

JavaScript

izd3

Introductory Interactivity JavaScript Snippets

The following snippets use jQuery for basic website interactivity.

Feel free to copy and paste all snippets in this document. No citation is required.

To use these snippets, you must load jQuery **before** your JavaScript script (e.g. cookie-banner.js) at the very *end* of the <body> element:

```
<script src="scripts/jquery-3.6.1.js"></script>
<script src="scripts/cookie-banner.js"></script>
</body>
```

Access Element Snippet & Examples

Access element(s) snippet:

```
$(“TODO_CSS_SELECTOR”)
```

Replace *TODO_CSS_SELECTOR* with *any* CSS selector. Any [CSS selector](#) will do (example: nav > ul > li).

Note: If the selector covers more than 1 element, this snippet will return all elements described the selector.

Access Element by id= Example

If you want to find a specific element by its id= attribute, use the **ID selector**. (Remember, id= must be **unique**.)

HTML:

```
<div id="example">
```

JavaScript:

```
$("#example")
```

Access Elements by class= Example

If you want to find elements by their class= attributes, use the **class selector**.

HTML:

```

```

```

```

```

```

JavaScript:

```
$(".slide")
```

This will return elements id="slide1", id="slide2", and id="slide3".

Access Element(s) by CSS Selector Example

Let's say that I only want to access the elements of class slide that are direct children of a <div id="slides">. Let's use the **child combinator selector**.

HTML:

```
<div class="slides">
```

```
  
```

```
  
```

```
  
```

```
</div>
```

```

```

JavaScript:

```
$(".slides > .slide")
```

This will return elements `id="slide1"`, `id="slide2"`, and `id="slide3"`.

Event Snippets & Examples

Event snippets let you react to an event that occurs on an element. For example, a user clicks on an element.

jQuery supports several [events](#). This includes [mouse](#) (e.g. click, right click, double click, hover, etc.), [keyboard](#) (e.g. key press, etc.), and [browser](#) (e.g. scroll, resize window) events.

Note: Each event snippet *includes* an access element(s) snippet.

On-Click Element Event Snippet & Example

When a user clicks on element.

```
$(".TODO_CSS_SELECTOR").click(function() {  
    // TODO: snippet(s) to respond to the click event.  
});
```

Example:

```
<button id="cookie-accept-button">Accept</button>  
  
$(".#cookie-accept-button").click(function() {  
    // TODO: hide cookie banner  
});
```

On-Page-Loaded Event Snippet

Once the page has loaded, the *ready* event will trigger. This event is useful to setting timers or initializing state.

Tip: This snippet can be useful for changing part of a page based on the time of day or day of week.

```
$(document).ready(function() {  
    // TODO: snippet(s) to respond to the ready event.  
});
```

On-Hover Element Event Snippet

When a user hovers with their mouse over an element.

Note: You will often find that using the CSS [:hover pseudo-class selector](#) is easier in most cases.

```
$(“TODO_CSS_SELECTOR”).hover(function() {  
    // TODO: snippet(s) to respond to when the mouse enters the element.  
}, function() {  
    // TODO: snippet(s) to respond to when the mouse leaves the element.  
});
```

On-Window-Resize Event Snippet

In some situations, you may want to respond when the user resizes the browser window.

Tip: When using hamburger menus the navigation menu is often hidden until the user clicks the hamburger menu. If the navigation menu is hidden and the user resizes their browser window, you may want to remove the hamburger menu, and show a navigation bar.

Replace *TODO_SCREEN_WIDTH* with a screen width like 500.

```
$(window).resize(function() {  
    if ($(window).width() < TODO_SCREEN_WIDTH) {  
        // TODO: snippet(s) to respond when the screen size is less than TODO_SCREEN_WIDTH  
    } else {  
        // TODO: snippet(s) to respond when the screen size greater than or equal to the  
        TODO_SCREEN_WIDTH  
    }  
});
```

Timer (every n seconds) Event Snippet & Example

If you want to do something every n seconds, you can set a timer *“event”*.

1. Place the timer event snippet inside the ready event snippet
2. Replace `TODO_TIME_MILLISECONDS` with the m milliseconds you want to pass before triggering the event. $1000 \times n$ seconds = m milliseconds.

```
setInterval(function() {  
    // TODO: snippet(s) to respond to when TODO_TIME_MILLISECONDS has passed.  
}, TODO_TIME_MILLISECONDS);
```

This snippet is useful to auto advance a carousel or a slideshow; every 10 seconds, go to the next slide.

```
setInterval(function() {  
    // TODO: show next slide.  
}, 10000);
```

Add/Remove CSS Class Snippets & Examples

These snippets let you add and remove CSS classes to elements.

Tip: For some CSS properties you may need to add the `!important` to force the browser to make the toggle change.

Note: Each Add/Remove CSS Class snippet *includes* an access element(s) snippet.

Add CSS Class to Element Snippet

This snippet will add a CSS class to an element.

```
$(“TODO_CSS_SELECTOR”).addClass(“TODO_CSS_CLASS_NAME”);
```

Replace `TODO_CSS_CLASS_NAME` with a CSS class name (omit the `.`).

Remove CSS Class from Element Snippet

This snippet will remove a CSS class from an element.

```
$(“TODO_CSS_SELECTOR”).removeClass(“TODO_CSS_CLASS_NAME”);
```

Replace `TODO_CSS_CLASS_NAME` with a CSS class name (omit the `.`).

CSS Class Show/Hide Snippet Example

Let's say you want to show #slide2 and also hide #slide1 and #slide3.

HTML:

```



```

CSS:

```
.hidden {
    display: none;
}
```

Hide all elements and show #slide2.

```
$(".slide").addClass("hidden");
$("#slide2").removeClass('hidden');
```

Toggle / Conditional Snippets & Example

Sometimes you want one thing to happen when you click a button and then something else to happen when you click the button again. Or sometimes you want something to happen, if an element has a particular CSS class on it.

Tip: Toggle/Conditional snippets are useful for hamburger menus. If the navigation menu/bar is visible (no hidden class), then hide it.

Replace `TODO_TOGGLE_ELEMENT` with a CSS selector that selects the HTML element you want to toggle some CSS class on: `TODO_TOGGLE_CSS_CLASS`.

```
if ($("#TODO_TOGGLE_ELEMENT").hasClass("TODO_TOGGLE_CSS_CLASS")) {
    // TODO: snippet(s) for when TODO_TOGGLE_CSS_CLASS IS on the TODO_TOGGLE_ELEMENT.
} else {
    // TODO: snippet(s) for when TODO_TOGGLE_CSS_CLASS is NOT on the
    TODO_TOGGLE_ELEMENT.
}
```

Toggle Snippet Example

```
<button id="toggle-button">Toggle Red/Blue</button>
```

```
<div id="toggle-element">
```

When the “Toggle” button is clicked, the background color will toggle between blue and red.

```
</div>
```

```
#toggle-element {
```

```
    background-color: blue;
```

```
}
```

```
.active {
```

```
    background-color: red !important;
```

```
}
```

```
$("#toggle-button").click(function() {
```

```
    if ($("#toggle-element").hasClass("active")) {
```

```
        $("#toggle-element").removeClass("active");
```

```
    } else {
```

```
        $("#toggle-element").addClass("active");
```

```
    }
```

```
});
```

Console Log Snippet & Example

Sometimes it's helpful to write a message to yourself to check that your JavaScript is working. Use this snippet to write a message to yourself. The message will show up in your browser's **Developer Tools Console** tab.

```
console.log("TODO_MESSAGE");
```

Replace *TODO_MESSAGE* with any message you want to write to yourself! This is helpful to see if your code is working or running at all.

You can also log elements by replacing *TODO_MESSAGE* with an access element snippet. **This is helpful to check if your CSS selector is finding/locating the correct elements.**

```
console.log($("#TODO_CSS_SELECTOR"));
```

As an example, let's say you want to check if your click event is working:

```
<button id="cookie-accept-button">Accept</button>
```

```
$("#cookie-accept-button").click(function() {  
    console.log("cookie accept button clicked!");  
});
```

If everything is working correctly, you should see the message "cookie accept button clicked!" in the Developer Tools Console when you click the button.

If you don't see the message when you click the button, that means your code isn't working correctly. Check that your HTML, CSS, and JavaScript code are correct. **The most common mistake is an incorrect CSS selector.** (Observe in this example \$("#cookie-accept-button") should be \$("##cookie-accept-button").)

Specific Interactivity Snippets

Some forms of interactivity are a bit more complex than just showing and hiding elements. This section presents some snippets useful for more advanced interactivity.

Note: This is a *living* list. Some Project 3 interactivity designs might need some snippets that don't yet exist here. If we see posts on Ed Discussions where a snippet is needed that doesn't exist here, we'll create it and update this document.

Carousel Snippets

A carousel is like a **rotating slide show**. The rotating part is a bit tricky. Here are a few snippets to help with *showing a slide*, *the next slide*, and *the previous slide*.

Your pseudocode should look something like this:

when #next-button clicked: (on-click-event snippet)

show next slide (next slide snippet)

```
$("#next-button").click(function() {  
    nextSlide();  
});
```

However, to use the nextSlide() snippet, you first need the big snippet below. **Copy** the big snippet below in its *entirety* once (only replace the TODOs; there is nothing else you should

do with this snippet) and then use the "Next Slide," "Previous Slide," and "Go-To-Slide-X" *below* snippets to respond to your "click" events.

Warning: Only change the TODOs! *This code is correct. You don't need to change it except for the TODOs.*

```
/* TODO: Replace TODO_SLIDES_CSS_SELECTOR with a CSS selector that selects
_all_ slides. (See "hint" above in access element snippet.) */
```

```
const SLIDES = $("TODO_SLIDES_CSS_SELECTOR");
```

```
/* See "show next slide snippet" to use */
```

```
function nextSlide() {
```

```
  /* TODO: Replace TODO_CSS_HIDDEN_CLASS with the name of the class you will use
to hide the non-active slides (i.e. 'hidden'). */
```

```
  let nextNum = SLIDES.index($("TODO_SLIDES_CSS_SELECTOR:not(.TODO_CSS_HIDDEN_CLASS)
+ 1 + 1; /* +1 for 0-index array */
```

```
  if (nextNum > SLIDES.length) {
```

```
    nextNum = 1;
```

```
  }
```

```
  showSlide(nextNum);
```

```
}
```

```
/* See "show previous slide snippet" to use */
```

```
function prevSlide() {
```

```
  /* TODO: Replace TODO_CSS_HIDDEN_CLASS with the name of the class you will use
to hide the non-active slides (i.e. 'hidden'). */
```

```
  let prevNum = SLIDES.index($("TODO_SLIDES_CSS_SELECTOR:not(.TODO_CSS_HIDDEN_CLASS)
- 1 + 1; /* +1 for 0-index array */
```

```
  if (prevNum <= 0) {
```

```
    prevNum = SLIDES.length;
```

```
  }
```

```
  showSlide(prevNum);
```

```
}
```

```
/* See "show slide X snippet" to use */
```

```
function showSlide(num) {
```

```

let index = num - 1;
let currentSlide = SLIDES.eq(index);
/* TODO: snippets to hide SLIDES and then show currentSlide
   Use 'SLIDES' or 'currentSlide' as the "Access Element Snippet"
   (e.g. 'SLIDES.addClass()' or 'currentSlide.removeClass()') */
}

```

Tip: If you want the dot status indicator at the bottom of the carousel to indicate the current slide, you will need a snippet like `let currentDot = DOTS.eq(index);` inside `showSlide(){ }` but just for the status dots. Where `DOTS` is `const DOTS = $("TODO_DOTS_CSS_SELECTOR");`.

Note: This snippet assumes you have a CSS **hidden** class and that you have hidden the slides you don't want to show when the page loads.

Show Next Slide Snippet

In your event snippet, use the following to go to the next slide:

```
nextSlide();
```

Show Previous Slide Snippet

In your event snippet, use the following to go to the previous slide:

```
prevSlide();
```

Show Slide X Snippet

If you want to go to a specific slide, for example when you click on the second status indicator dot.

```
showSlide(2);
```

Form Snippets

Forms are not permitted for Project 3 or the Group Project. This section is to be used for Lab 11.

On-Change-Input Event Snippet

The change event occurs when a form input's value changes; the user types or selects a value. This event is limited to `<input>` elements, `<textarea>` boxes and `<select>` elements. For select boxes, checkboxes, and radio buttons, the event is fired immediately when the user makes a selection with the mouse, but for the other element types the event is deferred until the element loses focus.

```
$(“TODO_CSS_SELECTOR”).change(function() {  
    // TODO: snippet(s) to react to the input’s value changing event.  
});
```

On-Form-Submit Event Snippet

When the user tries to submit a form, this snippet will intercept the request and let you check the form data for validity.

Replace the TODO with *as many* **Check Input Validity Snippets** as you need (one for each input component).

```
$(“TODO_CSS_SELECTOR”).submit(function() {  
    let formValid = true;  
    // TODO: only “Check Input Validity” snippets.  
    return formValid;  
});
```

Check Input Validity Snippet

Check the HTML component validation criteria (i.e. required).

Note: Use *Check Input Validity* snippets **inside** a single *On-Form-Submit Event* snippet!

```
if( $(“TODO_CSS_SELECTOR”).prop(“validity”).valid ) {  
    // TODO: snippet(s) for if input is valid  
} else {  
    // TODO: snippet(s) for if input is NOT valid  
    formValid = false;  
}
```

Example:

```
<div id="feedback-email" class="feedback hidden">Please enter a valid email address.</div>
```

```
<input type="email" name="email" id="request-email" required/>
```

```
$(“form”).submit(function() {  
    let formValid = true;  
    if( $(“#request-email”).prop(“validity”).valid ) {  
        // hide feedback message  
        $(“#feedback-email”).addClass(“hidden”);  
    } else {  
        // show feedback message  
        $(“#feedback-email”).removeClass(“hidden”);  
        formValid = false;  
    }  
    return formValid;  
});
```