School of Infocomm

C219 Front-end Web Development

Lesson 4

Using Advanced CSS to Enhance Web Pages

START

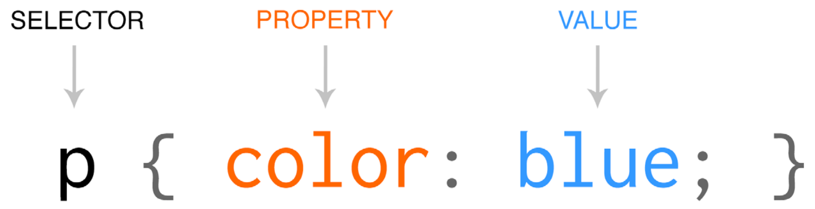
{ What is CSS? }
And How It Works!
{;}
CSS
ATE A PRO
EBSITE

What is CSS?

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

CSS handles the look and feel of a web page, including colours, fonts, spacing, layout and much more.

CSS syntax:



SELECTOR

PROPERTY

VALUE

```
p { color: blue; }
```

Advantages of CSS

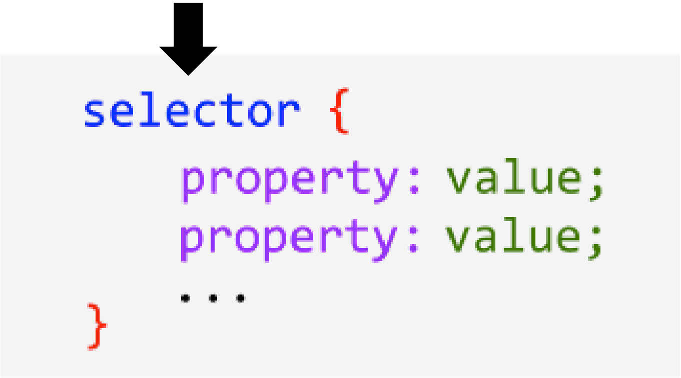
- CSS saves time – You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.
- Pages load faster – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- Easy maintenance – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility – Style sheets allow content to be optimised for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices.
- Global web standards – Old HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CSS Selectors

CSS selectors are used to select the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Basic selectors
- Combinator selectors
- Pseudo-class selectors
- Pseudo-elements selectors
- Attribute selectors



```
selector {
    property: value;
    property: value;
    ...
}
```

Basic Selectors

Basic CSS selectors select elements based on name, id and class. CSS selectors can be grouped.

Selector Click on the links to try the examples	Example	Example description
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
element.class	p.title	Selects only <p> elements with class="title"
*	*	Selects all elements
element	p	Selects all <p> elements
element, element, ...	div, p	Selects all <div> elements and all <p> elements

Combinator Selectors

A CSS selector can contain more than one selector. Between the selectors, we can include a combinator. There are four different combinators in CSS:

Selector Click on the links to try the examples	Example	Description
`element element`	`div p`	Descendant selector (space): Selects all <p> elements inside <div> elements
`element > element`	`div > p`	Child selector (>): Selects all <p> elements where the parent is a <div> element
`element + element`	`div + p`	Adjacent sibling selector (+): Selects the first <p> element that are placed immediately after <div> elements
`element1 ~ element2`	`p ~ ul`	General sibling selector (~): Selects every <ul> element that are preceded by a <p> element

Pseudo-class Selectors

A pseudo-class is used to define a special state of an element. Here are some common ones:

Selector — Click on the links to try the examples	Example	Example description
`:active`	`a:active`	Selects the active link
`:hover`	`a:hover`	Selects links on mouse over
`:visited`	`a:visited`	Selects all visited links
`:focus`	`input:focus`	Selects the <input> element that has focus
`:nth-child(n)`	`p:nth-child(2)`	Selects every <p> element that is the second child of its parent
`:last-child`	`p:last-child`	Selects every <p> elements that is the last child of its parent

View all CSS Pseudo Classes here *(scroll to the bottom of the page)*

Pseudo-elements Selectors

A CSS pseudo-element is used to style specified parts of an element.

Selector Click on the links to try the examples	Example	Example description
`::after`	`p::after`	Insert something after the content of each <p> element
`::before`	`p::before`	Insert something before the content of each <p> element
`::first-letter`	`p::first-letter`	Selects the first letter of each <p> element
`::first-line`	`p::first-line`	Selects the first line of each <p> element
`::marker`	`::marker`	Selects the markers of list items
`::selection`	`p::selection`	Selects the portion of an element that is selected by a user

Attribute Selectors

The attribute selector is used to select elements with a specified attribute.

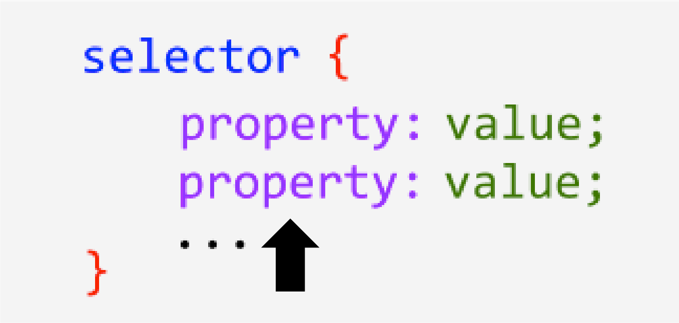
Selector Click on the links to try the examples	Example	Example description
[attribute]	[target]	Selects all elements with a target attribute
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"

View all CSS Attribute Selectors here *(scroll to the bottom of the page)*

CSS Properties

The CSS selector determines what HTML elements to target. CSS properties specify what to style of the targeted HTML elements.

Each property has a set of possible values. Some properties can affect any type of element, while others apply only to particular groups of elements.



```
selector {
    property: value;
    property: value;
    ...
}
```

```
h1 {
   background: #ff3399; //applies a background colour
   margin: 0 10px; //top and bottom margin of 0px, left and right margin of 10px
}
```

Advanced CSS Properties

There are 541 distinct CSS property names. The basic properties like color, background, font-size and margin have already been covered in the past.

Today you will learn some advanced CSS properties like:

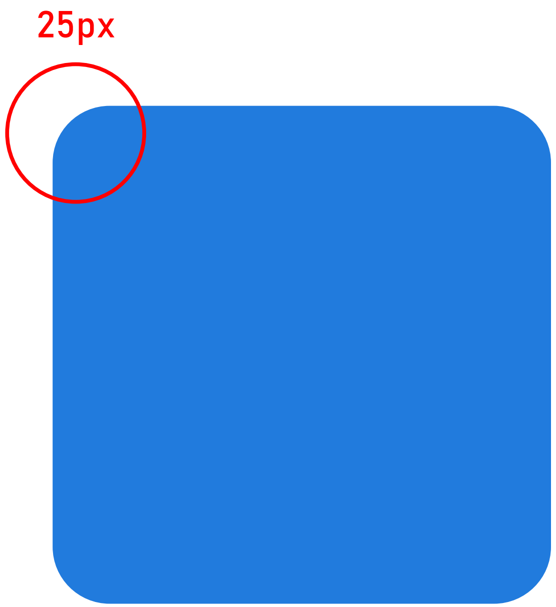
- Gradients
- Shadows
- Transitions
- Animations
- Filters

CSS Rounded Corners

The CSS border-radius property defines the radius of an element's corners. You can apply to elements with a background colour, on borders, and on images. The border-radius property can have up to four values.

```
.roundbox {
   border-radius: 25px; //or
   border-radius: 25px 25px 25px 25px;
}
```

TRY IT YOURSELF



CSS Opacity

The opacity of an element can be set using the RGBA value or opacity property.

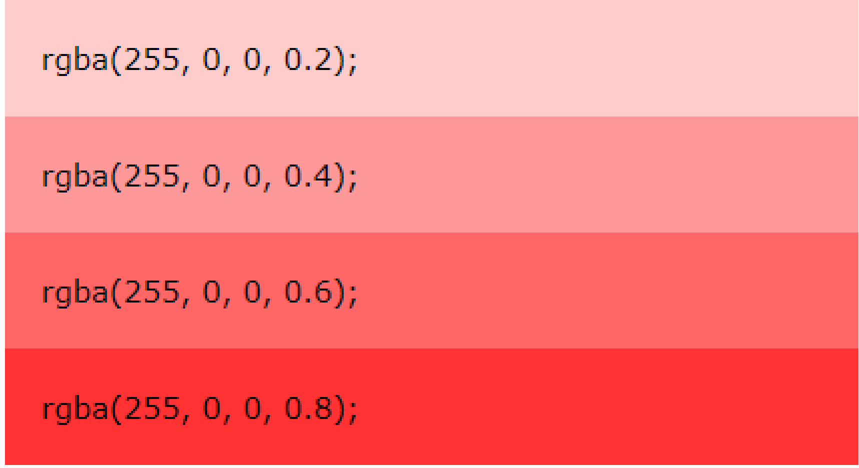
USING RGBA

```
#box {
    background-color: rgba(255, 0, 0, 0.3); //red background with 30% opacity
}
```

USING OPACITY

```
#box {
    background-color: red; //red background
    opacity: 0.3; //30% opacity (applied to #box and its contents)
}
```

TRY IT YOURSELF


rgba(255, 0, 0, 0.2);

rgba(255, 0, 0, 0.4);

rgba(255, 0, 0, 0.6);

rgba(255, 0, 0, 0.8);

CSS Gradients

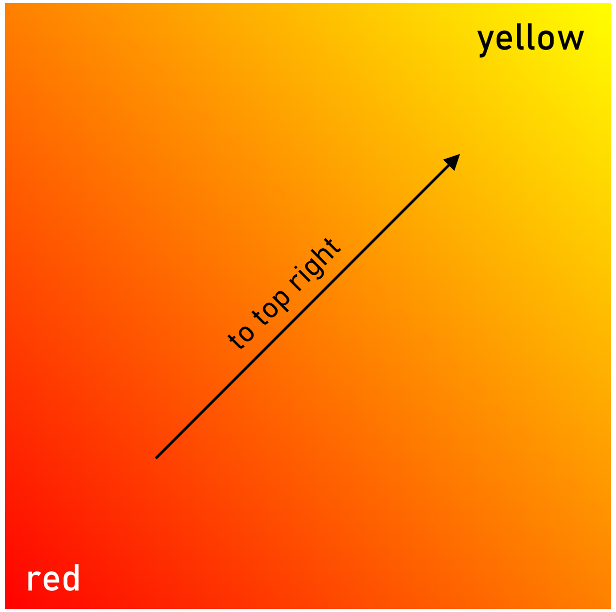
CSS gradients let you display smooth transitions between two or more specified colours.

CSS defines two types of gradients:

- Linear gradients (up/down/left/right/diagonal)
- Radial gradients (defined by their center)

```
#box {
   background: linear-gradient(to top right, red, yellow)
   //red to yellow diagonal gradient
}
```

TRY IT YOURSELF



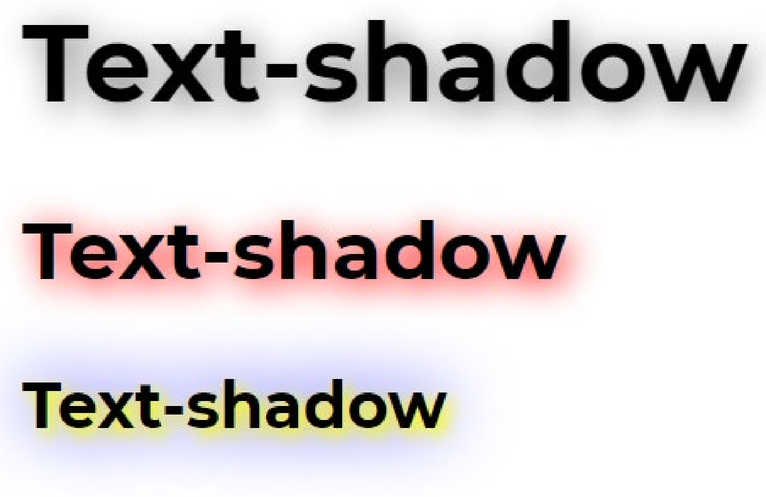
CSS Text-shadow

The text-shadow property adds shadow to text.

This property accepts a comma-separated list of multiple shadows to be applied to the text.

```
h1 {
    text-shadow: 2px 2px 10px grey;
    //syntax: offset-x | offset-y | blur-radius | colour
}
```

TRY IT YOURSELF

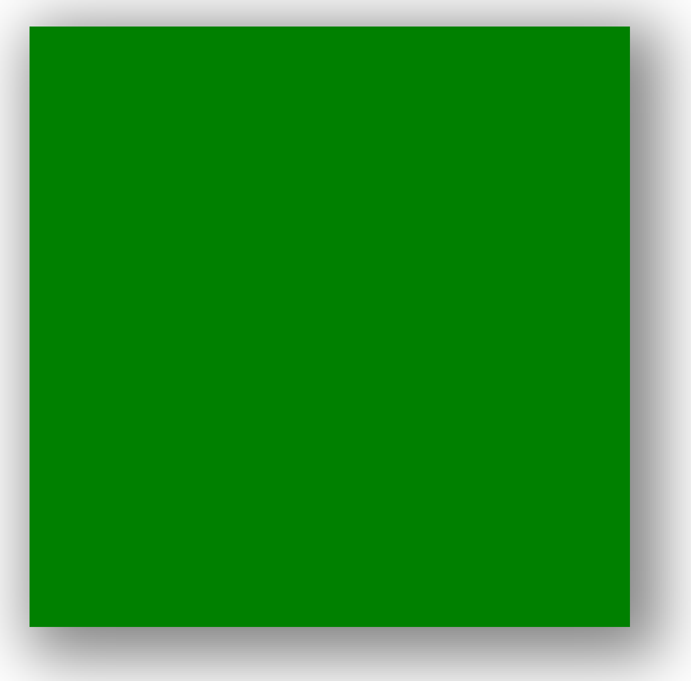


CSS Box-shadow

The box-shadow property attaches one or more shadows to an element. The syntax is similar to text-shadow, except that it can contain two more values: spread and inset.

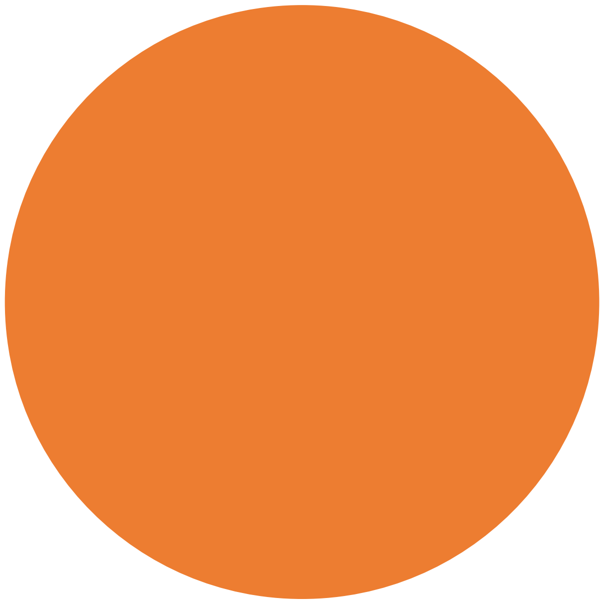
```
div {
   box-shadow: 5px 5px 20px 5px grey;
   //syntax: offset-x | offset-y | blur-radius | spread | colour | inset
}
```

TRY IT YOURSELF



CSS Transitions

CSS transitions provide a way to control animation when changing CSS properties. CSS transitions let you decide which properties to animate (by listing them explicitly), when the animation will start (by setting a delay), how long the transition will last (by setting a duration), and how the transition will run (by defining a timing function, e.g. linearly or quick at the beginning, slow at the end).



CSS Transitions

To create a transition effect, you can specify the CSS property you want to add an effect to, and the duration of the effect. You can add more than one CSS property.

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}
```

Specify a new value for the width property when a user hovers over the <div> element:

```
div:hover {
  width: 300px;
}
```

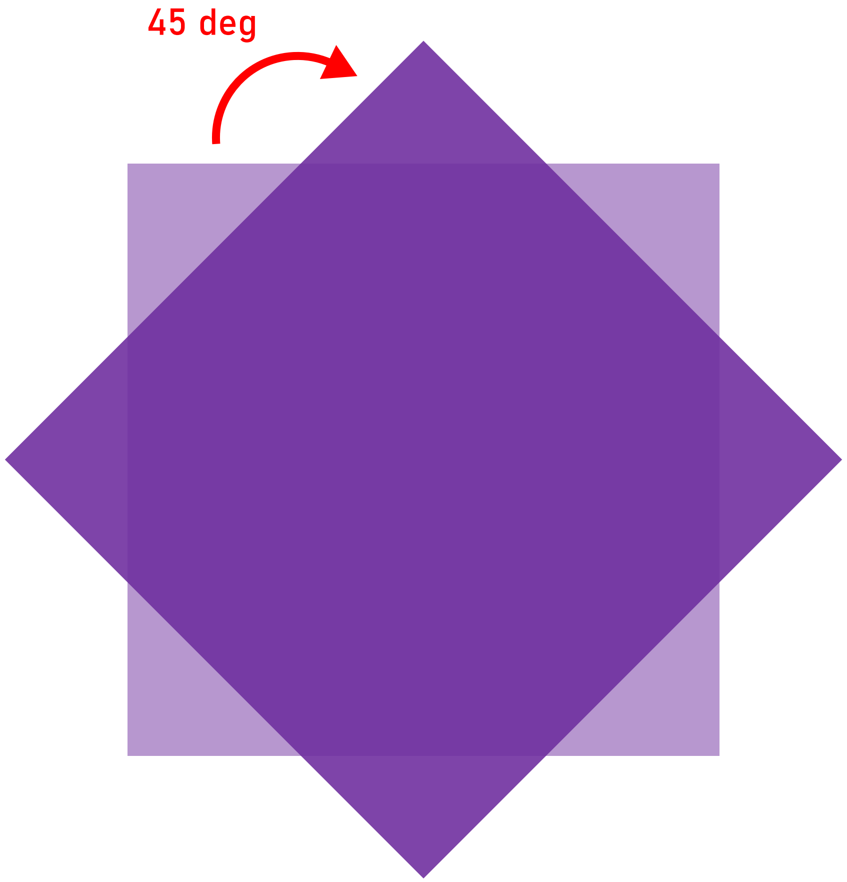
TRY IT YOURSELF

CSS Transforms

CSS transforms allow you to move, rotate, scale, and skew elements. With the CSS transform property you can use 2D transformation methods like translate(), rotate(), scale() and skew().

```
div {
    transform: rotate(45deg);
    //syntax: transformation | angle / width / height / axes
}
```

TRY IT YOURSELF

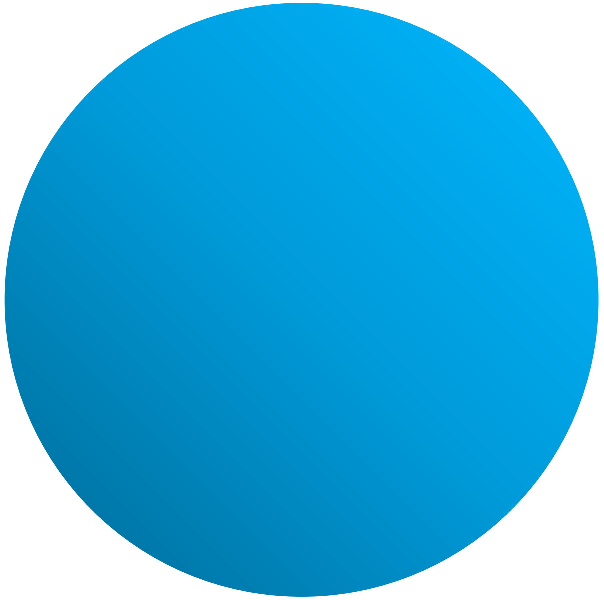


CSS Animations

A CSS animation lets an element gradually change from one style to another. You can change as many CSS properties you want, as many times as you want.

```
div {
   animation: example 5s linear 2s infinite;
   //syntax: animation name | duration | timing function | delay | iteration
}
```

TRY IT YOURSELF



Read more about CSS animations here.

CSS Filters

The filter CSS property applies graphical effects like blur or colour shift to an element. Filters are commonly used to adjust the rendering of images, backgrounds, and borders.

```
img {
    filter: grayscale(100%);
    //syntax: filter name | value
}
```

TRY IT YOURSELF

Read more about CSS filters here.

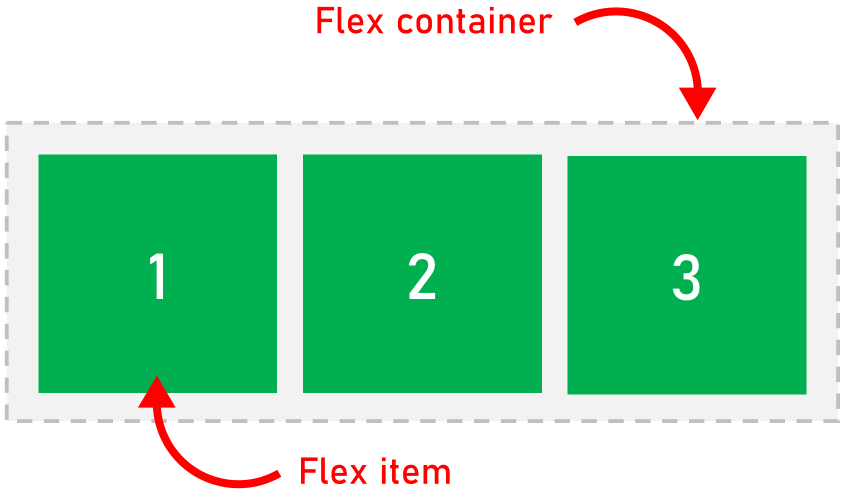
CSS Flexbox

The CSS Flexbox aims at providing a more efficient way to lay out, align and distribute space among items in a container.

The main idea behind the flex layout is to give the container the ability to alter its items' width, height and order) to best fill the available space. A flex container expands items to fill available free space or shrinks them to prevent overflow.

Try out all the exercises in the 4 chapters of CSS Flexbox by clicking the button below.

TRY IT YOURSELF

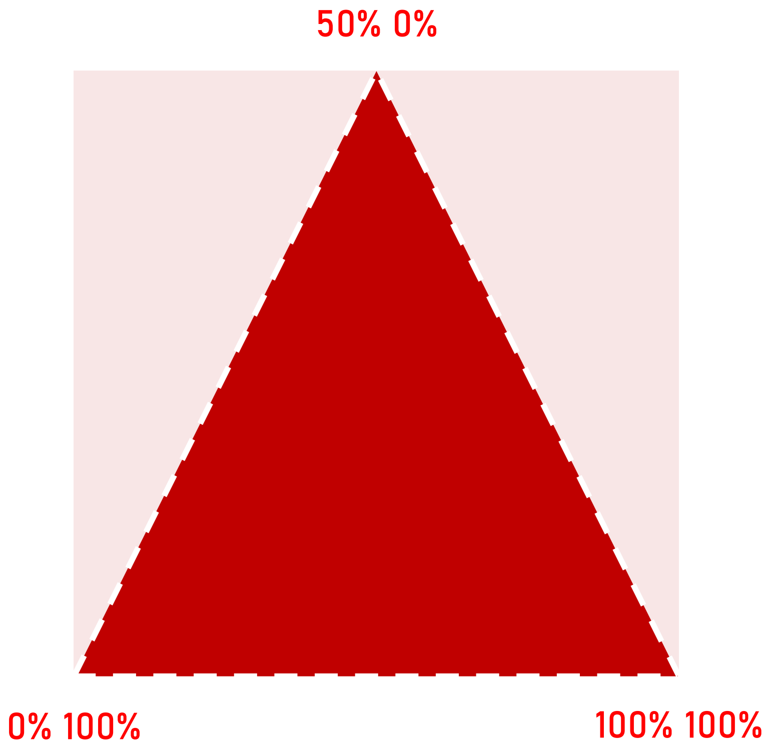


CSS Clip-path

The clip-path property in CSS allows you to specify a specific region of an element to display, with the rest being hidden.

```
.triangle {
   background: crimson;
   width: 250px;
   height: 200px;
   clip-path: polygon(50% 0%, 100% 100%, 0% 100%);
}
```

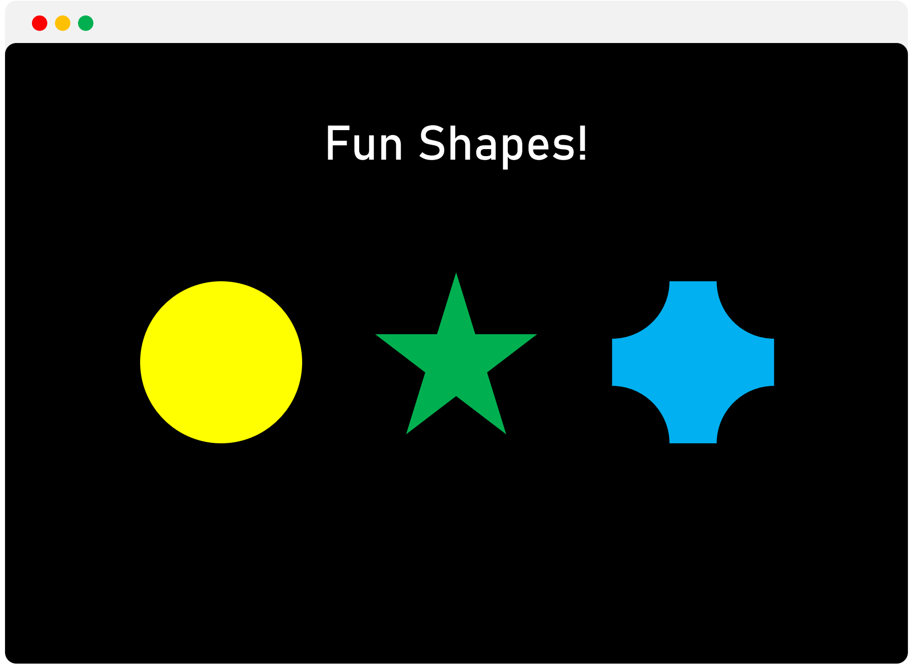
TRY IT YOURSELF



Teamwork Assignment

Create a web page consisting of different CSS shapes that perform animations on hover. Here are the requirements:

- Title of web page is Fun Shapes!
- Each student must create a different shape using CSS only
- The animations must be a combination of a colour and a transformation change on hover (e.g. when you hover over a yellow circle, it changes to a larger blue square)
- Smooth transitions must be applied
- Compile all shapes into one web page



Go Further! Center the group of shapes vertically and horizontally in the page.

Quiz

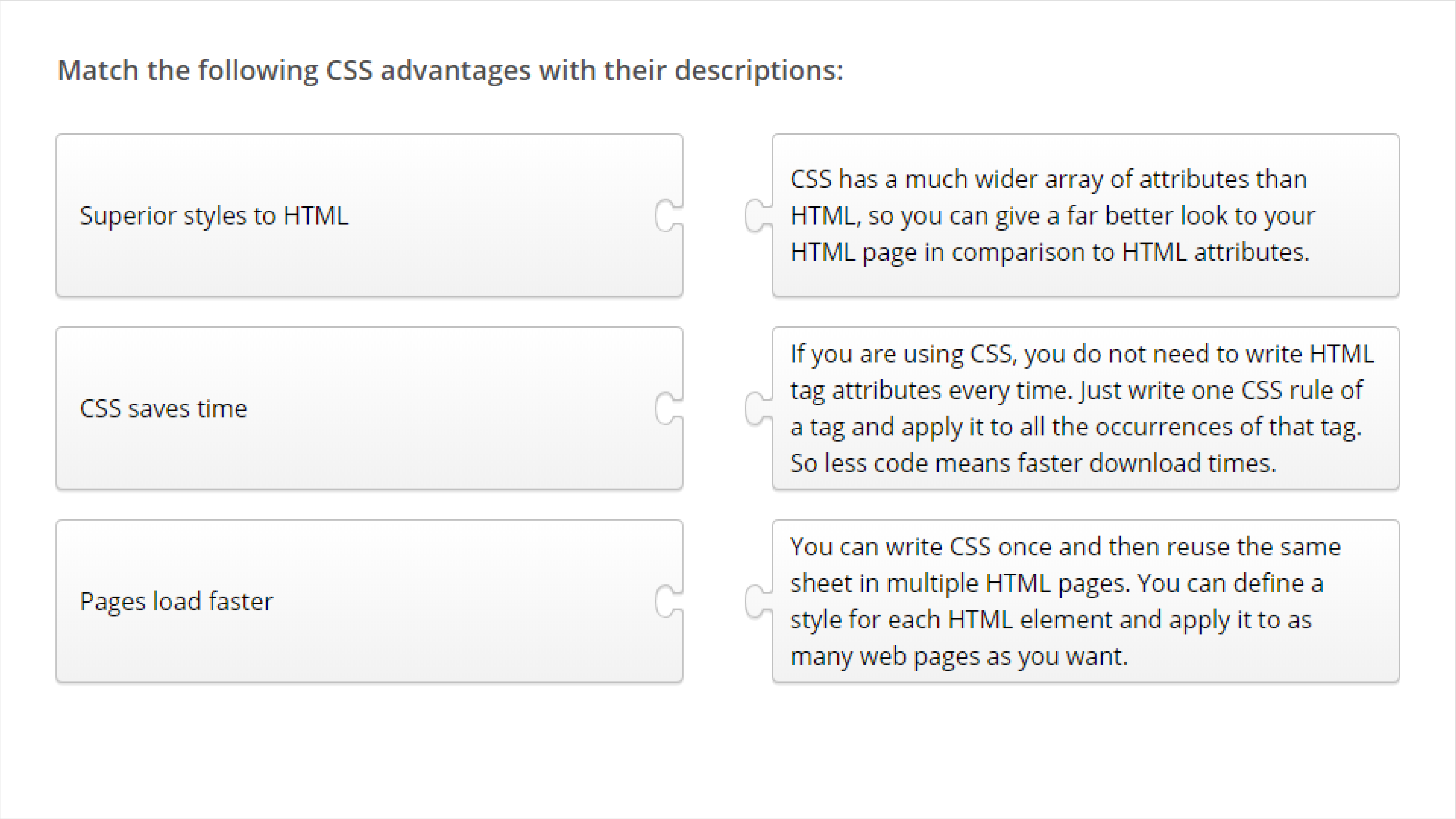
Test your knowledge on CSS!

START

You have unlimited attempts.
You are encouraged to score full marks before proceeding.

☑ Quiz

Click the **Quiz** button to edit this object



Match the following CSS advantages with their descriptions:

Superior styles to HTML

CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

CSS saves time

If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Pages load faster

You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.

L04 Assignment

Enhance your World Xplore website.

Requirements:

a. Animate your logo on hover, using CSS animation and transform(s).

b. Implement styles to your navigation to indicate the current page, and also hover effects.

c. Using only CSS and HTML, implement a drop down menu for Package Tours containing three menu items.

d. Create a zoom, blur, grayscale or flip effect for your article images on hover

e. Code two more pages for the World Xplore website

Deliverables

- Team submission [1]
  - One web page
- Individual submission
  - World Xplore website [2]
  - Zip everything and submit

Submit all deliverables by 2359 today

[1] To be submitted by team leader only. Indicate the student names at the bottom of the web page for the shape that they did.

[2] Home page and two other pages (you can choose any two pages)