI—Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Experiencing focus

Let's try some of the focus techniques we just discussed. Using Chrome, go to this [airline site mockup page](#)  and search for a specific ticket **using only keyboard input**. The page doesn't accept mouse input, so you can't fudge the exercise (not that we don't trust you ;-).

Udacity *Down Under* Flights Hotels Rental Cars

☒ One Way ☐ Round Trip ☐ Multi City

Where ya headed, mate?

Full Name

Enter your address

Departure Arrival Depart Date Return Date

mm/dd/yyyy mm/dd/yyyy

Preferred seat type

No preference

☒ Receive promotional offers?

Search

About us Join our newsletter Need Support? © 2015 Udacity

The ticket parameters you should specify are:

- one way
- to Melbourne
- leaving on 12 October 2017 (10/12/2017)
- returning on 23 October 2017 (10/23/2017)
- window seat
- do not want to receive promotional offers

When you successfully complete the form with no input errors and activate the Search button, the form will simply clear and reset. Go ahead and complete the form, then come back.

Let's examine how the form uses your keyboard input. Starting with your first few Tab presses, the browser highlights the navigation items for Flights, Hotels, and Rental Cars. As you continue to press Tab you proceed to the radiobutton group where you can choose from Round Trip, One Way, or Multi City using the arrow keys.

Continue through the name and address fields, filling in the required information. When you arrive at the destination select element, you can use the arrow keys to choose a city, or you can start typing to autocomplete the field. Similarly, in the date fields, you can use the arrow keys or just type a date.

Selecting a seat type also relies on the arrow keys, or you can type "w", "a", or "n" to jump to a seat option. Then you can disable the promotional offers default by pressing the spacebar while the checkbox is focused. Finally, focus the Search button and press Enter to submit the form.

It's very handy to interact with a form using just the keyboard and to not have to switch to the mouse and back to complete a task. Because all of the elements used in the form are native HTML tags with implicit focus, the form works fine with the keyboard, and you don't have to write any code to add or manage focus behavior.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.