START SCREENCAST

INTERACTIONS

WEEKLY OVERVIEW

WEEK 7	Interactions / Functions
WEEK 8	JS Lab / Holiday Break

LEARNING OBJECTIVES

- Familiarity with how transitions and transformations can be triggered using JavaScript
- Add event listeners for mouse, keyboard, and document events.
- Practice thinking and working through some common JavaScript interactions.

AGENDA

Review

Positioning / Transitions Review

Triggering Transitions

Sidebar Lab

jQuery Events

Scrollmations Lab

Modal Window Lab (If time permits)

HOMEWORK REVIEW

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Review homework assignment

TYPE OF EXERCISE

- Groups of 3 - 4

TIMING

10 min

1. Review your solution for the Mad Libs Homework
2. Have one person jot down any questions your group has
3. Were you able to complete a bonus task? Share with your group!

REVIEW • JS BASICS

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Review variables, conditionals and data types

LOCATION

- Starter Code > Compare That

TIMING

5 min 1. With a partner, write pseudo code for the compare two numbers program.

5 min 2. Write JavaScript to add functionality

CSS POSITIONING – REVIEW

CSS POSITIONING

STATIC

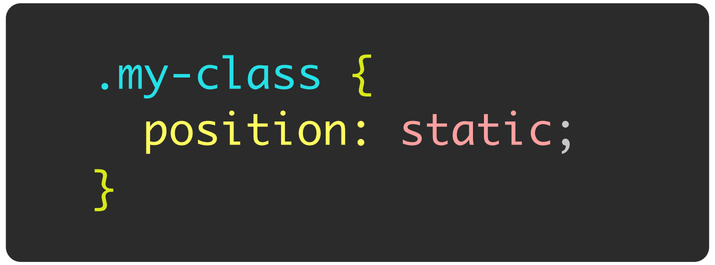
RELATIVE

ABSOLUTE

FIXED

STATIC POSITIONING

- Default positioning
- Normal flow of the document
- Elements render in order, as they appear in the document flow.



```css
.my-class {
  position: static;
}
```

CSS POSITIONING

STATIC

RELATIVE

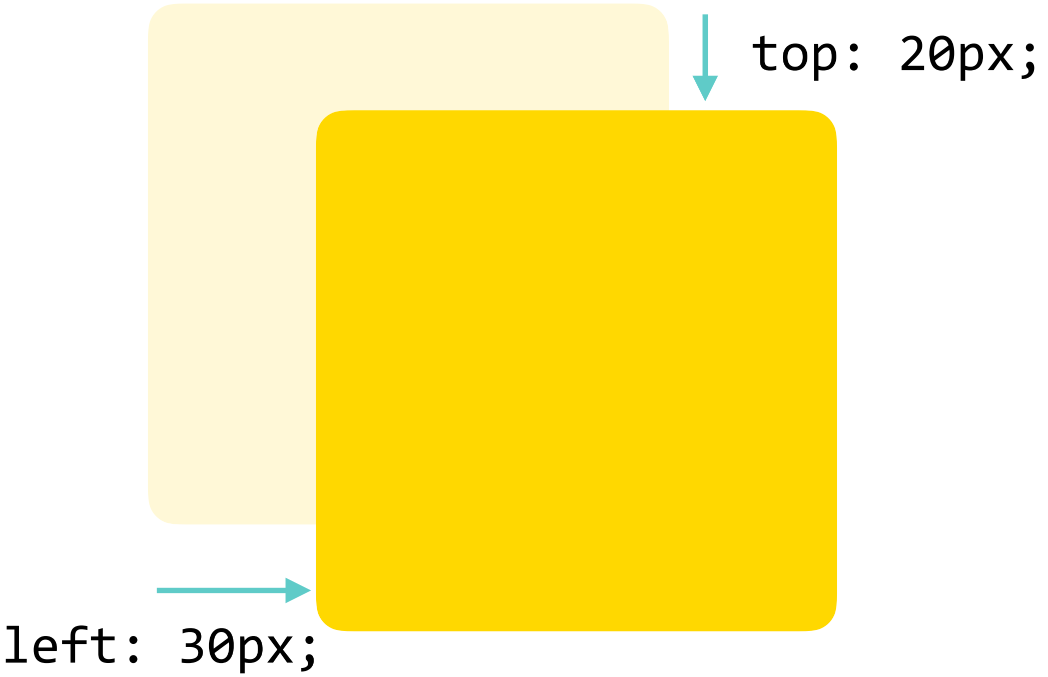
ABSOLUTE

FIXED

RELATIVE POSITIONING

- Moves an element *relative to where it would have been in normal flow*.
- For example: `left: 20px` adds 20px to an element's **left** position

```css
.my-class {
  position: relative;
  top: 20px;
  left: 30px;
}
```



top: 20px;

left: 30px;

CSS POSITIONING

STATIC

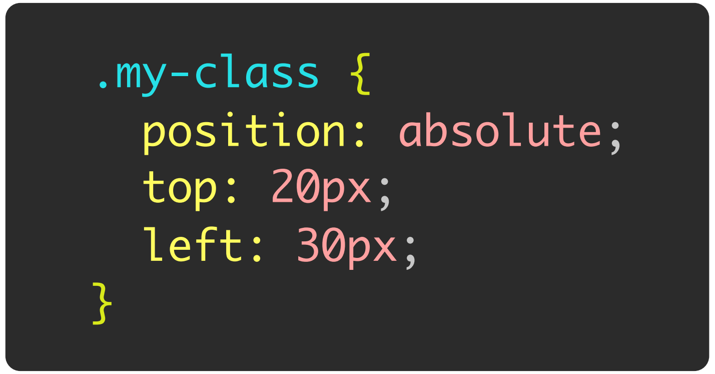
RELATIVE

ABSOLUTE

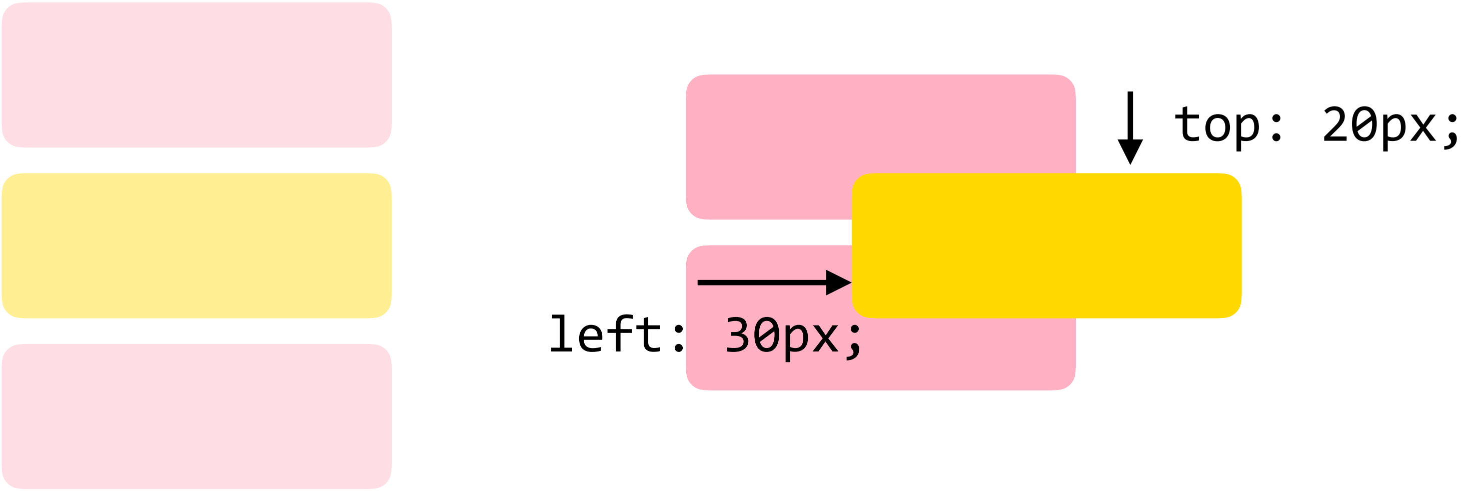
FIXED

ABSOLUTE POSITIONING

- Element is taken out of the normal flow of the document.
- No longer affects the position of other elements on the page (they act like it's not there).
- You can add the *right*, *top*, *left* and *bottom* properties to specify where the element should appear



```
.my-class {
  position: absolute;
  top: 20px;
  left: 30px;
}
```

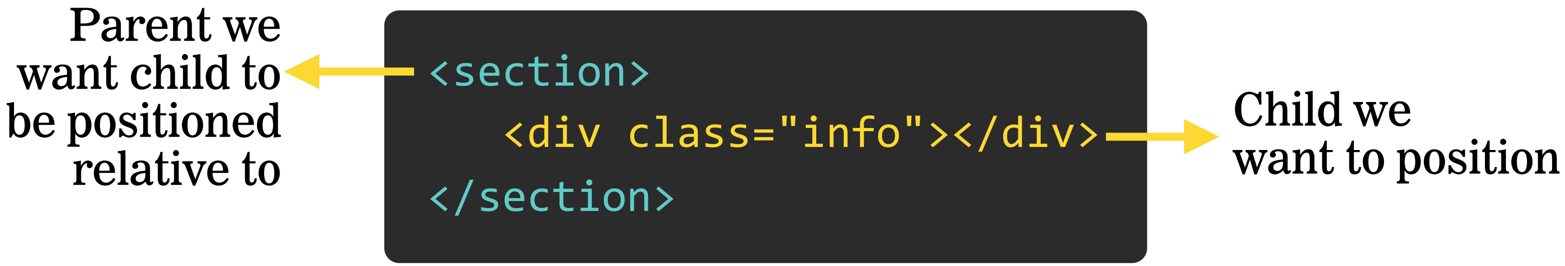


POSITIONING THINGS ABSOLUTELY

- When using `position: absolute`, *top*, *bottom*, *left* and *right* values will be relative to the element's *closest ancestor that has any position other than static*.

To position an element absolutely:

1. Set `position: relative` on ancestor element
2. Set `position: absolute` on child element and use top, right, bottom and left values to position.



```
<section>
   <div class="info"></div>
</section>
```

Parent we want child to be positioned relative to

Child we want to position

POSITIONING THINGS ABSOLUTELY

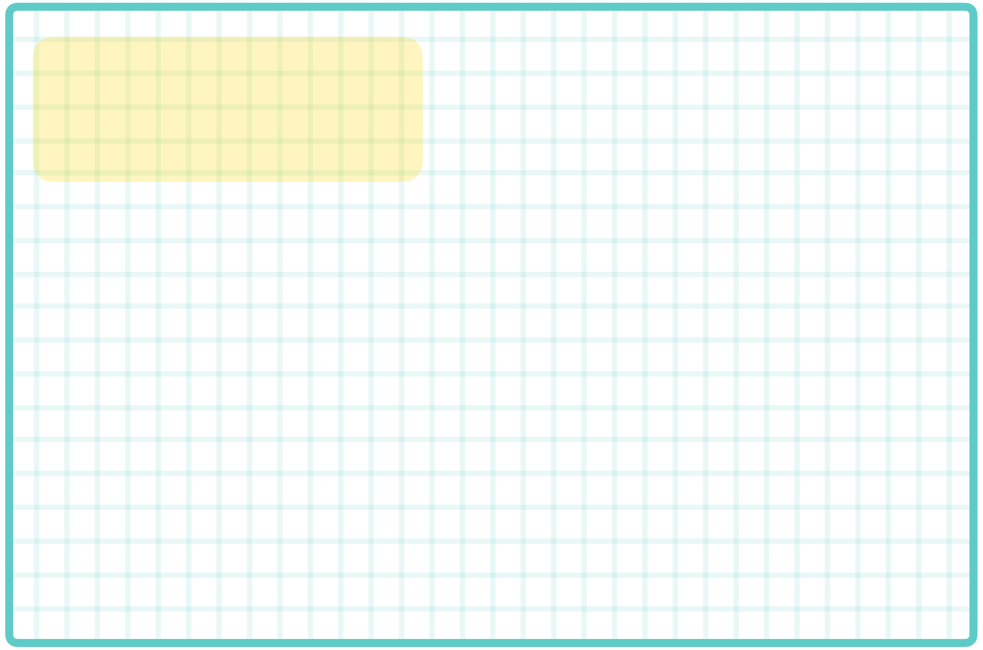
- When using `position: absolute`, *top*, *bottom*, *left* and *right* values will be relative to the element's *closest ancestor that has any position other than static*.

To position an element absolutely:

1. Set `position: relative` on ancestor element
2. Set `position: absolute` on child element and use top, right, bottom and left values to position.

```
<section>
   <div class="info"></div>
</section>
```

```
section {
  position: relative;
}
```



POSITIONING THINGS ABSOLUTELY

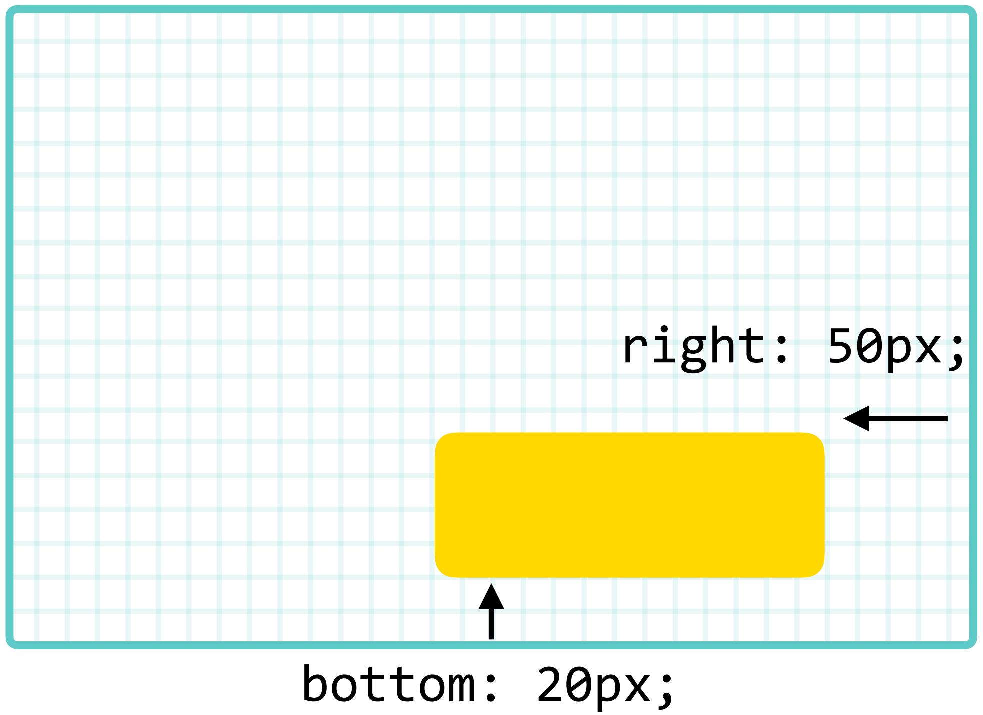
- When using `position: absolute`, *top*, *bottom*, *left* and *right* values will be relative to the element's *closest ancestor that has any position other than static.*

To position an element absolutely:

1. Set `position: relative` on ancestor element
2. Set `position: absolute` on child element and use top, right, bottom and left values to position.

```
<section>
   <div class="info"></div>
</section>
```

```
.info {
  position: absolute;
  right: 50px;
  bottom: 20px;
}
```



CSS POSITIONING

STATIC

RELATIVE

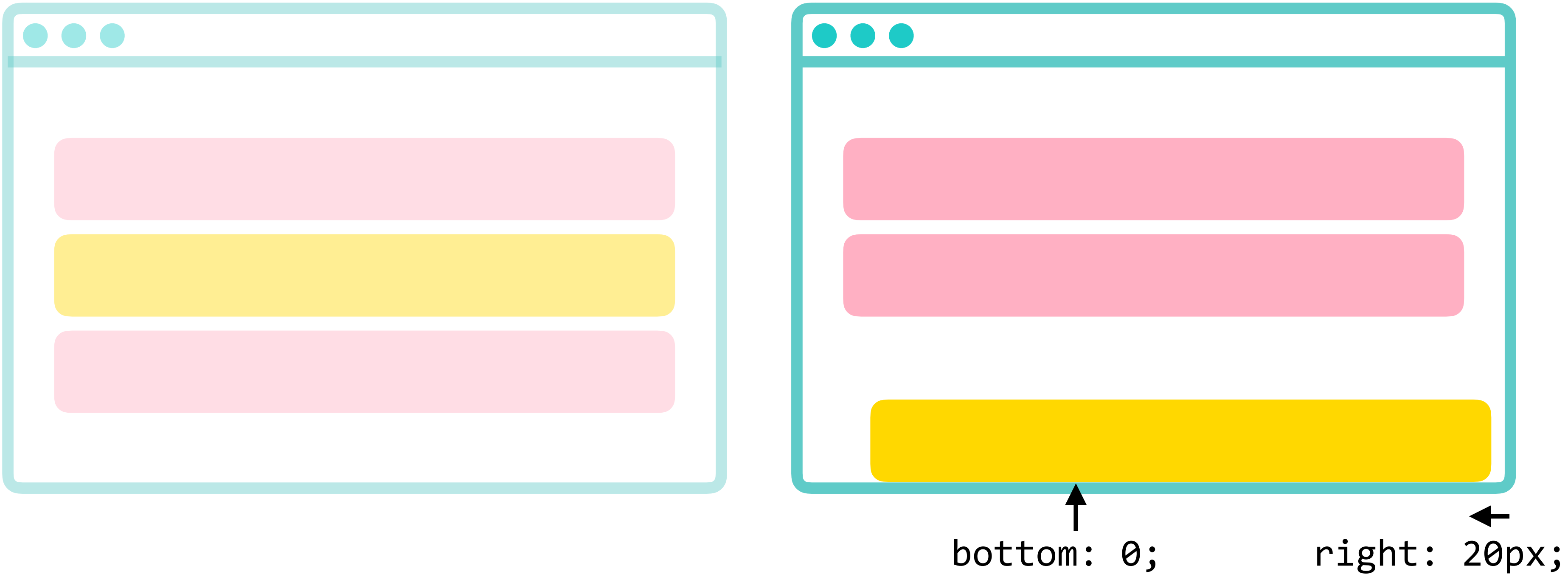
ABSOLUTE

FIXED

FIXED POSITIONING

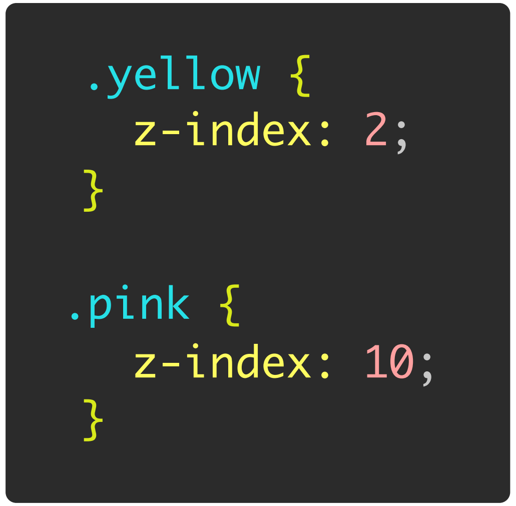
- Positioned in relation to *the browser window*
- When the user scrolls, it stays in the same place.
- Use *right*, *top*, *left* and *bottom* properties to position the element in relation to the browser window.

```css
.my-class {
  position: fixed;
  bottom: 0;
  right: 30px;
}
```



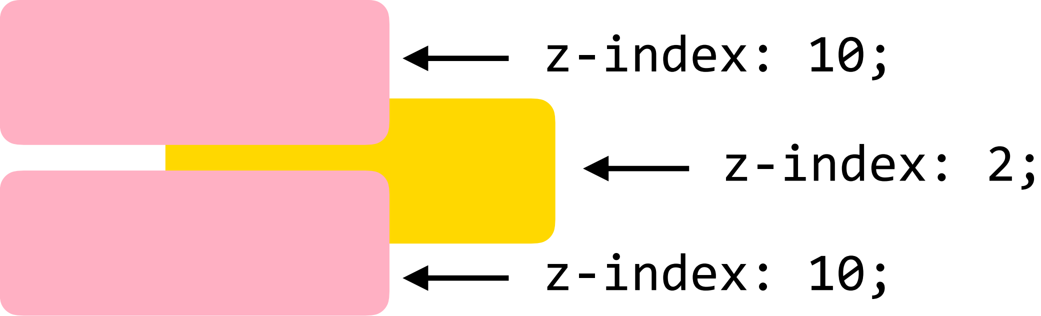
OVERLAPPING ELEMENTS — Z-INDEX

- With relative, absolute, and fixed positioning, elements can overlap.
- We can use z-index to control which elements are layered on top of each other.
- This property takes a number — the higher the number the closer that element is to the front.



```
.yellow {
  z-index: 2;
}

.pink {
  z-index: 10;
}
```

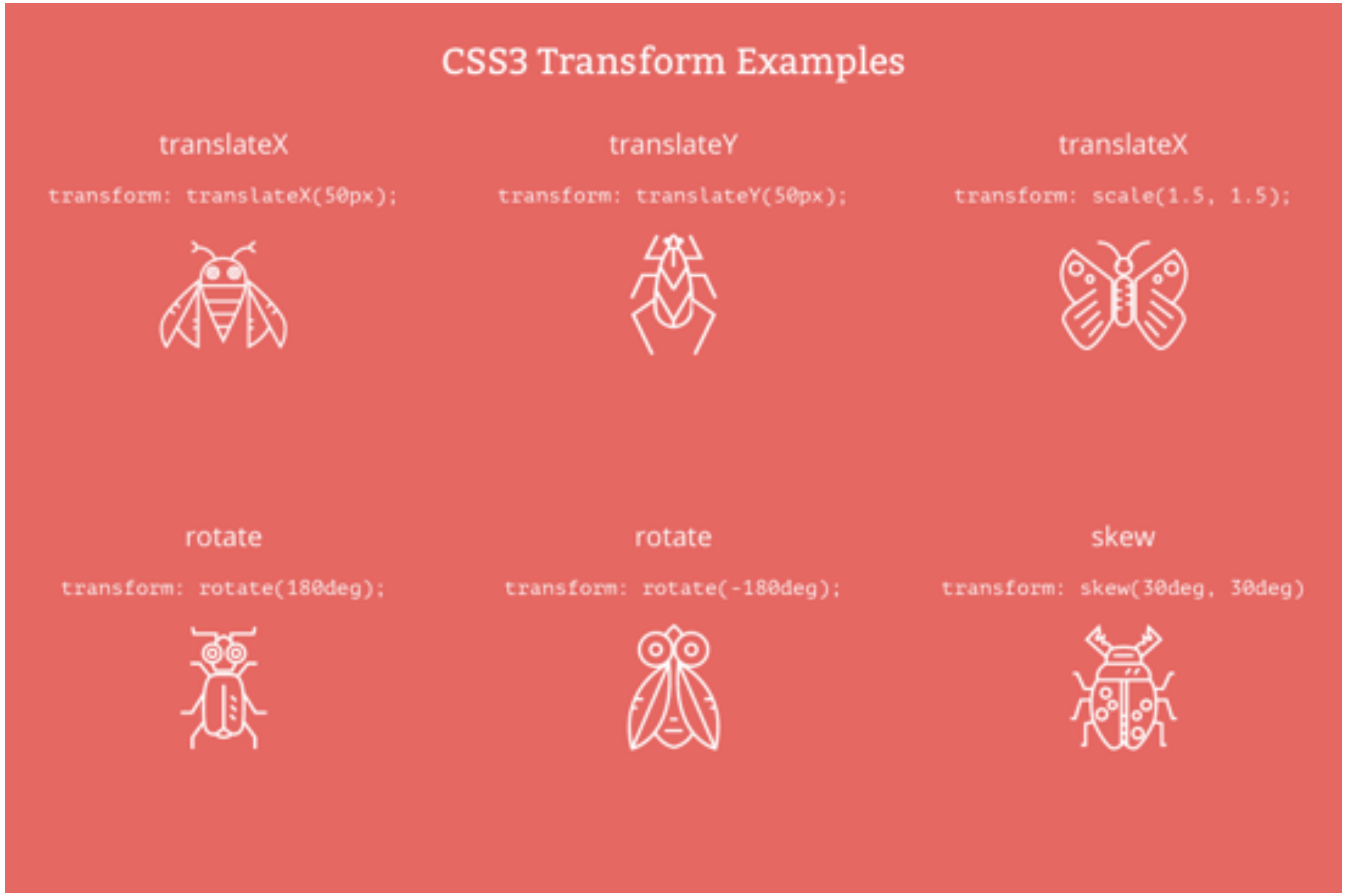


Think of this like 'bring to front' and 'send to back' in programs like Adobe Illustrator.

ANIMATION

TRANSITIONS – REVIEW

LET'S TAKE A CLOSER LOOK — TRANSITIONS



CSS3 Transform Examples

translateX

`transform: translateX(50px);`

translateY

`transform: translateY(50px);`

translateX

`transform: scale(1.5, 1.5);`

rotate

`transform: rotate(180deg);`

rotate

`transform: rotate(-180deg);`

skew

`transform: skew(30deg, 30deg)`

TRANSITIONS

- Provide a way to control animation speed when changing properties
- Instead of having property changes take effect immediately, you can have them take place over a period of time.

```
.example {
 transition: [transition-property] [transition-duration] [transition-timing-function] [transition-delay];
}
```

TRANSITIONS

DELAY

- Length of time before the transition starts

```
div {
 transition: background-color 0.5s ease 2s;
}
```

```
.example {
 transition: [transition-property] [transition-duration] [timing-function] [transition-delay];
}
```

ANIMATION

TRIGGERING TRANSITIONS

TRIGGERING TRANSITIONS

There are two ways to trigger CSS transitions:

1. Using the :hover CSS pseudo-class
2. Adding a class with jQuery

ACTIVITY — TRIGGERING TRANSITIONS

EXERCISE

KEY OBJECTIVE

- Practice using CSS transitions

TYPE OF EXERCISE

- Individual/Partner Lab

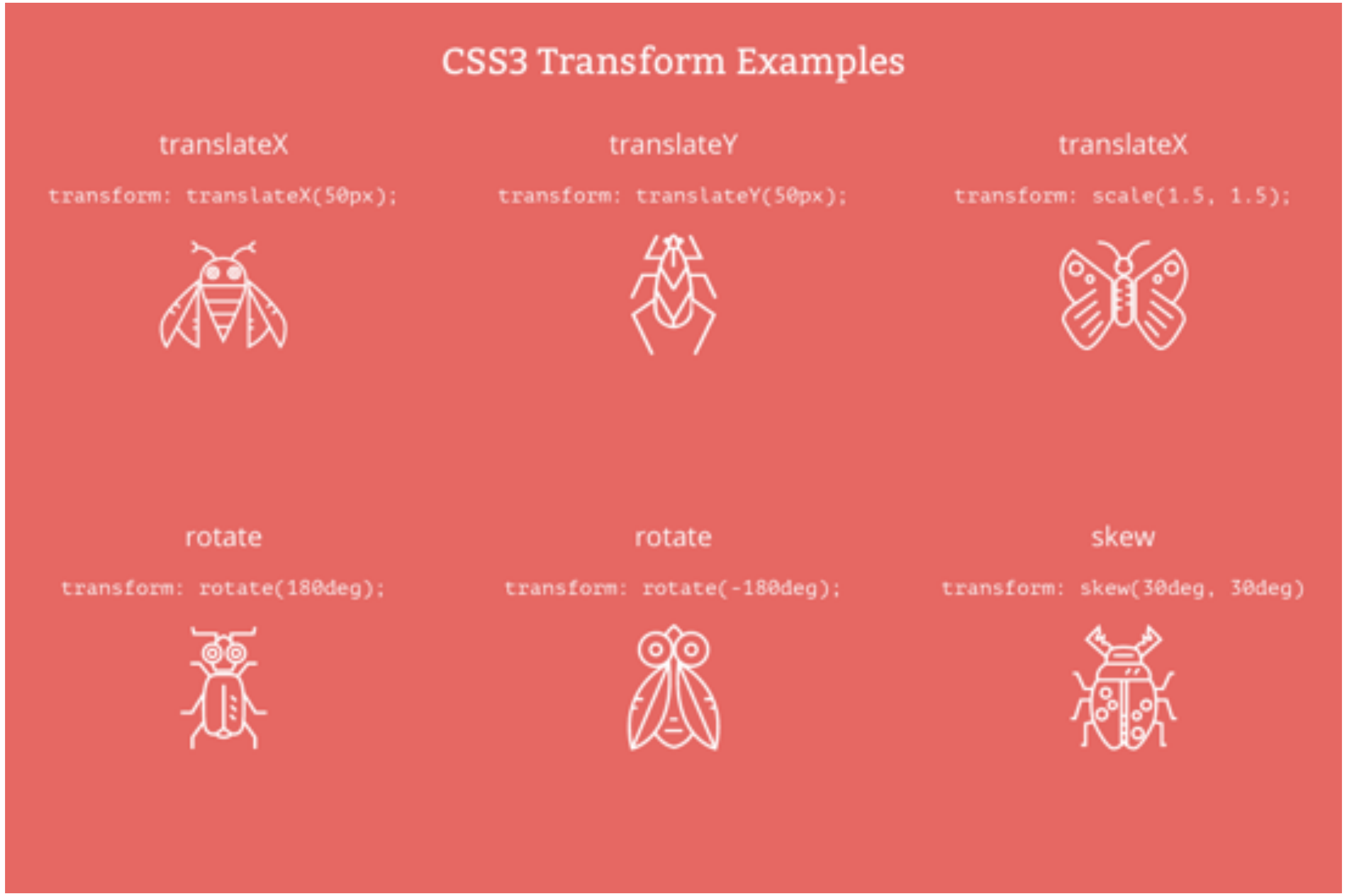
TIMING

10 min

1. Follow steps 1-2 in the main.css file in starter code > triggering_transitions.
2. Follow step 3 in the main.js file in starter code > triggering_transitions.

TRANSFORMATIONS – REVIEW

LET'S TAKE A CLOSER LOOK — TRANSFORM



CSS3 Transform Examples

translateX

```
transform: translateX(50px);
```

translateY

```
transform: translateY(50px);
```

translateX

```
transform: scale(1.5, 1.5);
```

rotate

```
transform: rotate(180deg);
```

rotate

```
transform: rotate(-180deg);
```

skew

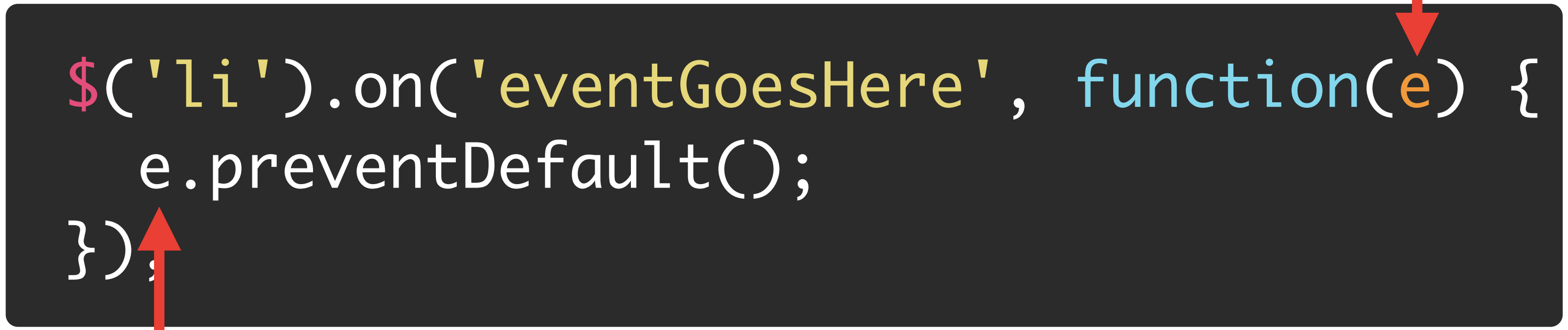
```
transform: skew(30deg, 30deg)
```

Syntax: W3 Schools

JQUERY METHODS — THE EVENT OBJECT

- The event object has properties and methods that tell you more about the event that took place.
- By using the preventDefault method, the default action of the event will not be triggered.

Parameter name



```
$('li').on('eventGoesHere', function(e) {
  e.preventDefault();
});
```

Use that name in the function and use dot notation to access its properties and methods.

CREATE EVENT LISTENERS

ANIMATION

SIDEBAR LAB

ACTIVITY

EXERCISE

LOCATION

- starter code > sidebar_menu

KEY OBJECTIVE

- Practice thinking through some common JS interactions

TIMING

3 min 1. Chat through how to tackle this interaction with a partner

15 min 2. Follow the instructions in main.css to make the menu interactive. One step will require writing JS

3. BONUS: Also move the .content section over when the hamburger icon is clicked.

JQUERY EVENTS

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

We can use the `on()` method to handle all events in jQuery.

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

selector

```
$('li')
  // your code here
}
```

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

method for all events

```
$        .on(
  // your code here
});
```

JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

type of event

```
$          'click'
  // your code here
}
```

MOUSE

click

dblclick

mouseenter

mouseleave

KEYBOARD

keypress

keydown

keyup

FORM

submit

change

focus

blur

DOCUMENT

resize

scroll

```
$('li').on('eventGoesHere'
  // your code here
});
```

JQUERY — REVIEW

EXERCISE

KEY OBJECTIVE

- Review jQuery selectors and events, get practice looking up new event types

TYPE OF EXERCISE

- Individual/paired

SMALL GROUP PLANNING

8 min

1. Follow the instructions in Starter Code> jQuery Events > js/main.js

SCROLLMATIONS LAB

MOUSE

click

dblclick

mouseenter

mouseleave

KEYBOARD

keypress

keydown

keyup

FORM

submit

change

focus

blur

DOCUMENT

resize

scroll

```
$('li').on('eventGoesHere'
  // your code here
});
```

ACTIVITY

EXERCISE

LOCATION

- starter code > sticky_nav

KEY OBJECTIVE

- Practice thinking through some common JS interactions

TIMING

6 min	1. Review these links: find out how far down the page the nav is and find out how far the user has scrolled.
8 min	2. Write pseudo code in main.js
15 min	3. Follow the instructions in main.css to make the sticky nav functional.
	4. Bonus: Fade in the annotations. Bonus: use setTimeout (Google it) to stagger the animations.

MODAL WINDOW LAB

ACTIVITY

EXERCISE

LOCATION

- starter code > modal_window

KEY OBJECTIVE

- Practice thinking through some common JS interactions

TIMING

20 min

1. Follow the instructions in your main.css file
2. Write your pseudo code for the close and open buttons
3. Write your JS

FORM BASICS

LEARNING OBJECTIVES

- Familiarity with how transitions and transformations can be triggered using JavaScript
- Add event listeners for mouse, keyboard, and document events.
- Practice thinking and working through some common JavaScript interactions.

WEEKLY OVERVIEW

WEEK 7	Interactions / Functions
WEEK 8	JS Lab / Holiday Break

EXIT TICKETS

Bookmark this link!

FEWD 51 Exit Ticket