



INTERESTING THINGS ABOUT CSS

UNDERSTANDING PSEUDO-CLASSES & PSEUDO-ELEMENTS

WHAT ARE PSEUDO-CLASSES?

- Pseudo-classes allow you to apply styles to elements based on their state or user interaction. They start with a colon :

EXAMPLE:

```
/* Change color when a button is hovered */
button:hover {
  background-color: blue;
  color: white;
}

/* Style input when focused */
input:focus {
  border: 2px solid red;
}

/* Change link color when active */
a:active {
  color: green;
}
```



INTERESTING THINGS ABOUT CSS

UNDERSTANDING PSEUDO-CLASSES & PSEUDO-ELEMENTS

WHAT ARE PSEUDO-ELEMENTS?

- Pseudo-elements allow you to style specific parts of an element, such as inserting content before or after it. They start with ::

EXAMPLE:

```
/* Add content before an h1 */  
h1::before {  
  content: "🔥 ";  
  color: orange;  
}  
  
/* Add content after an h1 */  
h1::after {  
  content: " 🚀";  
}
```



INTERESTING THINGS ABOUT CSS

LEARNING CSS TRANSITIONS

WHAT ARE TRANSITIONS?

- CSS transitions allow property changes to occur smoothly over a specified duration instead of happening instantly.

EXAMPLE:

```
/* Add content before an h1 */  
h1::before {  
  content: "🔥 ";  
  color: orange;  
}  
  
/* Add content after an h1 */  
h1::after {  
  content: " 🌈";  
}
```



INTERESTING THINGS ABOUT CSS

LEARNING CSS TRANSITIONS

WHAT ARE TRANSITIONS?

- CSS transitions allow property changes to occur smoothly over a specified duration instead of happening instantly.

EXAMPLE:

```
/* Add content before an h1 */
h1::before {
  content: "🔥 ";
  color: orange;
}

/* Add content after an h1 */
h1::after {
  content: " 🌈";
}
```

TRANSITION PROPERTIES

- transition-property: Defines which property to animate (e.g., color, background, all)
- transition-duration: Specifies the time the transition takes (e.g., 0.5s, 1s)
- transition-timing-function: Defines the speed curve (ease, linear, ease-



INTERESTING THINGS ABOUT CSS

UNDERSTANDING CSS TRANSFORMATIONS

The transform property allows you to modify the appearance of an element by moving, rotating, scaling, or skewing it.

Transform Functions

- `translate(x, y)`: Moves the element
- `rotate(deg)`: Rotates the element
- `scale(x, y)`: Enlarges or shrinks the element
- `skew(x, y)`: Skews the element along the X and Y axis

EXAMPLE:

```
/* Move element 50px right and 20px down */
.box {
  transform: translate(50px, 20px);
}

/* Rotate element by 45 degrees on hover */
.box:hover {
  transform: rotate(45deg);
}

/* Scale element 1.5 times when active */
.box:active {
  transform: scale(1.5);
}
```



INTERESTING THINGS ABOUT CSS

UNDERSTANDING CSS ANIMATIONS

What are Animations?

CSS animations allow elements to change styles over time without user interaction.

Animation Properties

- `@keyframes`: Defines the steps of the animation
- `animation-name`: Specifies the animation name
- `animation-duration`: Defines how long the animation lasts
- `animation-timing-function`: Controls the speed curve
- `animation-iteration-count`: Specifies how many times the animation runs (infinite, 1, 2, etc.)

EXAMPLE:

```
/* Define animation */
@keyframes slide {
  0% { transform: translateX(0); }
  50% { transform: translateX(100px); }
  100% { transform: translateX(0); }
}

/* Apply animation */
.box {
  animation: slide 2s infinite ease-in-out;
}
```



PROJECT EXERCISE : STYLE YOUR FORM!

UNDERSTANDING CSS ANIMATIONS

Description:

Make your form look good with CSS! Add colors, rounded corners, and make it interactive. Learn to style different parts of the form when someone uses it (like when they hover or click).

Hints:

- **Pseudo-classes (:hover, :focus, :active)**
 - Change how things look when you point at them (hover), click in them (focus), or click on them (active). Think about button colors or input field borders.
- **CSS Transitions:**
 - Make style changes smooth, not instant. Use transition to control how styles change over time.
- **Transforms (2D):**
 - Move, rotate, or scale things! Try making your button slightly bigger or move a bit when you hover using transform.
- **Transforms (3D - Challenge):**
 - Make things look like they have depth. This is harder! Look at perspective and rotateX to make elements tilt. (Optional!)
- **CSS Animation (Challenge):**
 - Make things move on their own! Use @keyframes to create simple moving effects, like a pulsing button. (Optional!)