CSC4130
Introduction to Human-Computer Interaction

Lecture 7

User Interface Technology: JavaScript

香 港 中 文 大 學 (深 圳)
The Chinese University of Hong Kong, Shenzhen

Outline

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

- JavaScript
- SVG and Canvas
- Interaction

Outline

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

- JavaScript
- SVG and Canvas
- Interaction

Javascript

- One of the core technologies of the World Wide Web
- Create dynamic and interactive web-based applications and systems
- Simple and easy to learn

Javascript variables

- Create variables
  - var x (global variable)
  - let x (local variable)
  - const x = 6
- Use variables
  - x = 6
  - y = x+6
- Print variables
  - console.log(x)

Javascript data types

- Arrays
  - var c = [0,1,2];
  - var m = [0,4.5,"HCI"];
  - m.push(67);
- Objects
  - obj = {key1:3, key2:4};
  - obj[key1] or obj.key1
  - obj.key3 = "new value"

Javascript functions

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

```
let x = myFunction(4, 3);   // Function is called, return value will end up in x

function myFunction(a, b) {
  return a * b;             // Function returns the product of a and b
}
```

Javascript loop

- For loop
- While loop
- Do loop

```
for (i=0; i<10; ++i) {
    console.log(i);
}
```

```
i = 3;
while (i<100) {
    console.log(i);
    i = i * 2;
}
```

```
i = 3;
do {
            console.log(i);
            i = i * 2;
} while (i<100);
```

Javascript switch

```
i = "some case";
switch (i) {
case "string literals ok":
    console.log("Yes");
    break;
case "some case":
    console.log("Unlike C");
    break;
}
```

Javascript inheritance

- JavaScript *does* support a notion of inheritance, but it does it without any classes. This means that there's no subclasses
- Instead of subclasses, JavaScript has the notion of a *prototype chain*. Every JavaScript object has a special field which points to another object. Then, every time you tell JavaScript to access a field from an object, it tries to find the field. If the field exists, then the lookup is performed. If, however, the field doesn't exist, then JavaScript checks for the presence of a special *prototype* field in the object. If that field is not null, then the JavaScript runtime performs a recursive access of the field in the prototype object.

Javascript inheritance

```
// Inheritance, with no classes
base = {
    v1: 1,
    v2: 2
};

derived = {
    v1: 5,
    v3: 3,
    v4: 4
};

console.log(base.v1);
console.log(derived.v1);
console.log(derived.v2);

// this calls sets the prototype of derived to be the base
Object.setPrototypeOf(derived, base);
console.log(derived.v1);
console.log(derived.v2);
```

Javascript inheritance

```
// Instead of using setPrototypeOf, use:
v = Object.create(null); // this is just the same as {}
v2 = Object.create(base);
v3 = Object.create(v2); // etc.
```

Javascript this variable

- JavaScript has a special variable that is available at every scope called *this*. When a function is called with a notation that resembles methods in typical object-oriented languages, say obj.method(), then *this* is bound to the object holding the method (in this case obj). *this* allows you to make changes to the local object

Javascript this variable

```
function otherObject(value)
{
    return {
        x: value,
        get: function() {
            return this.x;
        },
        set: function(newValue) {
            this.x = newValue;
        }
    };
}
```

Javascript practices

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Write a procedure that takes an array as a parameter, iterates over every object in that array, and prints the value of the field “foo” to the console

```
var objects = [{foo: 3, bar: "abc"},
               {foo: 5, bar: "def"}];

function printobject(array){
    for (var i=0;i<array.length;i++){
        console.log(objects[I].foo);
    }
}
```

Javascript practices

香 港 中 文 大 學 (深 圳)
The Chinese University of Hong Kong, Shenzhen

- Write a procedure that takes an array as a parameter, iterates over every object in that array and returns a new array with all the values of the field “foo”

```
function printobject(array){
    let new_array = [];
    for (var i=0;i<array.length;i++){
        new_array.push(array[I].foo);
    }
    return new_array;
}
```

Javascript practices

- Write a procedure keys that takes an object and returns a list of its keys

Javascript practices

- Write a procedure pairs that takes an object and returns a list of lists

Javascript practices

- Write a procedure to convert Celsius to Fahrenheit. (0 degrees Celsius is 32 degrees Fahrenheit; 100 degrees Celsius is 212 degrees Fahrenheit)

Outline

- JavaScript
- SVG and Canvas
- Interaction

Scalable vector graphics (SVG)

- A procedure-based way for drawing graphics content
- “Vector” graphics refers to graphical systems that tare specified independent of coordinates, and can thus be draw and zoomed with no artifacts
- Compare with “Raster” graphics (include typical image formats like .jpg and .png) that just specify an array of pixels

SVG basics

- In html, one encodes the instructions directly with the svg

```
<svg width="…" height="…">
… instructions…
</svg>
```

- Instructions provide commands draw many simple shapes (circles, ellipses, rectangles, lines, path, text, …) included as a set of tags (called nodes or elements)
- Each type of node has a different set of key defining attributes (e.g., a circle must define it's center position (cx,cy) and radius (r)

SVG style

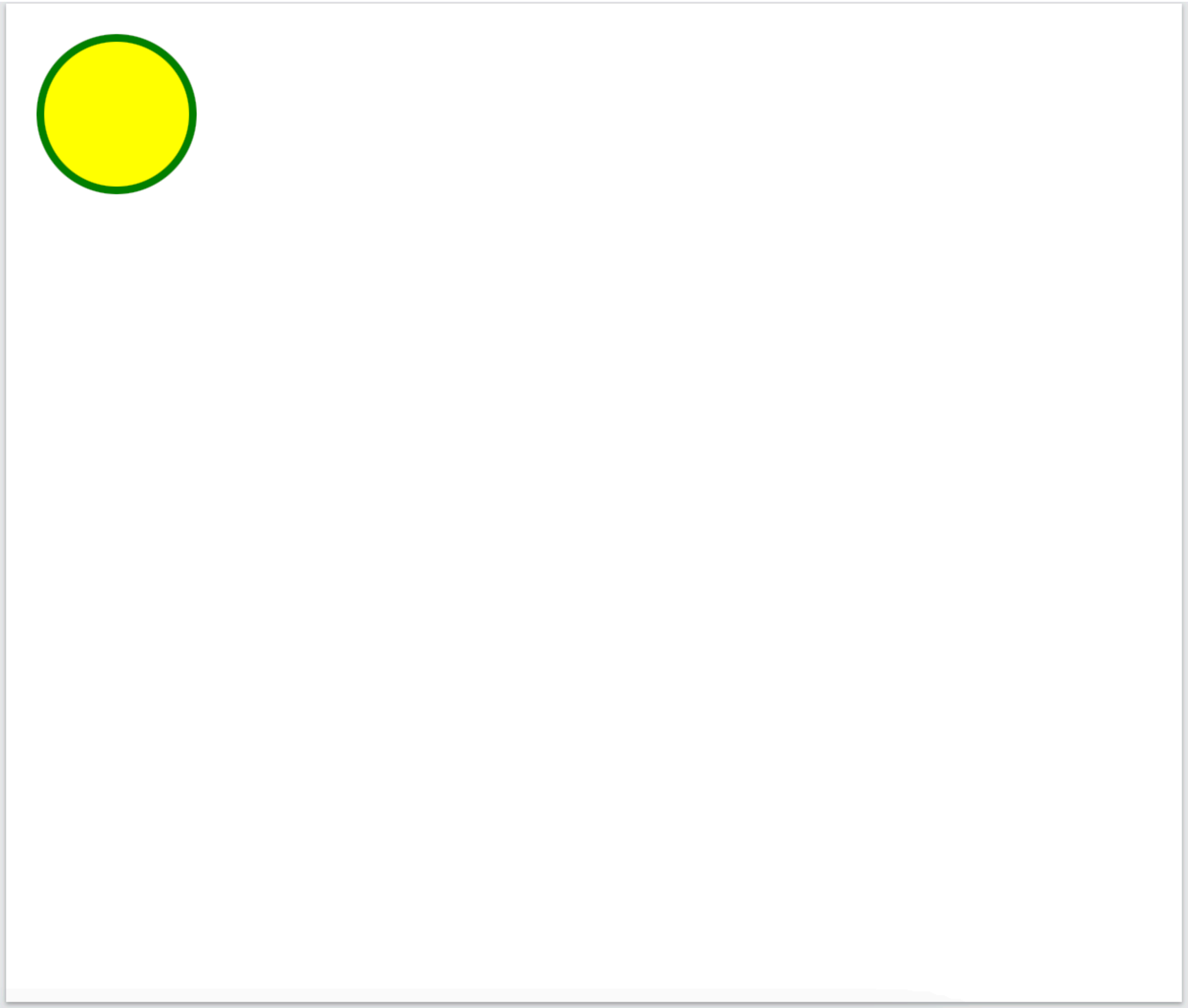
- Can apply style (like with CSS type styles), but many of properties have different names from the usual ones for HTML tags
- SVG refers to these as presentation attributes and a frequent point of confusion since they overload the CSS style syntax
- See https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute#presentation_attributes

Drawing with SVG

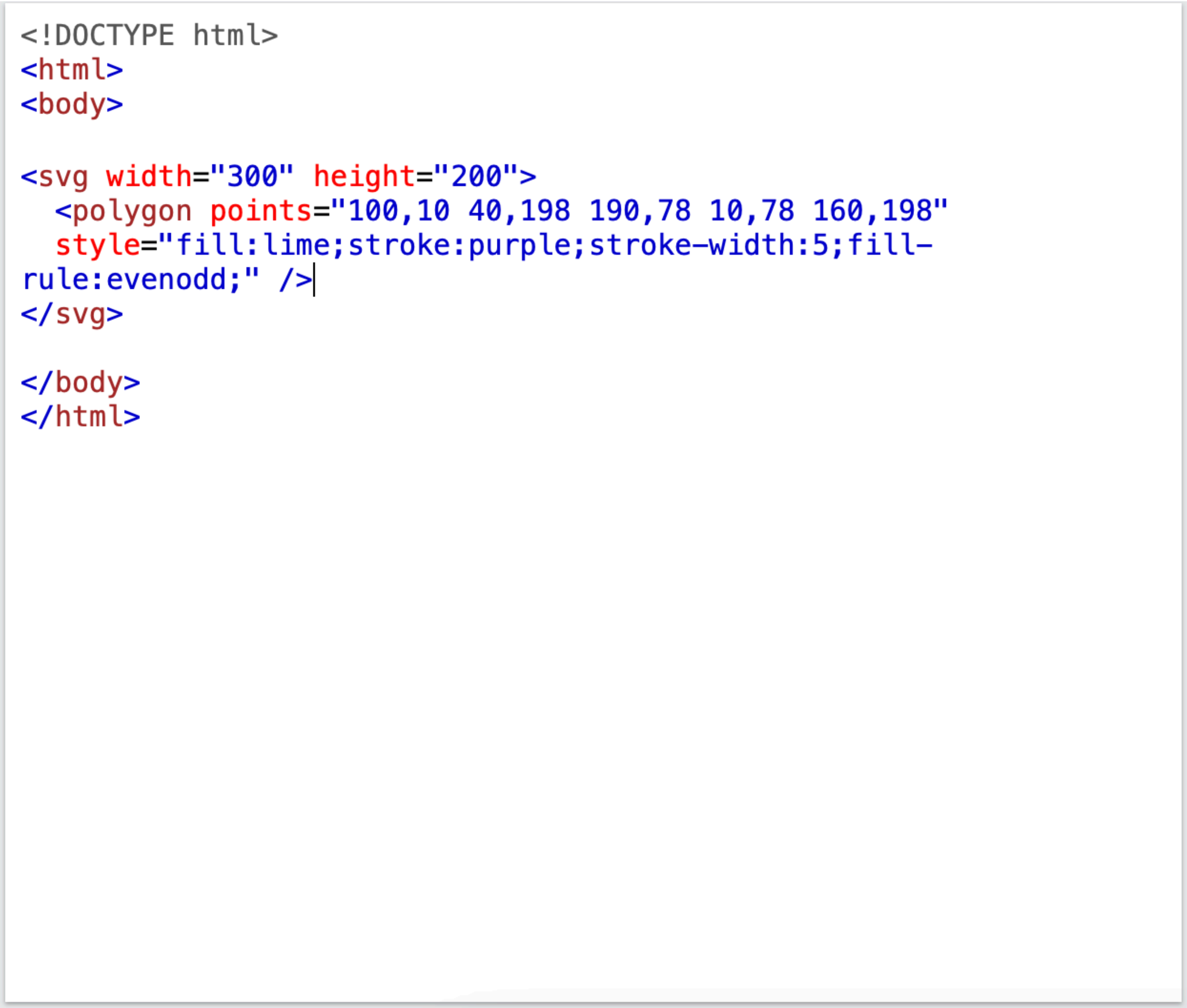
- Instruction are applied one-by-one, and new tags are drawn on top of existing ones
- Use a two-dimensional coordinate system yo specify most drawing
  - Note that top-left corner is (0,0)
- Can apply various transformations using the `transform` attribute, this is particular useful if one groups elements using the svg group node <g></g>

Drawing with SVG

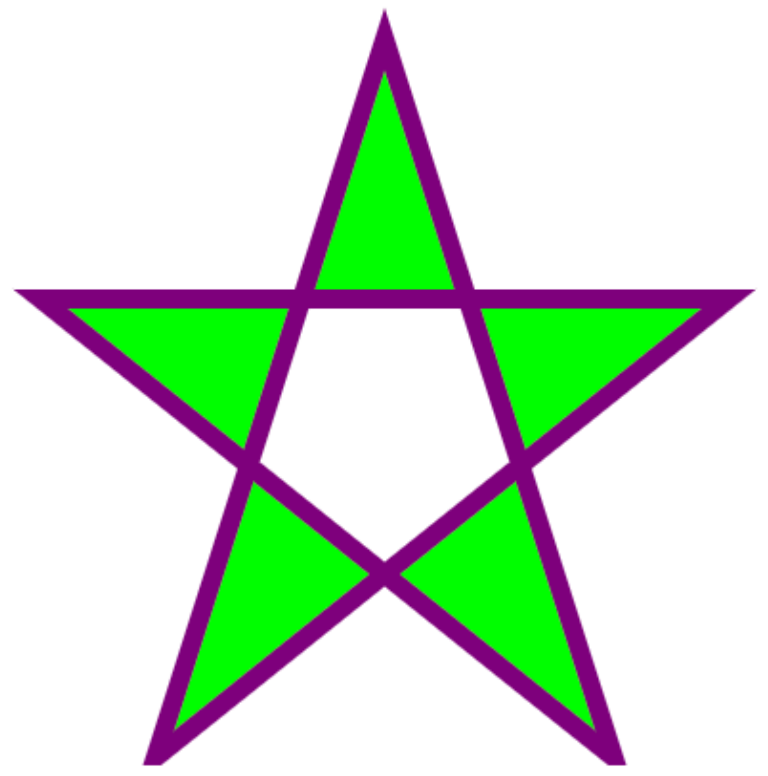
```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
  stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```



Drawing with SVG

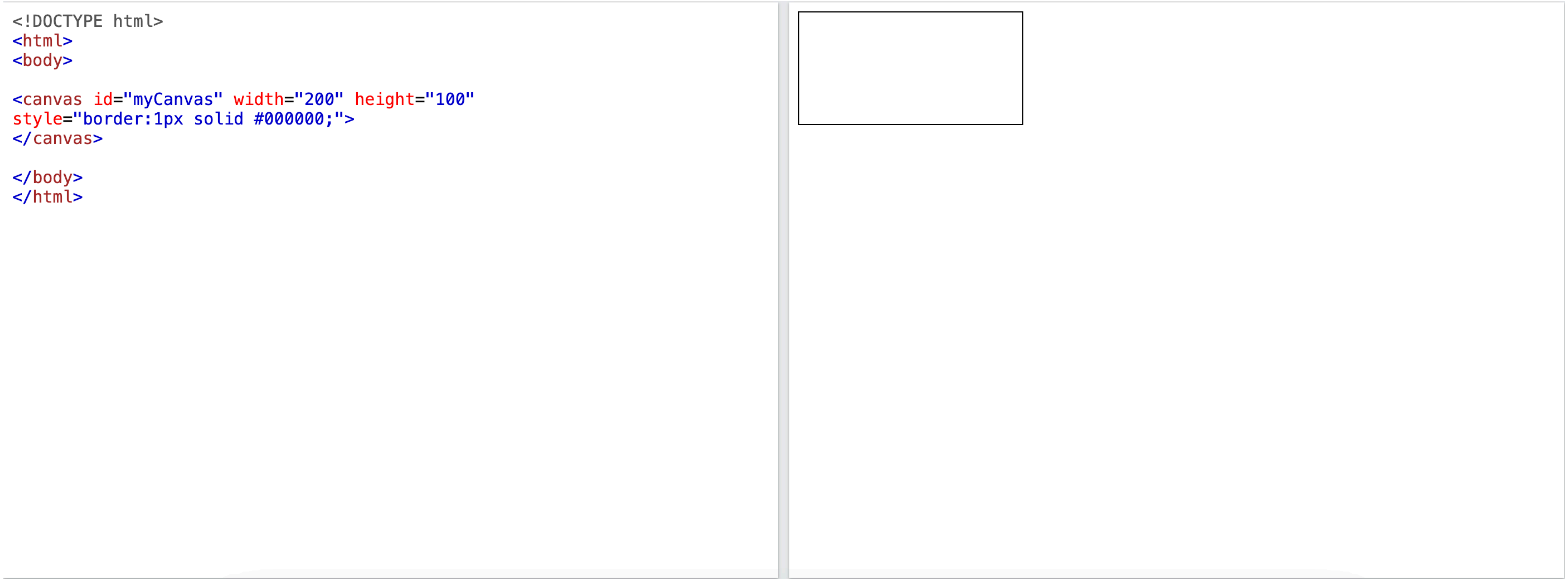


```
<!DOCTYPE html>
<html>
<body>

<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-
rule:evenodd;" />
</svg>

</body>
</html>
```



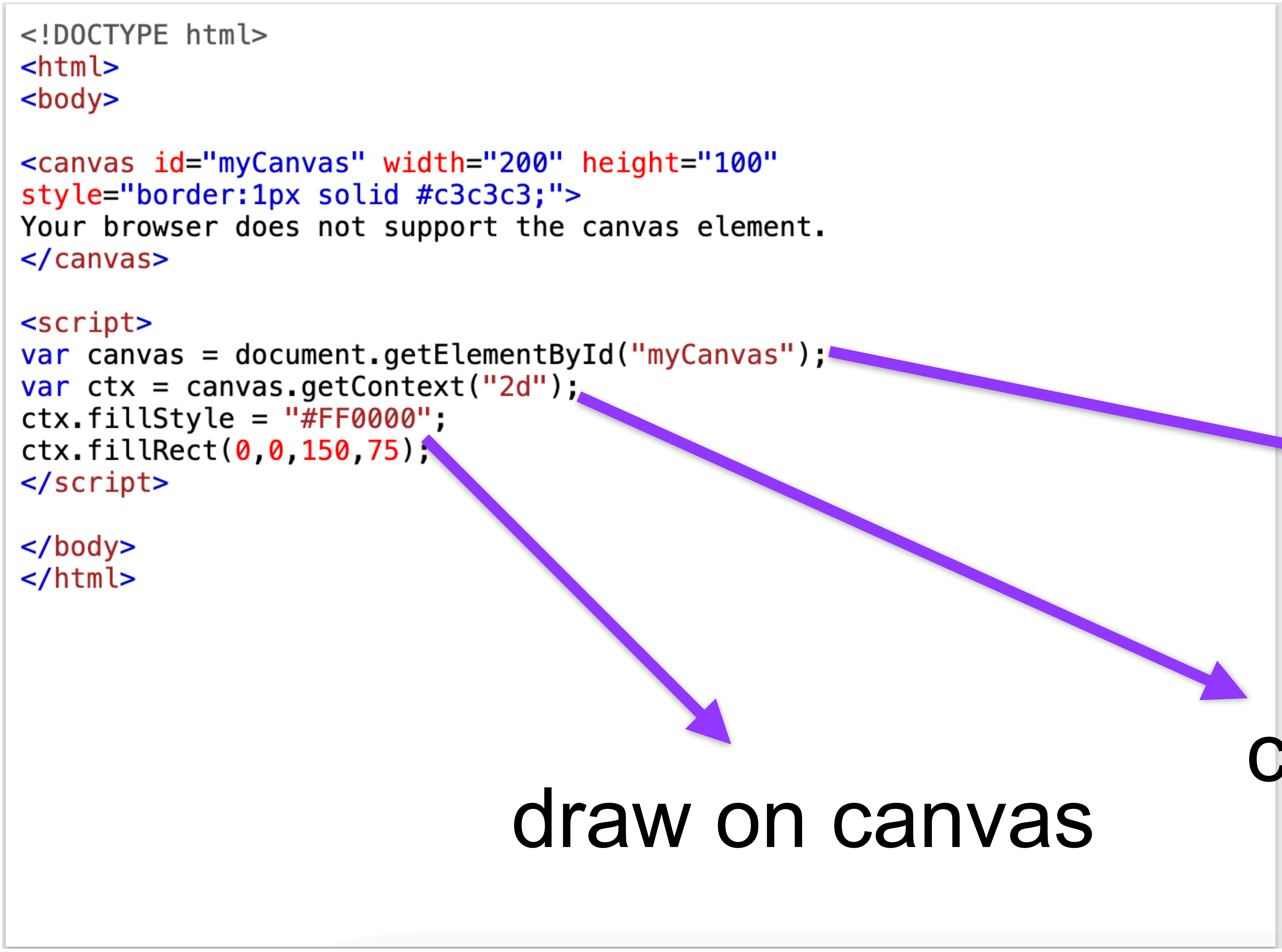
Canvas

- Used for draw graphics
- Only a container and must use a script to draw graphics

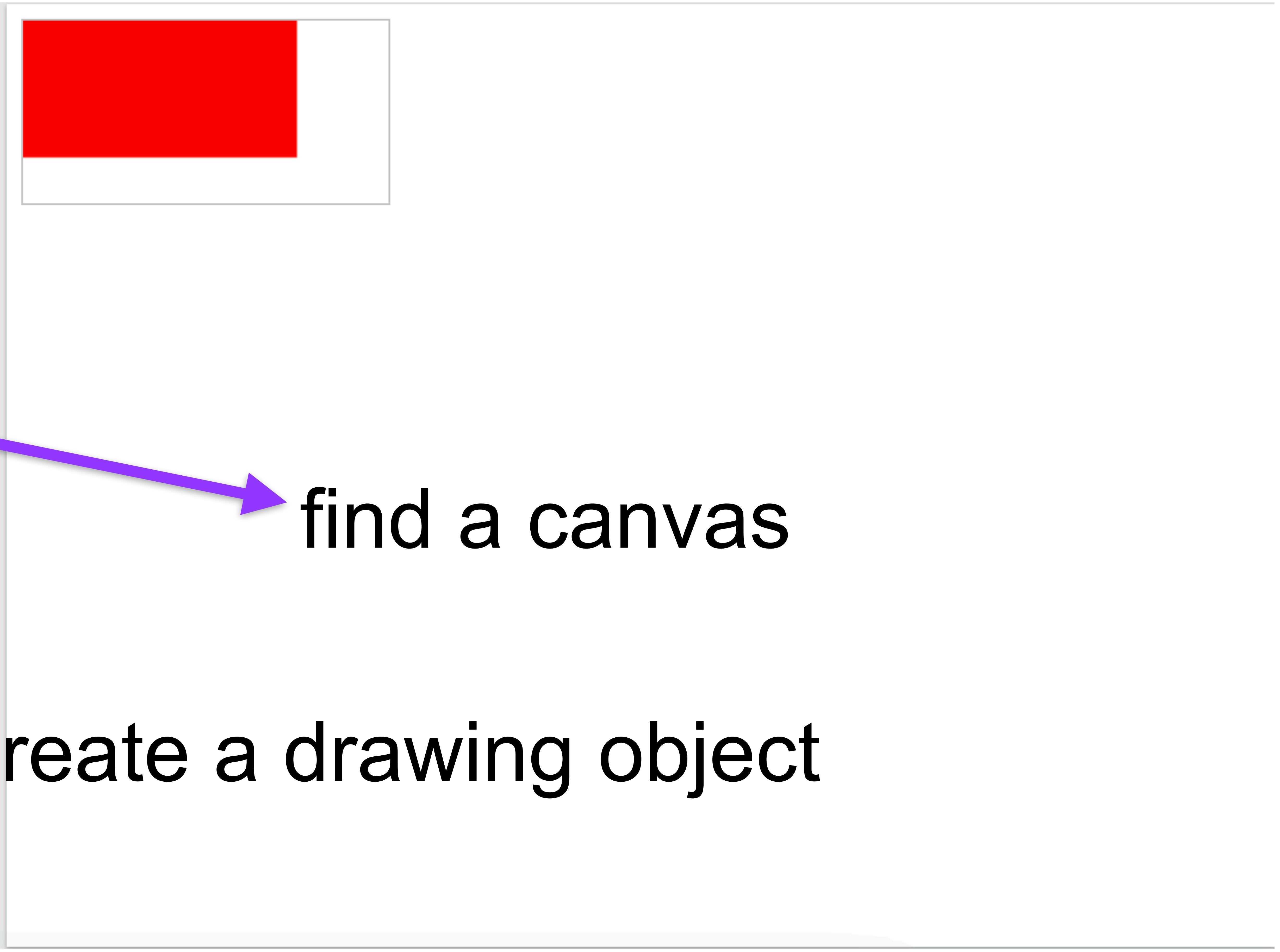


```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
</canvas>

</body>
</html>
```

Drawing with canvas



```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>

<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
</script>

</body>
</html>
```



find a canvas

create a drawing object

draw on canvas

Outline

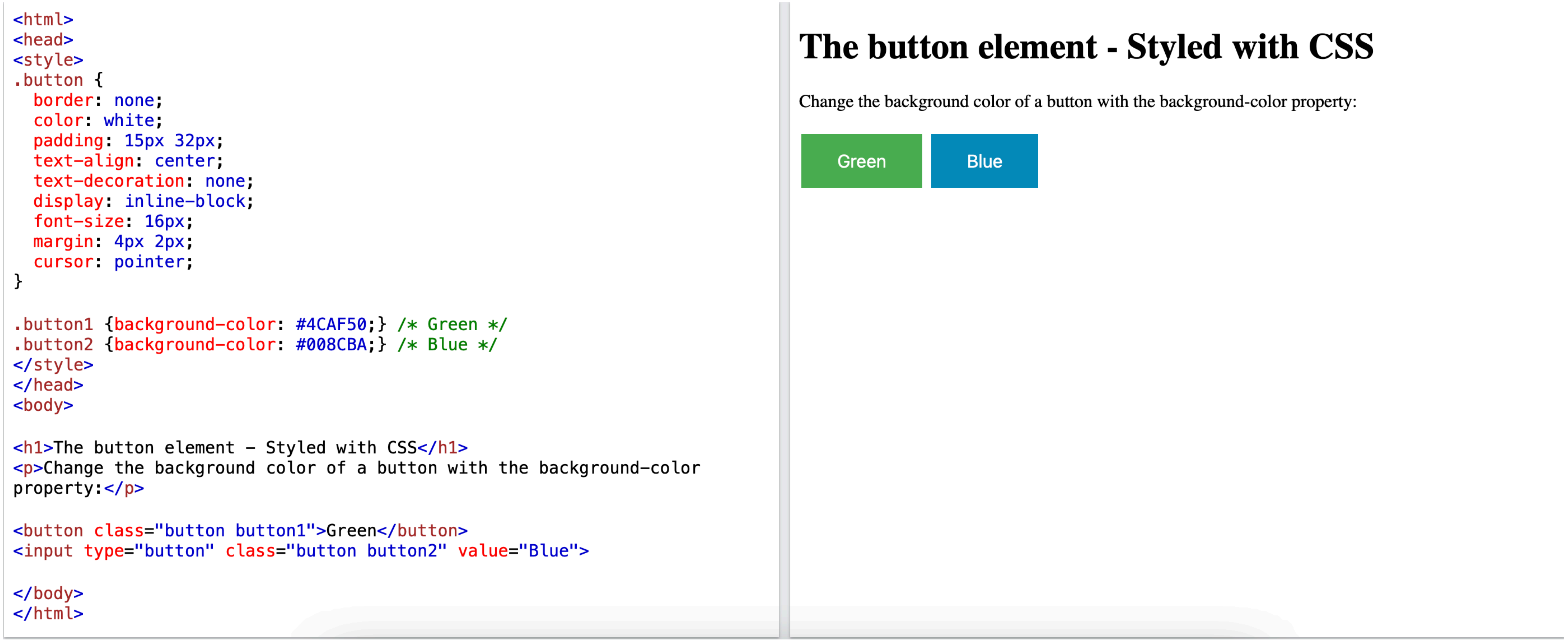
- JavaScript
- SVG and Canvas
- Interaction

Interaction with JavaScript and HTML

- Interaction tags
  - Button
  - Text
  - Radio
  - Option
  - ect.

Bar

- Syntax
  - <button>content</button>
  - <input type=‘button’ value=“content”> </input>



```
<html>
<head>
<style>
.button {
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}

.button1 {background-color: #4CAF50;} /* Green */
.button2 {background-color: #008CBA;} /* Blue */
</style>
</head>
<body>

<h1>The button element - Styled with CSS</h1>
<p>Change the background color of a button with the background-color
property:</p>

<button class="button button1">Green</button>
<input type="button" class="button button2" value="Blue">

</body>
</html>
```

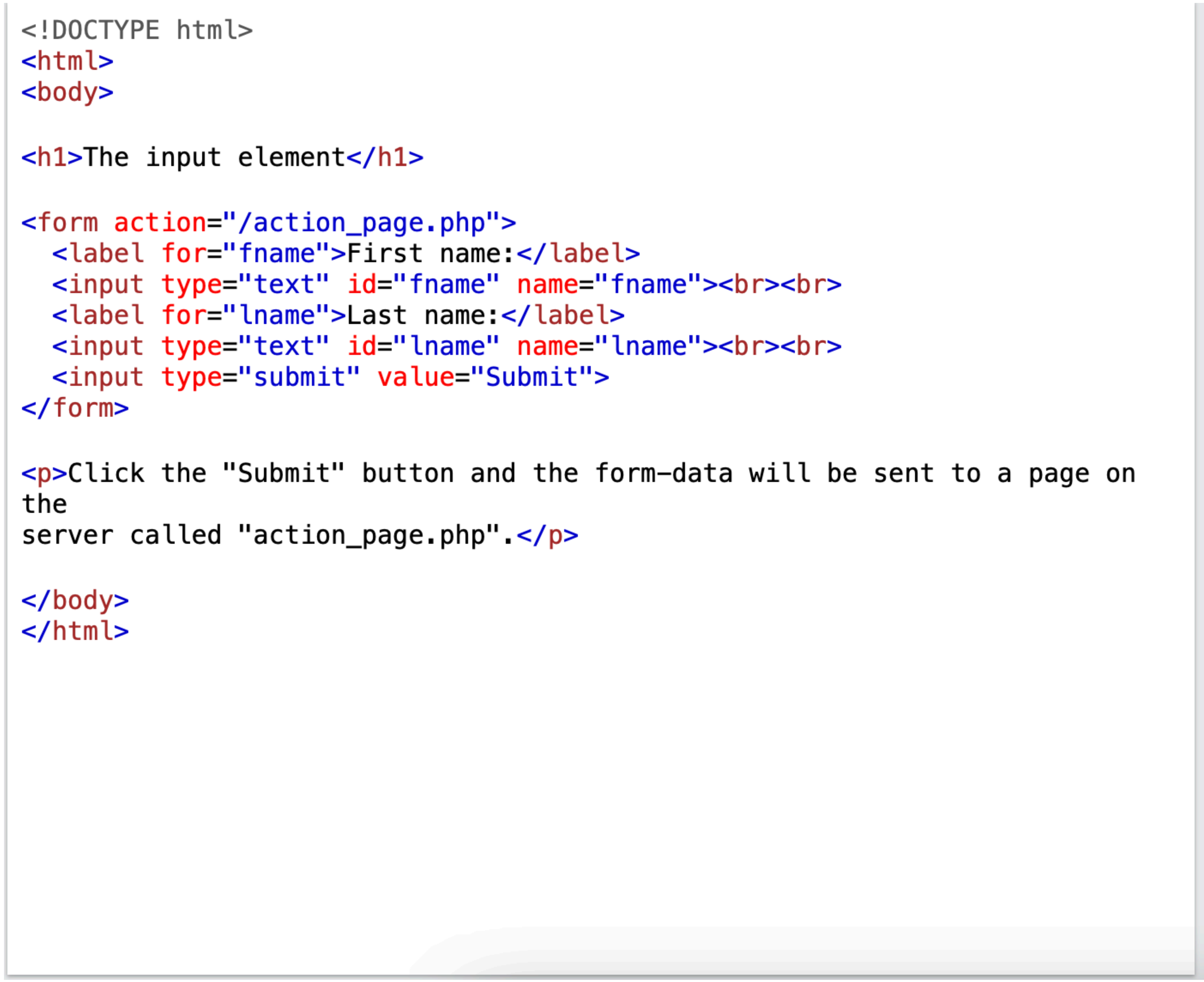
The button element - Styled with CSS

Change the background color of a button with the background-color property:

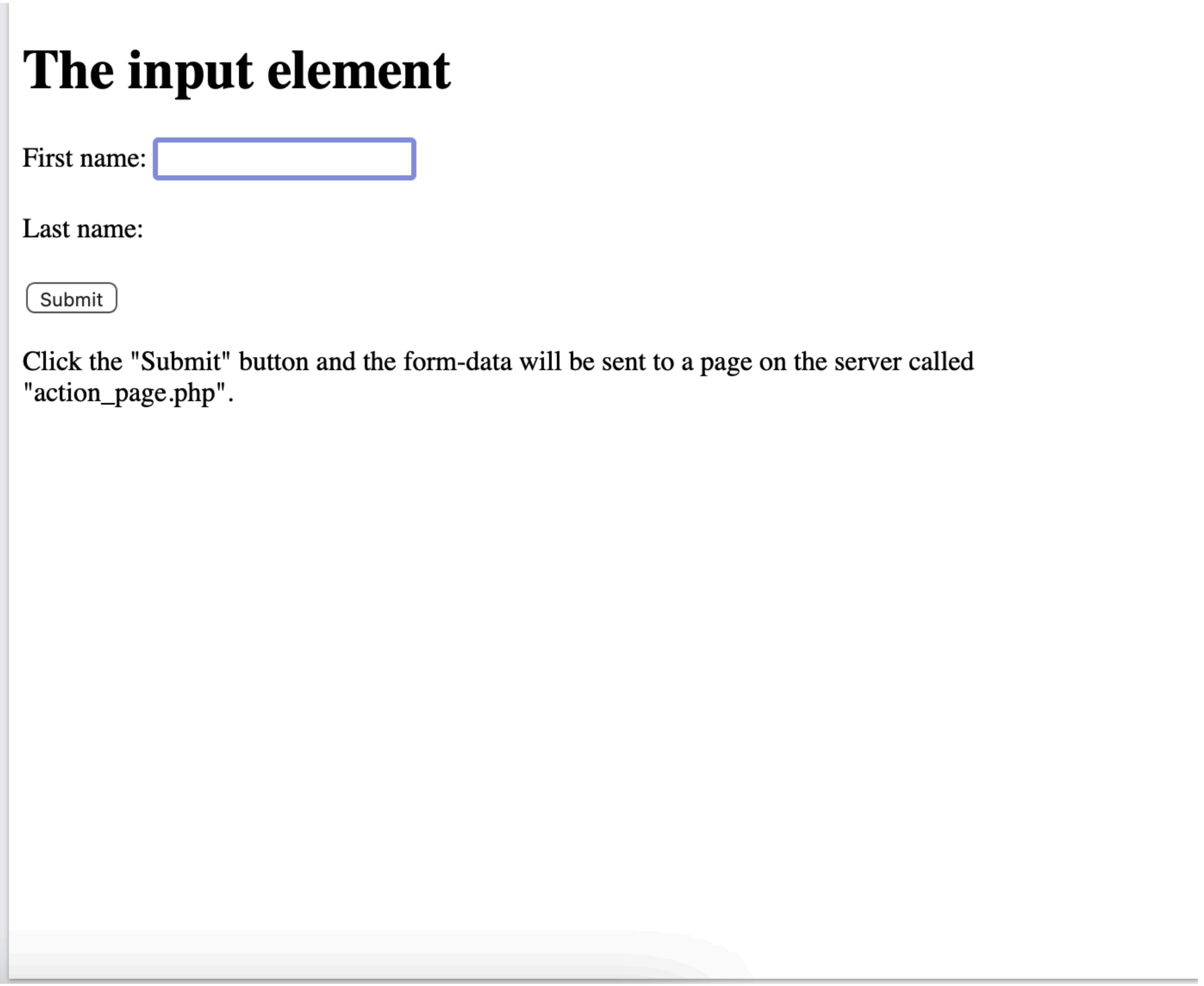
Text

香 港 中 文 大 學（深 圳）
The Chinese University of Hong Kong, Shenzhen

- Syntax
  - <input type="text">



```
<!DOCTYPE html>
<html>
<body>

<h1>The input element</h1>

<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click the "Submit" button and the form-data will be sent to a page on
the
server called "action_page.php".</p>

</body>
</html>
```



The input element

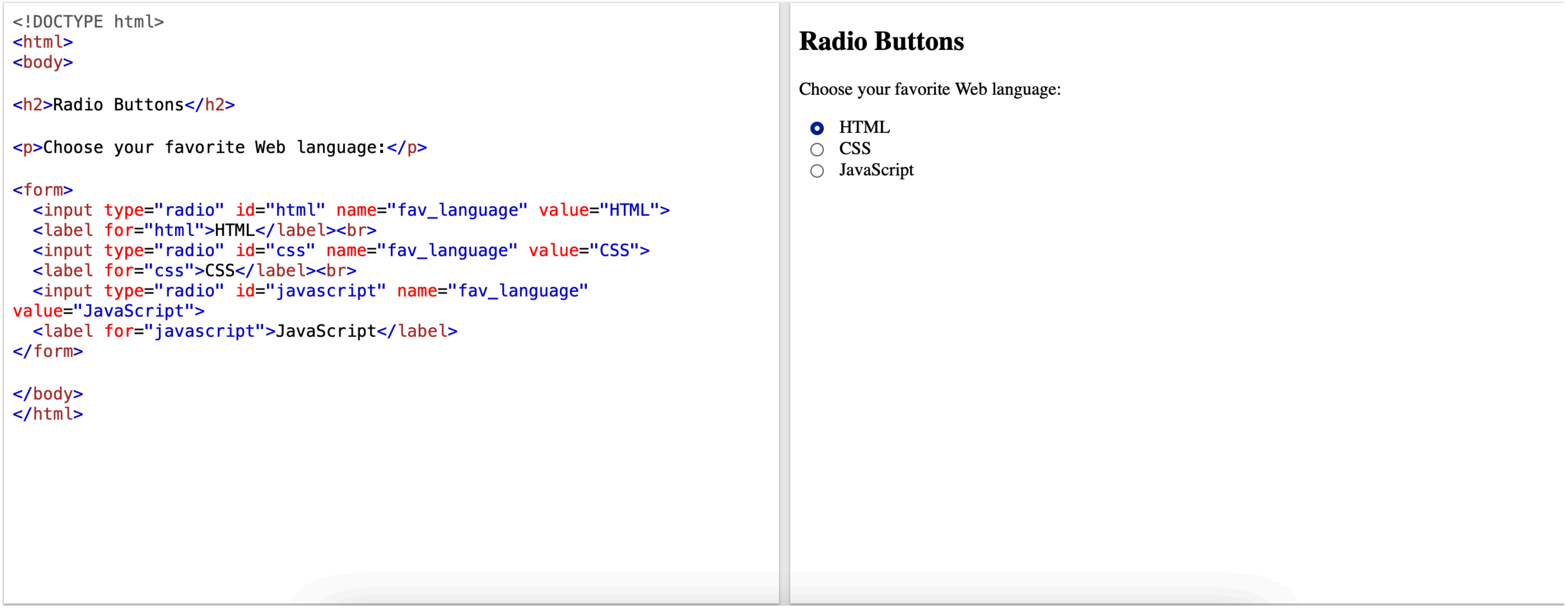
First name:

Last name:

Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php".

Radio

- Syntax
  - <input type=“radio” value= “content”>



```
<!DOCTYPE html>
<html>
<body>

<h2>Radio Buttons</h2>

<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>

</body>
</html>
```

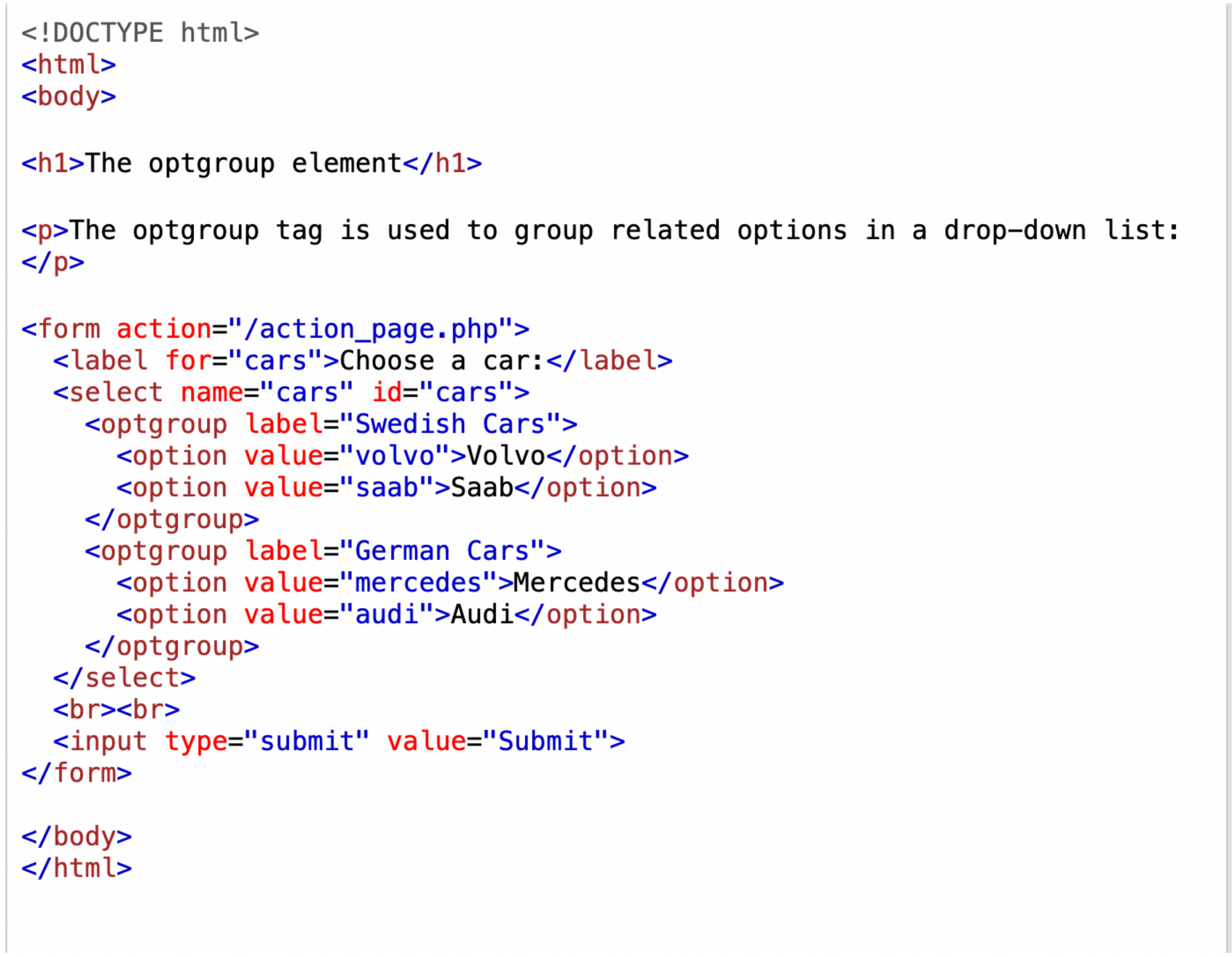
Radio Buttons

Choose your favorite Web language:

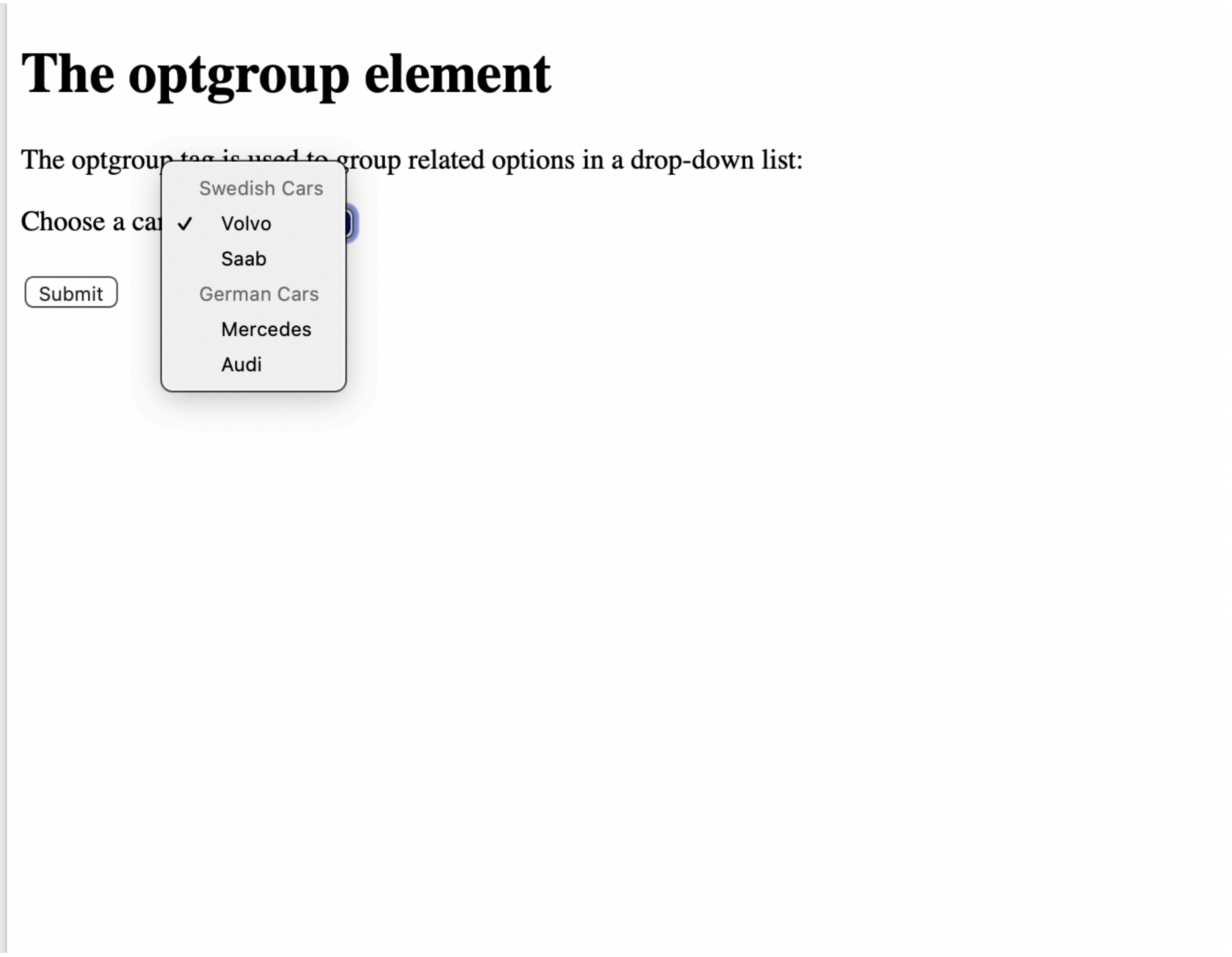
[✓ HTML]
CSS
JavaScript

Option

- Syntax
  - <select name = “name”> <option value=“value”>value</option> </select>



```
<!DOCTYPE html>
<html>
<body>

<h1>The optgroup element</h1>

<p>The optgroup tag is used to group related options in a drop-down list:
</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select name="cars" id="cars">
    <optgroup label="Swedish Cars">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
    </optgroup>
    <optgroup label="German Cars">
      <option value="mercedes">Mercedes</option>
      <option value="audi">Audi</option>
    </optgroup>
  </select>
  <br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```



The optgroup element

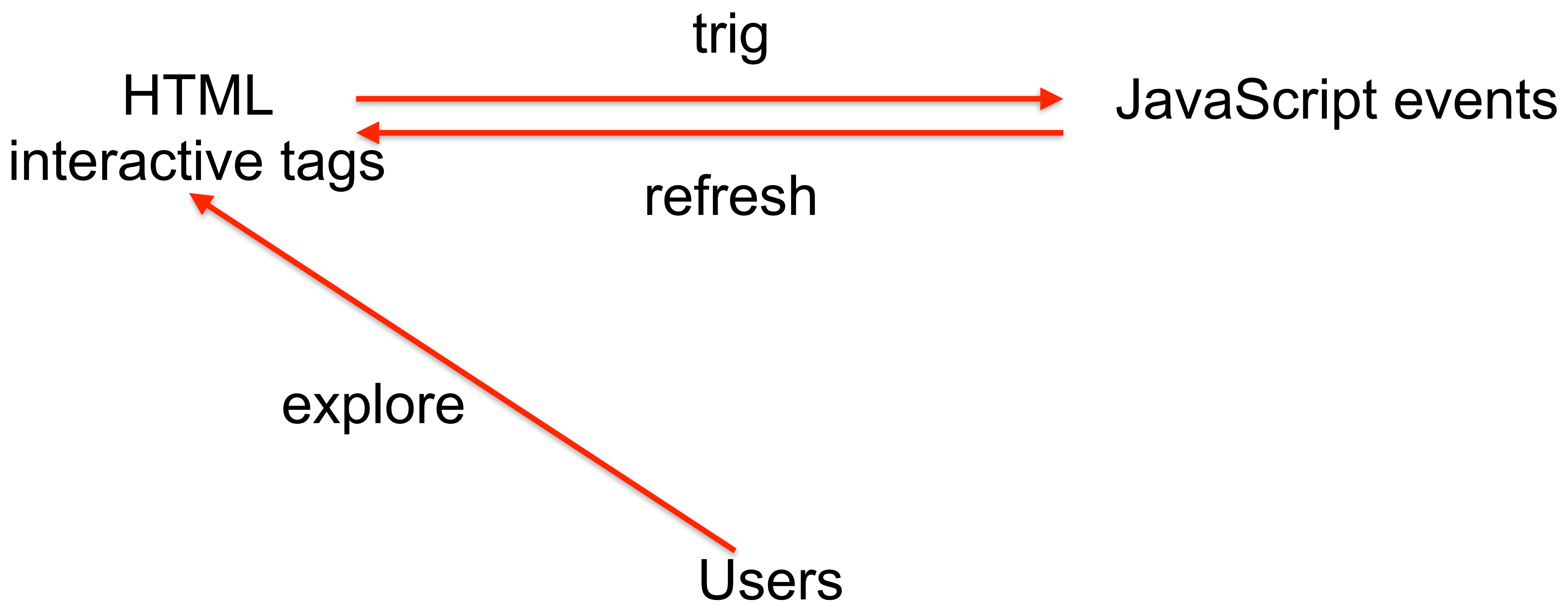
The optgroup tag is used to group related options in a drop-down list:

Choose a car:

More interactive tags

- Checkbox
- Color
- Passward
- Search
- Date/time/month/number
- Examples can be found at https://www.w3schools.com/tags/att_input_type.asp

Interactive tags and JavaScript events



JavaScript events

- Something a user does
  - An HTML input field is changed
  - An HTML button is clicked
  - An mouse is moved
  - etc.

JavaScript events

Event	**Description**
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Button and JavaScript event

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Create a button tag in HTML

```
<button class="button button1">Green</button>
```

$\downarrow$

```
<button class="button button1" onclick="click_button('green');">Green</button>
```

- Complete click_button function on JavaScript

```
click_button = function(c){
  color = c;
  console.log("You click "+color);
}
```

Slider and JavaScript event

- Create a slider tag in HTML

```
<button onclick="Click()">Display the chosen value</button>
<label for="vol">Volume (between 0 and 50):</label>
<input type="range" id="vol" name="vol" min="0" max="50" value="25" step="1">
<p id="demo"> </p>
```

- Get slider value in Javascript and show in HTML

```
function Click(){
        var slider = document.getElementById("vol");
        var value = slider.value;
        document.getElementById("demo").innerHTML = value;
}
```

EventListener

- Add or remove an event to a element
  - Mousemove
  - Mousedown
  - Mouseup
  - etc.

Mousmove event

```
window.addEventListener('load', ()=>{
    document.addEventListener('mousemove', display);
});

function display(event){
    let x = Math.random();
    document.getElementById("demo").innerHTML = x;
}
```

Interaction with SVG

- Interact with one object
- Interact with multiple objects and multiple listeners

Interaction practice

- Create a button in HTML and once users click the button, a rectangle with random color will be shown

Additional resources

- Javascript: https://www.w3schools.com/js/default.asp
- SVG: https://www.w3schools.com/graphics/svg_intro.asp