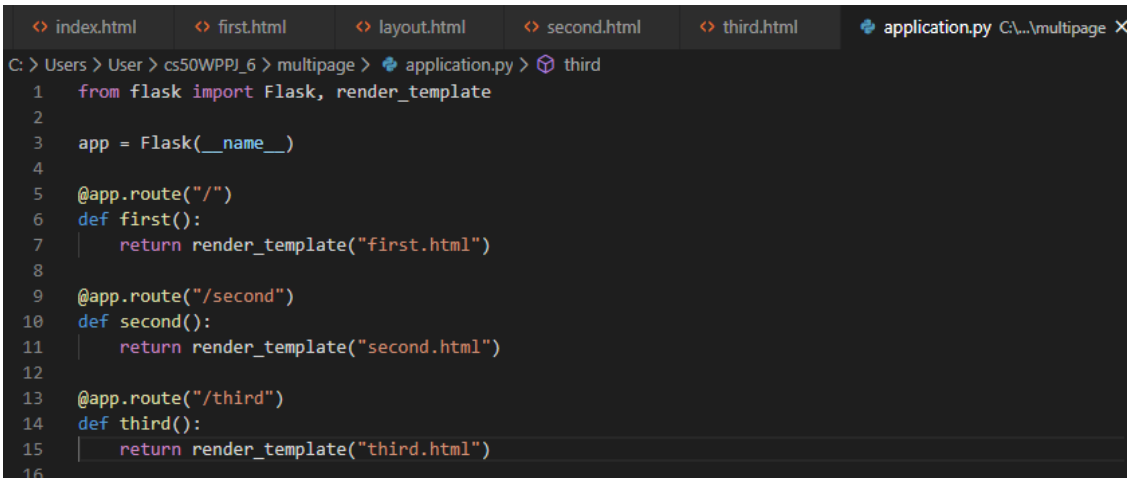
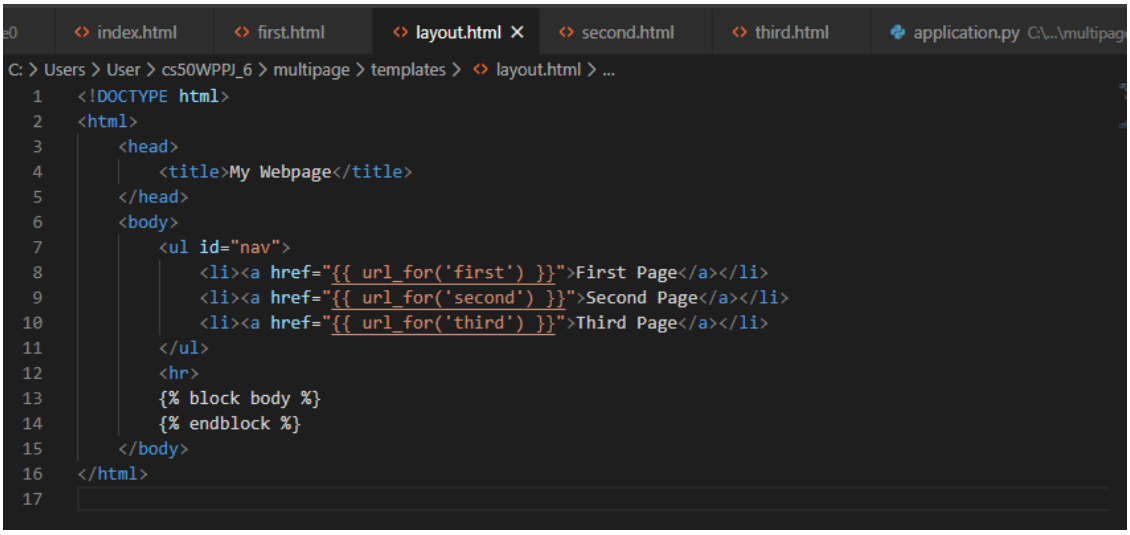
CS50's Web Programming with Python and JavaScript_Lecture 6_FrontEnd_Templates_D3

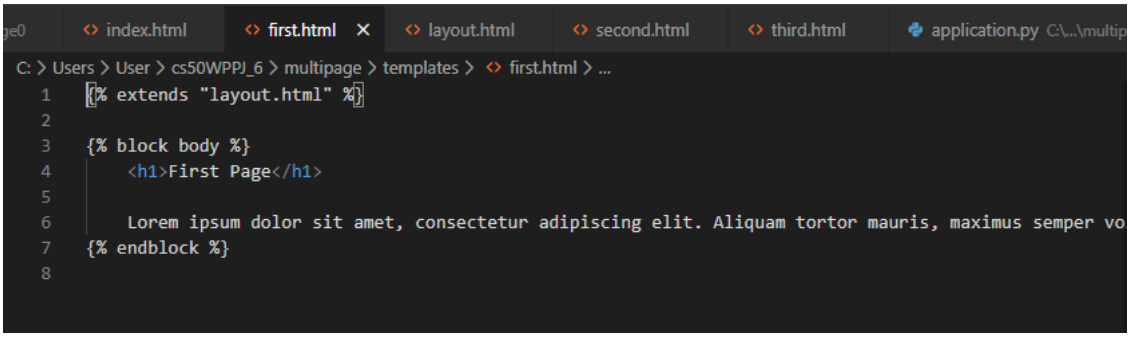
April 9, 2023

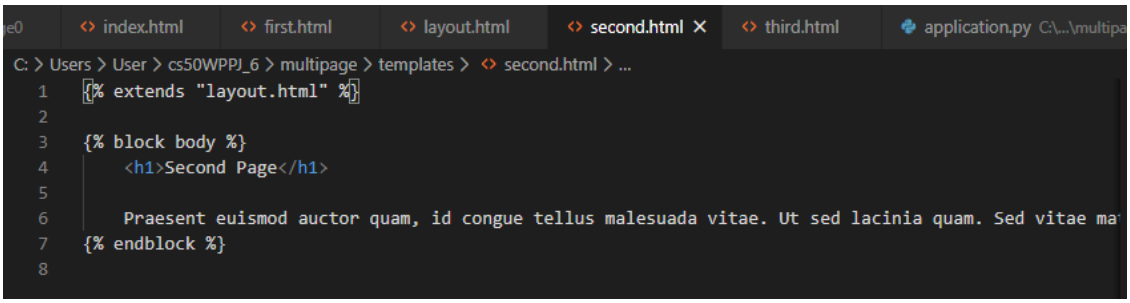
0.0.1 Multi-page vs single-page

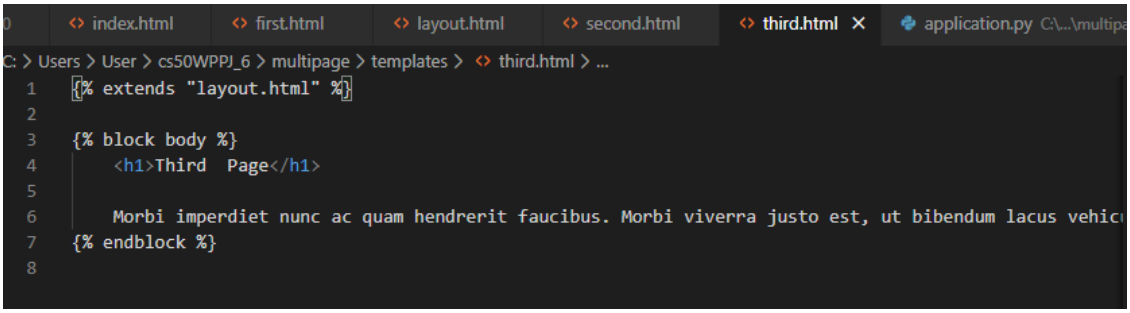
0.0.2 Multi-page example



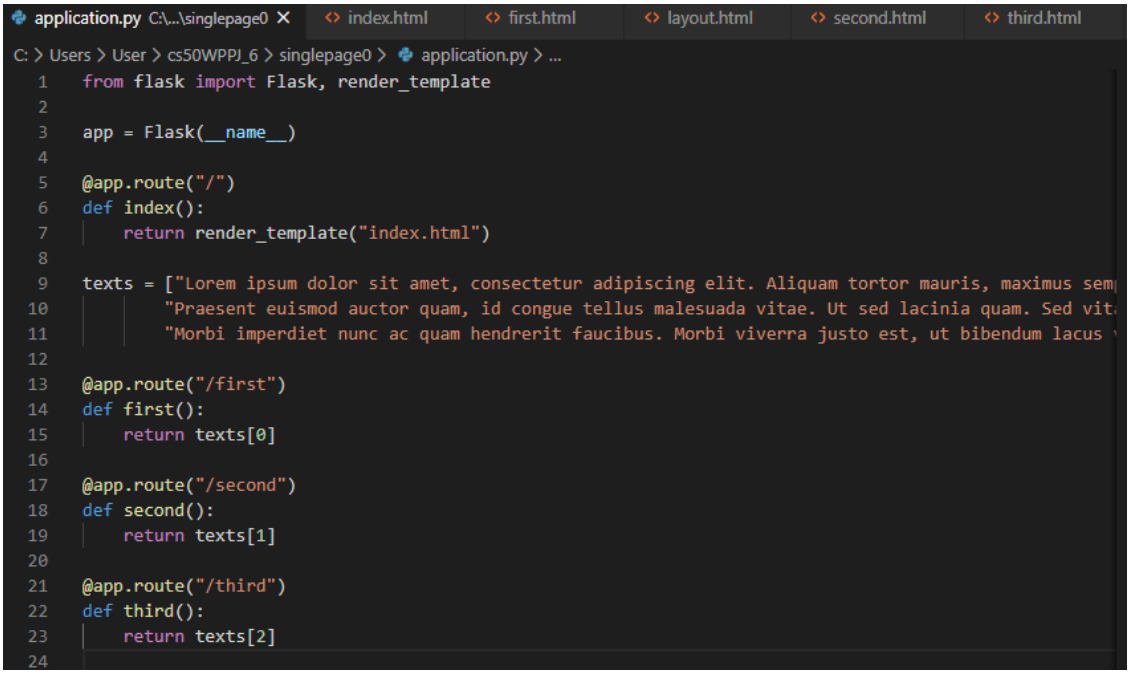



index.html
first.html
layout.html
second.html
third.html
application.py
C: > Users > User > cs50WPPJ_6 > multipage > templates > first.html > ...
{% extends "layout.html" %}
{% block body %}
<h1>First Page</h1>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tortor mauris, maximus semper vo
{% endblock %}


index.html
first.html
layout.html
second.html
third.html
application.py
C: > Users > User > cs50WPPJ_6 > multipage > templates > second.html > ...
{% extends "layout.html" %}
{% block body %}
<h1>Second Page</h1>
Praesent euismod auctor quam, id congue tellus malesuada vitae. Ut sed lacinia quam. Sed vitae ma
{% endblock %}


index.html
first.html
layout.html
second.html
third.html
application.py
C: > Users > User > cs50WPPJ_6 > multipage > templates > third.html > ...
{% extends "layout.html" %}
{% block body %}
<h1>Third Page</h1>
Morbi imperdiet nunc ac quam hendrerit faucibus. Morbi viverra justo est, ut bibendum lacus vehic
{% endblock %}

0.0.3 Same application in single-page



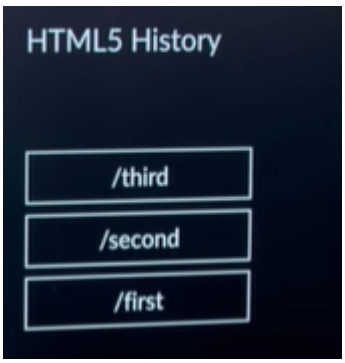
application.py C:\...\singlepage0 | index.html | first.html | layout.html | second.html | third.html

C: > Users > User > cs50WPPJ_6 > singlepage0 > templates > index.html > html > head > script > document.addEventListener('D

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <script>

            document.addEventListener('DOMContentLoaded', () => {

                // Start by loading first page.
                load_page('first');

                // Set links up to load new pages.
                document.querySelectorAll('.nav-link').forEach(oo => { // argument name could be
                    // named anything as it stands for all the elements under .nav-link
                    oo.onclick = () => {
                        load_page(oo.dataset.page);
                        return false;
                    };
                });
            });

            // Renders contents of new page in main view.
            function load_page(name) {
                const request = new XMLHttpRequest();
                request.open('GET', `/${name}`);
                request.onload = () => {
                    const response = request.responseText;
                    document.querySelector('#body').innerHTML = response;
                };
                request.send();
            }

        </script>
    </head>
    <body>
        <ul id="nav">
            <li><a href="" class="nav-link" data-page="first">First Page</a></li>
            <li><a href="" class="nav-link" data-page="second">Second Page</a></li>
            <li><a href="" class="nav-link" data-page="third">Third Page</a></li>
        </ul>
        <hr>
        <div id="body">
        </div>
    </body>
</html>
```

Now, new content is shown without reloading the page. One problem with this is that the URL doesn't change. Now, what if we want to change the URL as we click but not reload the page?

0.0.4 HTML5 History API



Use **history.pushState()**

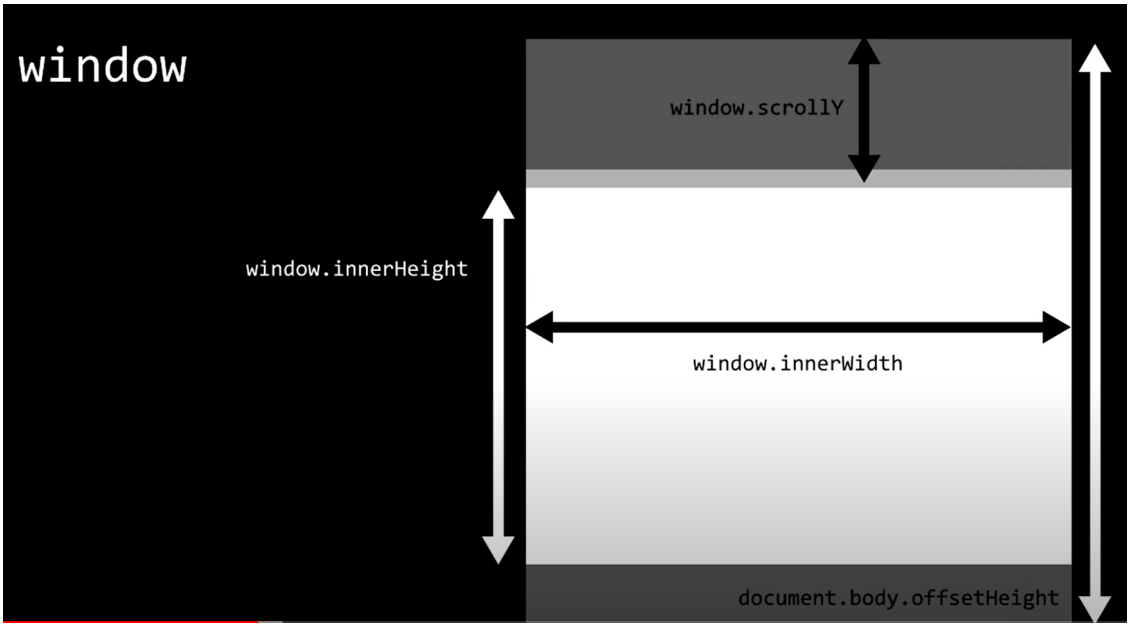
The only change will be in the **load_page()** function and the inclusion of **history.pushState()**. **history.pushState(data, title, URL)**

```
// Renders contents of new page in main view.
function load_page(name) {
    const request = new XMLHttpRequest();
    request.open('GET', `/${name}`);
    request.onload = () => {
        const response = request.responseText;
        document.querySelector('#body').innerHTML = response;

        // Push state to URL.
        document.title = name;
        history.pushState(null, name, name);
    };
    request.send();
}
```

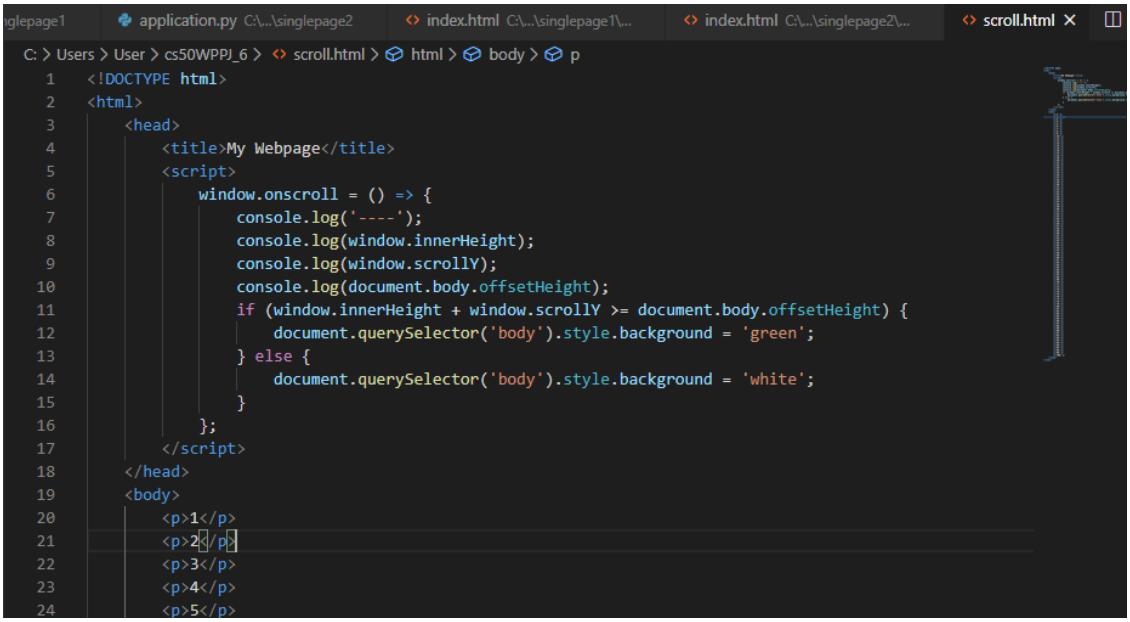
There is still problem with this, as when we click back button, we go back but the content doesn't change.

```
// Update text on popping state.
window.onpopstate = e => { // someone pressed the back button
    // e is event that took place
    const data = e.state; // this contains title and text
    document.title = data.title;
    document.querySelector('#body').innerHTML = data.text;
};

// Renders contents of new page in main view.
function load_page(name) {
    const request = new XMLHttpRequest();
    request.open('GET', `/${name}`);
    request.onload = () => {
        const response = request.responseText;
        document.querySelector('#body').innerHTML = response;

        // Push state to URL.
        document.title = name;
        history.pushState({'title': name, 'text': response}, name, name); // (data, title, UR
    };
    request.send();
}
```

0.1 window

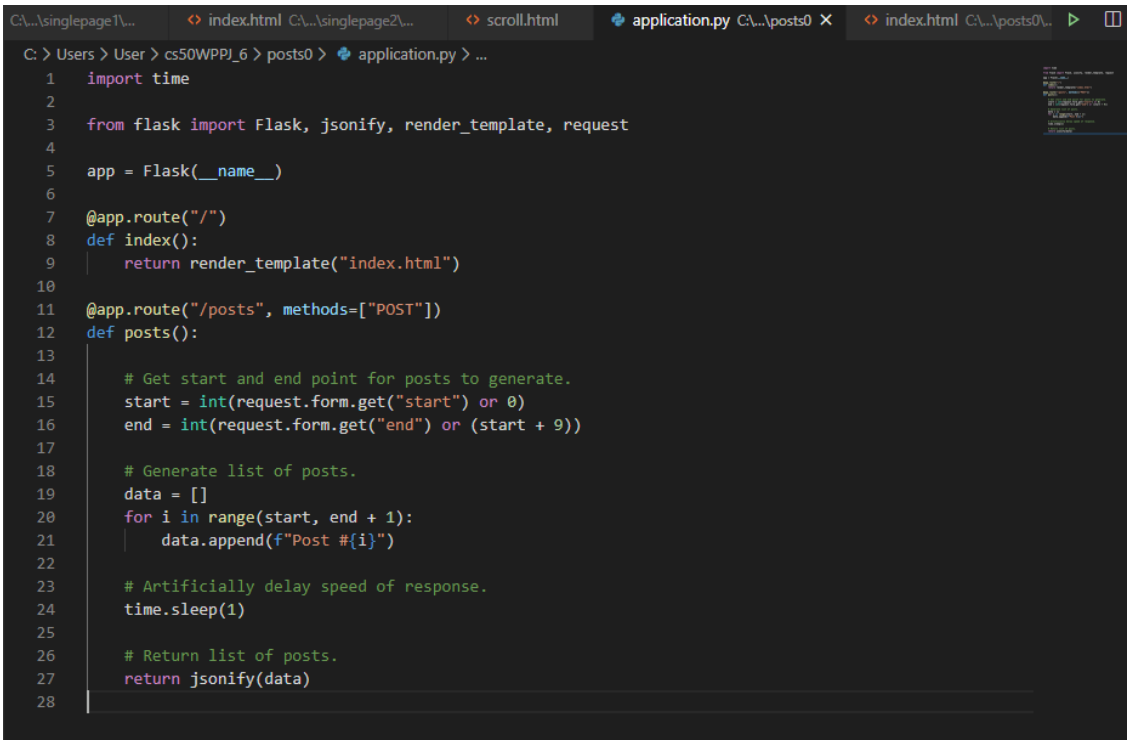


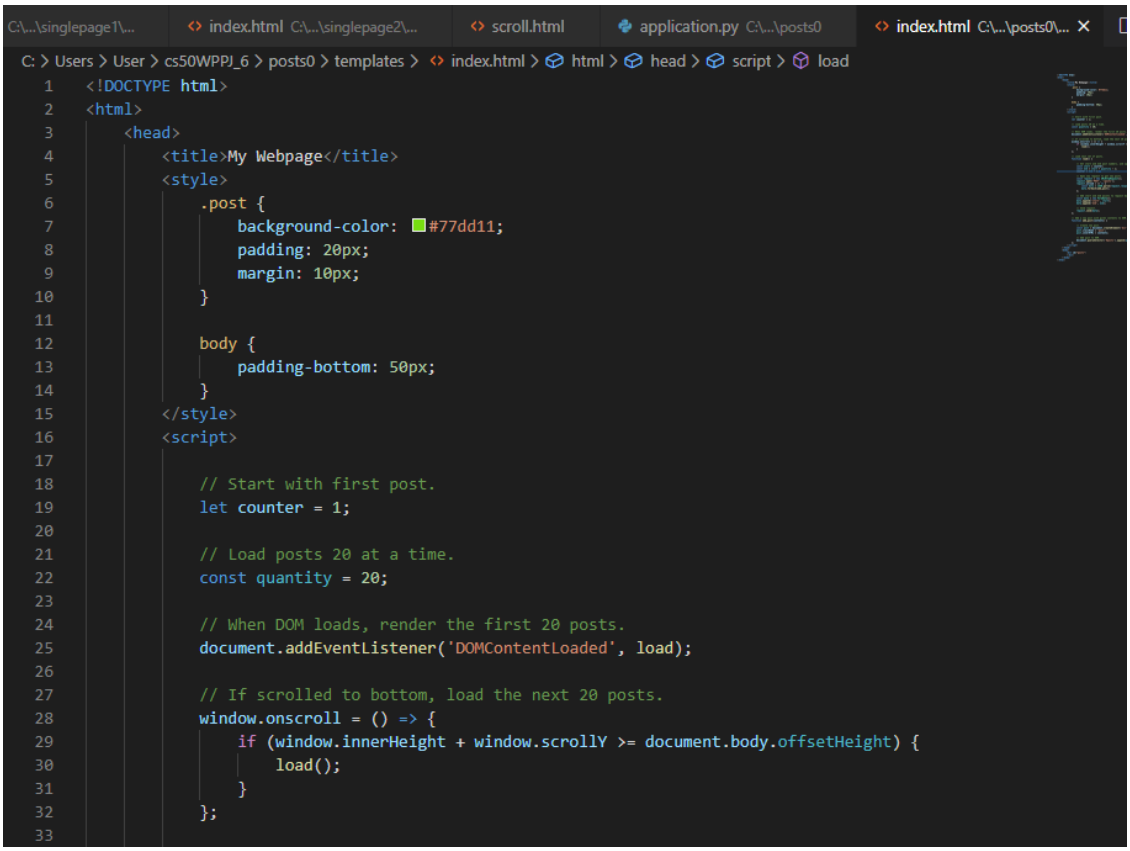
0.1.1 Turn the page's background to blue when we reach the bottom



0.1.2 Automatically increasing posts or show more posts as user scrolls down

Infinite scrolling




C:\...\singlepage1\...
index.html C:\...\singlepage2\...
scroll.html
application.py C:\...\posts0
index.html C:\...\posts0\...
C: > Users > User > cs50WPPJ_6 > posts0 > templates > index.html > html > head > script > load
<!DOCTYPE html>
<html>
<head>
<title>My Webpage</title>
<style>
.post {
background-color: #77dd11;
padding: 20px;
margin: 10px;
}
body {
padding-bottom: 50px;
}
</style>
<script>
// Start with first post.
let counter = 1;
// Load posts 20 at a time.
const quantity = 20;
// When DOM loads, render the first 20 posts.
document.addEventListener('DOMContentLoaded', load);
// If scrolled to bottom, load the next 20 posts.
window.onscroll = () => {
if (window.innerHeight + window.scrollY >= document.body.offsetHeight) {
load();
}
};

```
            // Load next set of posts.
            function load() {

                // Set start and end post numbers, and update counter.
                const start = counter;
                const end = start + quantity - 1;
                counter = end + 1;

                // Open new request to get new posts.
                const request = new XMLHttpRequest();
                request.open('POST', '/posts');
                request.onload = () => {
                    const data = JSON.parse(request.responseText);
                    data.forEach(add_post);
                };

                // Add start and end points to request data.
                const data = new FormData();
                data.append('start', start);
                data.append('end', end);

                // Send request.
                request.send(data);
            };

            // Add a new post with given contents to DOM.
            function add_post(contents) {

                // Create new post.
                const post = document.createElement('div');
                post.className = 'post';
                post.innerHTML = contents;

                // Add post to DOM.
                document.querySelector('#posts').append(post);
            };
        </script>
    </head>
    <body>
        <div id="posts">
        </div>
    </body>
</html>

```

0.1.3 Hide posts – posts1

Add this in the syle

```
.hide  {
    float: right;
}
```

Now, modify the **add_post()** function by adding the new lines starting from **const hide**

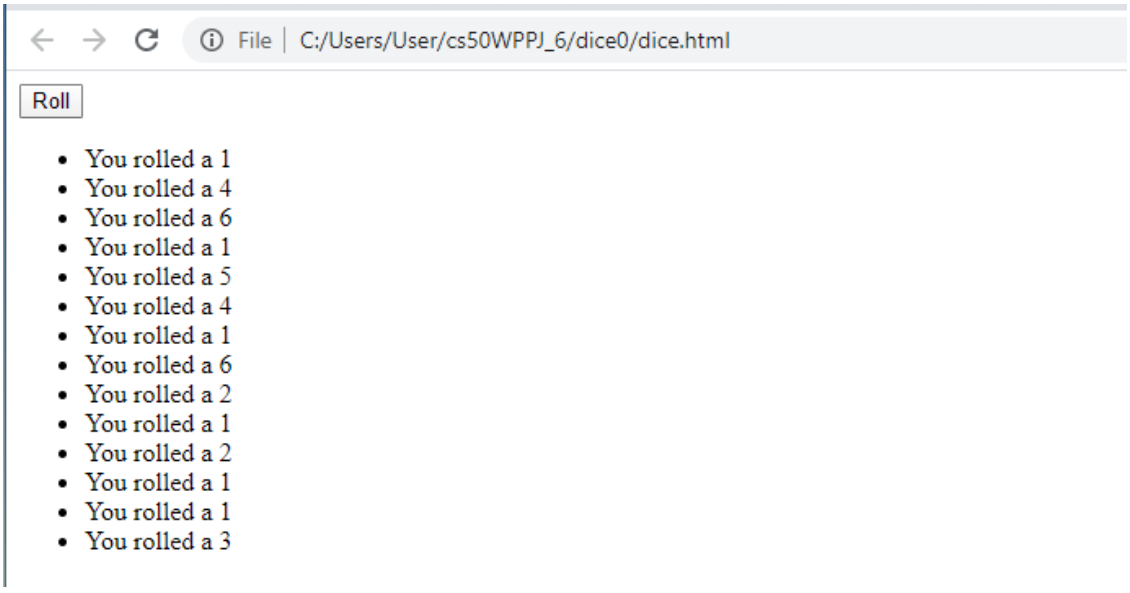
```
// Add a new post with given contents to DOM.
function add_post(contents) {

    // Create new post.
    const post = document.createElement('div');
    post.className = 'post';
    post.innerHTML = contents;

    // Add button to hide post.
    const hide = document.createElement('button');
    hide.className = 'hide';
    hide.innerHTML = 'Hide';
    post.append(hide);

    // When hide button is clicked, remove post.
    hide.onclick = function() {
        this.parentElement.remove();
    };

    // Add post to DOM.
    document.querySelector('#posts').append(post);
};
```

0.1.4 JS templates

mustache, underscore, handlebar

0.1.5 HandleBars (Jinja on the server for Flask and HandleBars on the browser/client)

```
tion.py C:\...\posts0   index.html C:\...\posts0\...   application.py C:\...\posts1   index.html C:\...\posts1\...   dice.html
> Users > User > cs50WPPJ_6 > dice0 > dice.html > ...
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.0.11/handlebars.min.js"></
        <script>

            // Template for roll results
            const template = Handlebars.compile("<li>You rolled a {{ value }}</li>");

            document.addEventListener('DOMContentLoaded', () => {
                document.querySelector('#roll').onclick = () => {

                    // Generate a random roll.
                    const roll = Math.floor((Math.random() * 6) + 1);

                    // Add roll result to DOM.
                    const content = template({'value': roll});
                    document.querySelector('#rolls').innerHTML += content;
                };
            });
        </script>
    </head>
    <body>
        <button id="roll">Roll</button>
        <ul id="rolls">
        </ul>
    </body>
</html>
```



index.html C:\...\singlepage2\... | scroll.html | application.py C:\...\posts0 | application.py C:\...\posts1 | dice.html C:\...\dice0 | dice.html C:\...\dice1

Users > User > cs50WPPJ_6 > dice1 > dice.html > html > head > style > img

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <style>
            li {
                line-height: 30px;
            }
            img {
                height: 30px;
                vertical-align: middle;
            }
        </style>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.0.11/handlebars.min.js"></script>
        <script>

            // Template for roll results
            const template = Handlebars.compile("<li>You rolled: <img src=\"img/{{ value }}.png\"></li>");

            document.addEventListener('DOMContentLoaded', () => {
                document.querySelector('#roll').onclick = () => {

                    // Generate a random roll.
                    const roll = Math.floor((Math.random() * 6) + 1);

                    // Add roll result to DOM.
                    const content = template({'value': roll});
                    document.querySelector('#rolls').innerHTML += content;
                };
            });
        </script>
    </head>
    <body>
        <button id="roll">Roll</button>
        <ul id="rolls">
        </ul>
    </body>
</html>

```

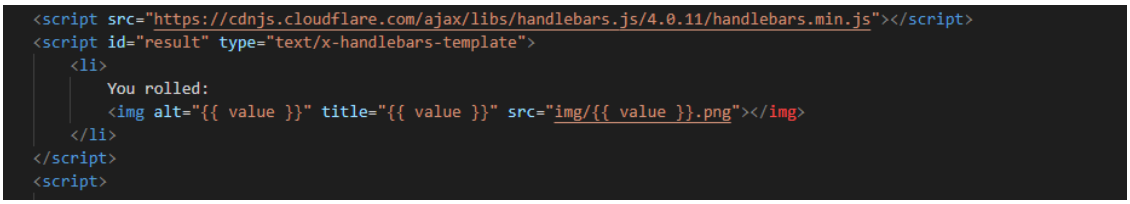
CS50's | cs50's | docun | HTML | Googl | Googl | Home | New Tab | M

File | C:/Users/User/cs50WPPJ_6/dice1/dice.html

Roll

- You rolled: 
- You rolled: 
- You rolled: 
- You rolled: 
- You rolled: 
- You rolled: 
- You rolled: 
- You rolled: 

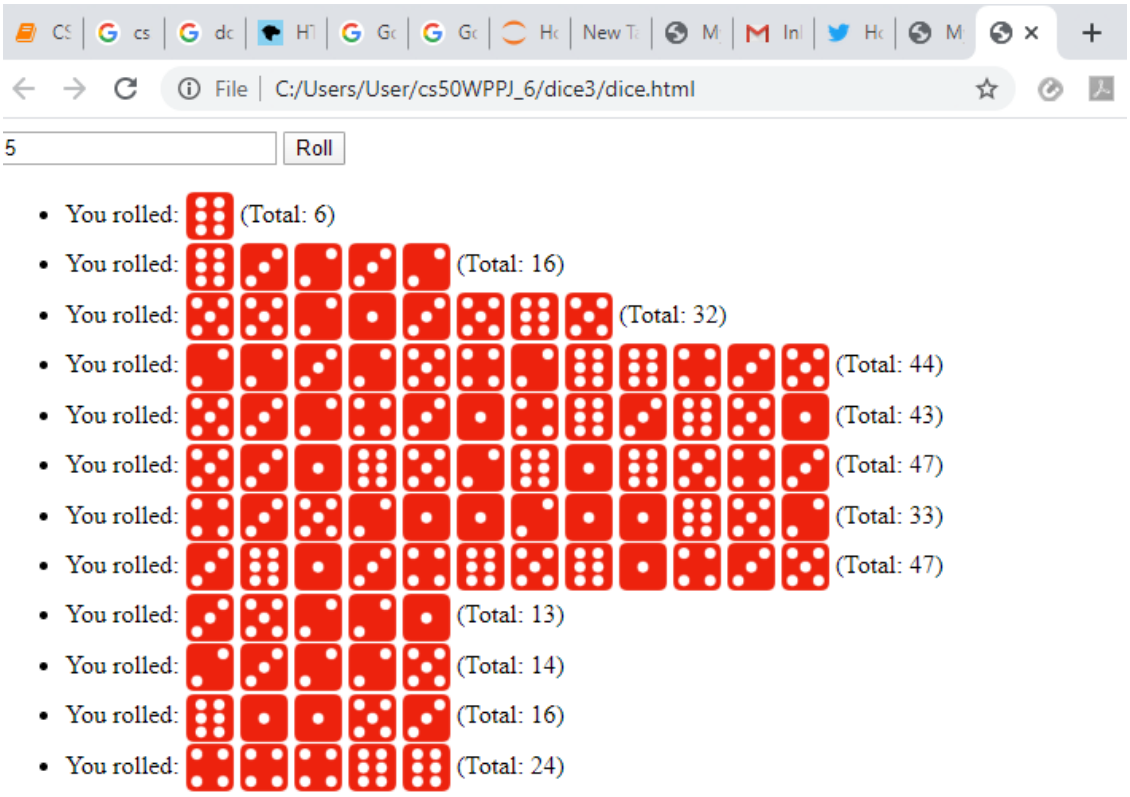
0.1.6 A more efficient way where we have less clutter of HTML and JS



0.1.7 Another version of dice

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.0.11/handlebars.min.js"></script>
<script id="result" type="text/template">
    <li>
        You rolled:
        {{#each values}}
            <img alt="{{ this }}" title="{{ this }}" src="img/{{ this }}.png">
        {{/each}}
        (Total: {{ total }})
    </li>
</script>
<script>
```

```
// Template for roll results
const template = Handlebars.compile(document.querySelector('#result').innerHTML);

document.addEventListener('DOMContentLoaded', () => {
    document.querySelector('#roll').onclick = ()  => {

        // Generate random rolls.
        const counter = parseInt(document.querySelector('#counter').value);
        const rolls = [];
        let total = 0;
        for (let i = 0; i < counter; i++) {
            const value = Math.floor(Math.random() * 6) +  1;
            rolls.push(value);
            total += value;
        };

        // Add roll results to DOM.
        const content = template({'values': rolls, 'total': total});
        document.querySelector('#rolls').innerHTML += content;
    };
});
```



0.1.8 Efficient way of creating infinite scrolling using template

Make these changes in **index.html**

Ignore **{{ contents }}** as Jinja as this is to be considered or processed by **handlebars**.

```
{% raw -%}
    {{ contents }}
{%- endraw %}
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.0.11/handlebars.min.js"></script>

<script id="post" type="text/x-handlebars-template">
    <div class="post">
        {% raw -%}
            {{ contents }}
        {%- endraw %}
        <button class="hide">Hide</button>
    </div>
</script>
```

```javascript
// If hide button is clicked, delete the post.
document.addEventListener('click', event => {
    const element = event.target;
    if (element.className === 'hide') {
        element.parentElement.remove();
    }
});
```

```javascript
// Add a new post with given contents to DOM.
const post_template = Handlebars.compile(document.querySelector('#post').innerHTML);
function add_post(contents) {

    // Create new post.
    const post = post_template({'contents': contents});

    // Add post to DOM.
    document.querySelector('#posts').innerHTML += post;
}
```

0.2 Animation

dice.html C:\...\dice2 | dice.html C:\...\dice3 | application.py C:\...\posts2 | index.html C:\...\posts2\... | animate0.html

C: > Users > User > cs50WPPJ_6 > animate0.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <style>
            @keyframes grow {
                from {
                    font-size: 20px;
                }
                to  {
                    font-size: 100px;
                }
            }

            h1 {
                animation-name: grow;
                animation-duration: 2s;
                animation-fill-mode: forwards;
            }
        </style>
    </head>
    <body>
        <h1>Welcome!</h1>
    </body>
</html>
```

[2]:
```
from IPython.display import Video
```

[1]:
```
Video("1.trec")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-372993b7396e> in <module>
----> 1 Video("1.trec")
```

```
NameError: name 'Video' is not defined
```

0.2.1 Animate – from left to right

animate1.html × animate0.html

C: > Users > User > cs50WPPJ_6 > animate1.html > ...

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <style>
            @keyframes move {
                from {
                    left: 0%;
                }
                to  {
                    left: 50%;
                }
            }

            h1 {
                position: relative;
                animation-name: move;
                animation-duration: 3s;
                animation-fill-mode: forwards;
            }
        </style>
    </head>
    <body>
        <h1>Welcome!</h1>
    </body>
</html>

```

0.2.2 Animate – from left to right to left

0.2.3 Animate – from left to right to left for infinity when clicked, stops when we click, run again when we click

Note the use of **animation-iteration-count** inside css and the use of **animationPlayState** inside JS.

animate1.html | animate2.html | animate3.html | animate0.html

: > Users > User > cs50WPPJ_6 > animate3.html > html > head > style

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Webpage</title>
        <style>
            @keyframes move {
                0% {
                    left: 0%;
                }
                50% {
                    left: 50%;
                }
                100% {
                    left: 0%;
                }
            }

            h1 {
                position: relative;
                animation-name: move;
                animation-duration: 3s;
                animation-fill-mode: forwards;
                animation-iteration-count: infinite;
            }
        </style>
        <script>
            document.addEventListener('DOMContentLoaded', () => {
                const h1 = document.querySelector('h1');
                h1.style.animationPlayState  = 'paused';
                document.querySelector('button').onclick = () => {
                    if (h1.style.animationPlayState  === 'paused')
                        h1.style.animationPlayState = 'running';
                    else
                        h1.style.animationPlayState  = 'paused';
                };
            });
        </script>
    </head>
    <body>
        <button>Click Here!</button>
        <h1>Welcome!</h1>
    </body>
</html>
```

0.2.4 Efficient way of creating infinite scrolling using template – with animation – posts3

```css
@keyframes hide {
    from {
        opacity: 1;
    }
    to {
        opacity: 0;
    }
}

.hide  {
    float: right;
}

.post {
    background-color: #77dd11;
    padding: 20px;
    margin-bottom: 10px;
    animation-name: hide;
    animation-duration: 2s;
    animation-fill-mode: forwards;
    animation-play-state: paused;
}
```

```javascript
// If hide button is clicked, delete the post.
document.addEventListener('click', event => {
    const element = event.target;
    if (element.className === 'hide') {
        element.parentElement.style.animationPlayState = 'running';
        // when the animation end, remove it
        element.parentElement.addEventListener('animationend', () =>  {
            element.parentElement.remove();
        });
    }
});
```

0.2.5 Efficient way of creating infinite scrolling using template – with animation wiuth smoother transition from below – posts3

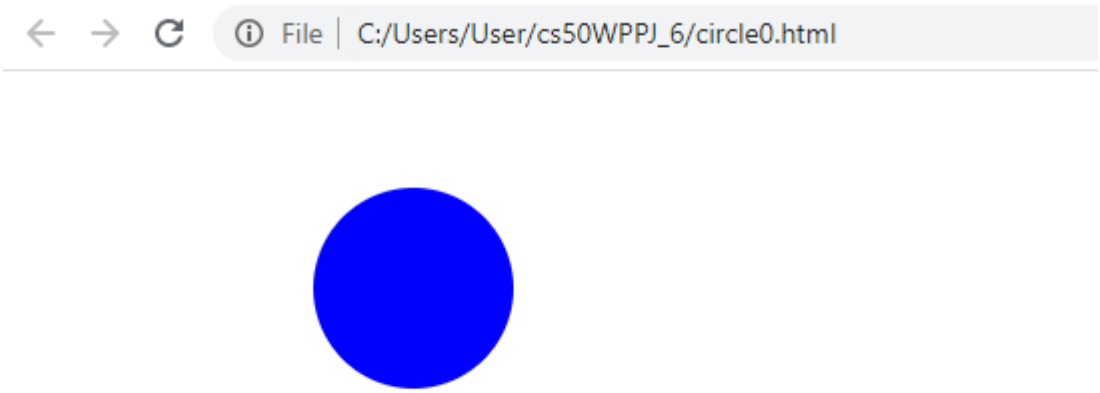
```css
@keyframes hide {
    0% {
        opacity: 1;
        height: 100%;
        line-height: 100%;
        padding: 20px;
        margin-bottom: 10px;
    }
    75% {
        opacity: 0;
        height: 100%;
        line-height: 100%;
        padding: 20px;
        margin-bottom: 10px;
    }
    100% {
        opacity: 0;
        height: 0px;
        line-height: 0px;
        padding: 0px;
        margin-bottom: 0px;
    }
}
```

0.3 SVG

ate2.html | animate3.html | application.py C:\...\posts3 | index.html C:\...\posts3\... | applicatio

C: > Users > User > cs50WPPJ_6 > circle0.html > ...

```html
<!DOCTYPE html>
<html>
    <body>
        <svg style="width:100%; height:800px">
            <circle cx="200" cy="100" r="50" style="fill:blue"/>
        </svg>
    </body>
</html>
```



te3.html | application.py C:\...\posts3 | index.html C:\...\posts3\... | applica

: > Users > User > cs50WPPJ_6 > circle1.html > ...

```
1   <!DOCTYPE html>
2   <html>
3       <head>
4           <script src="https://d3js.org/d3.v4.min.js"></script>
5       </head>
6       <body>
7           <svg id="svg" style="width:100%; height:800px"/>
8       </body>
9       <script>
10
11          const svg = d3.select('#svg');
12
13          svg.append('circle')
14              .attr('cx', 200)
15              .attr('cy', 200)
16              .attr('r', 90)
17              .style('fill', 'green');
18
19      </script>
20  </html>
```

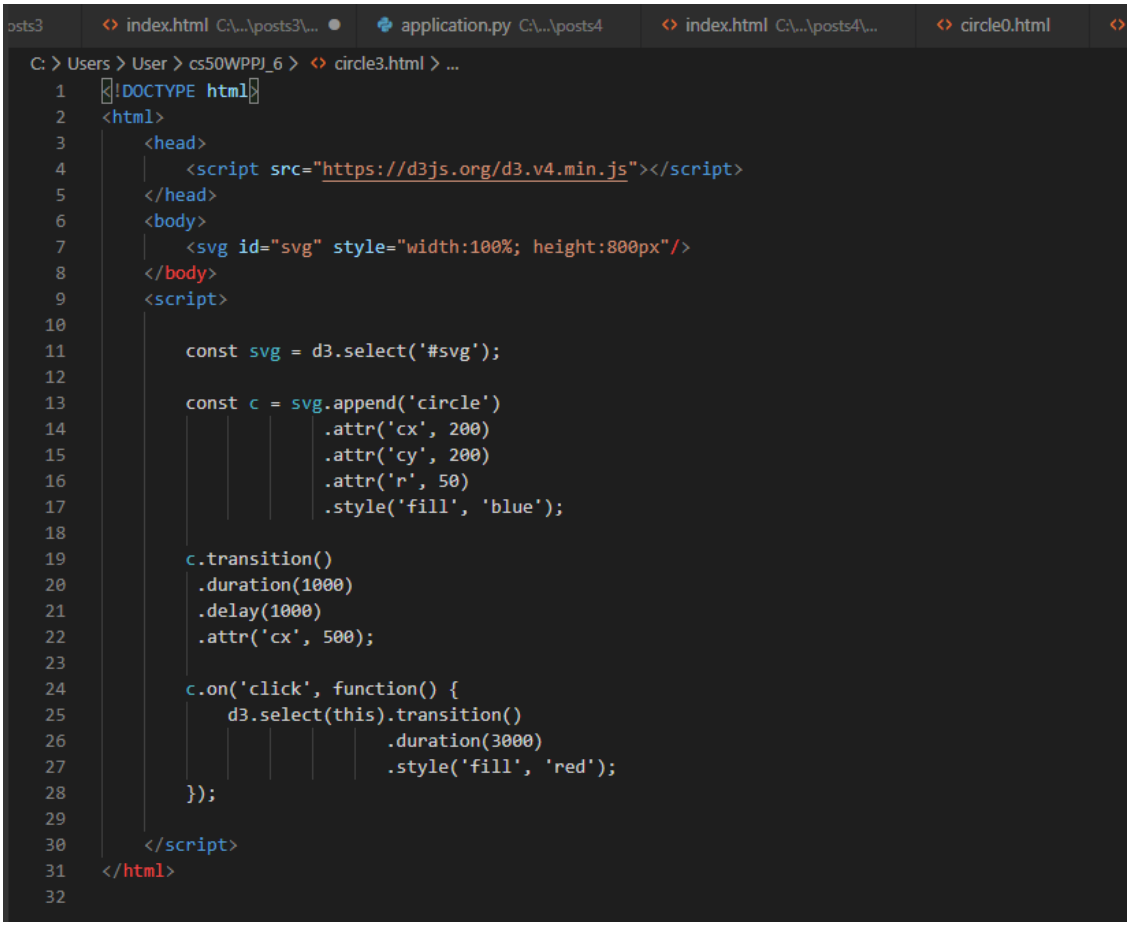

File | C:/Users/User/cs50WPPJ_6/circle1.html

0.3.1 Apply animation to SVG

: > Users > User > cs50WPPJ_6 > <> circle2.html > ...

```
<!DOCTYPE html>
<html>
    <head>
        <script src="https://d3js.org/d3.v4.min.js"></script>
    </head>
    <body>
        <svg id="svg" style="width:100%; height:800px"/>
    </body>
    <script>

        const svg = d3.select('#svg');

        const c = svg.append('circle')
                     .attr('cx', 200)
                     .attr('cy', 200)
                     .attr('r', 50)
                     .style('fill', 'blue');

        c.transition()
         .duration(1000)
         .attr('cx', 500)
         .attr('cy', 500)
         .style('fill', 'red');

    </script>
</html>
```

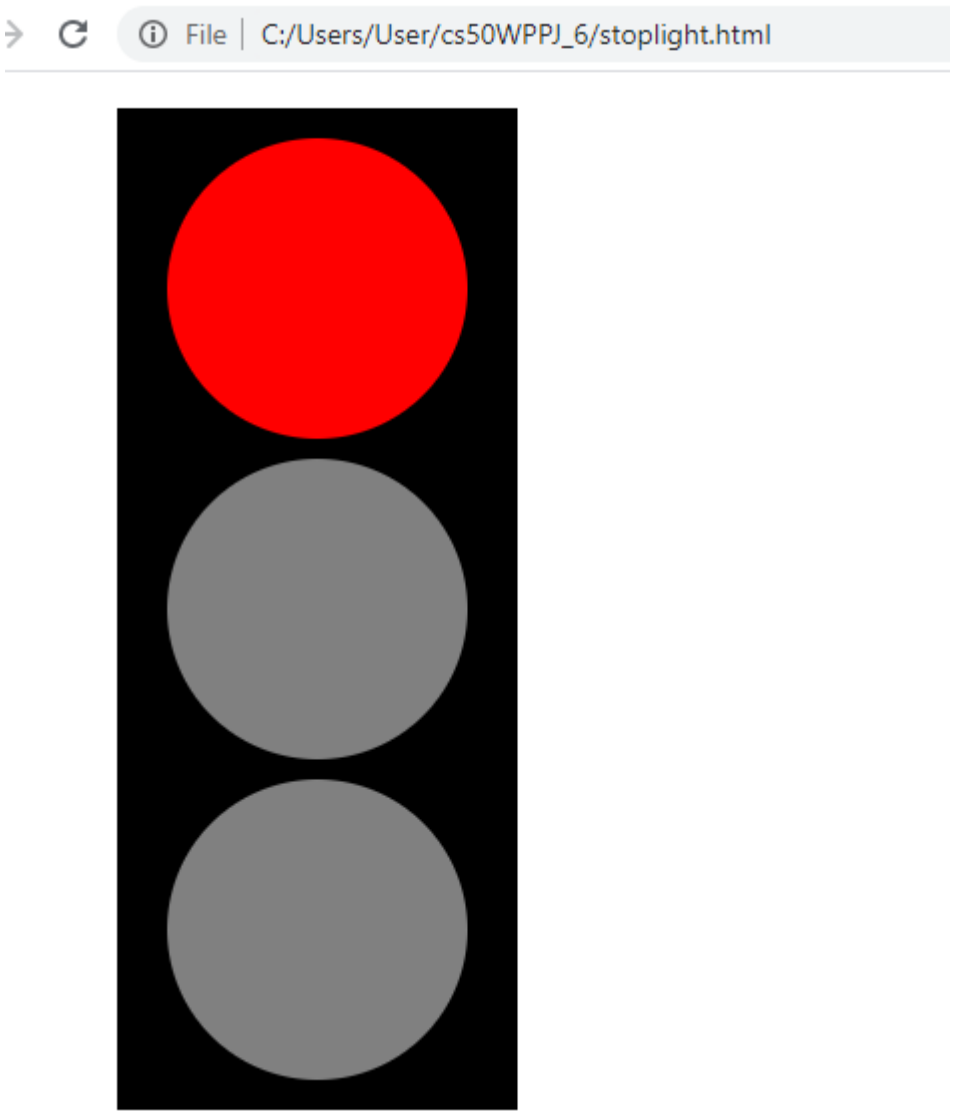
0.3.2 Another example – circle3



0.3.3 Stoplight

```
<!DOCTYPE html>
<html>
    <head>
        <script src="https://d3js.org/d3.v4.min.js"></script>
    </head>
    <body>
        <svg id="svg" style="width:100%; height:800px"/>
    </body>
    <script>

        const svg = d3.select('#svg');

        svg.append('rect')
           .attr('x', 100)
           .attr('y', 10)
           .attr('width', 200)
           .attr('height', 500)
           .style('fill', 'black');

        const red = svg.append('circle')
                       .attr('cx', 200)
                       .attr('cy', 100)
                       .attr('r', 75)
                       .style('fill', 'grey');

        const yellow = svg.append('circle')
                          .attr('cx', 200)
                          .attr('cy', 260)
                          .attr('r', 75)
                          .style('fill', 'grey');

        const green = svg.append('circle')
                         .attr('cx', 200)
                         .attr('cy', 420)
                         .attr('r', 75)
                         .style('fill', 'grey');
```

```
        red.on('click', () => {
            red.style('fill', 'red');
            yellow.style('fill', 'grey');
            green.style('fill', 'grey');
        });

        yellow.on('click', () => {
            red.style('fill', 'grey');
            yellow.style('fill', 'yellow');
            green.style('fill', 'grey');
        });

        green.on('click', () => {
            red.style('fill', 'grey');
            yellow.style('fill', 'grey');
            green.style('fill', 'green');
        });

    </script>
</html>
```


File
C:/Users/User/cs50WPPJ_6/stoplight.html

0.3.4 Draw points as we move mouse – no way to stop drawing

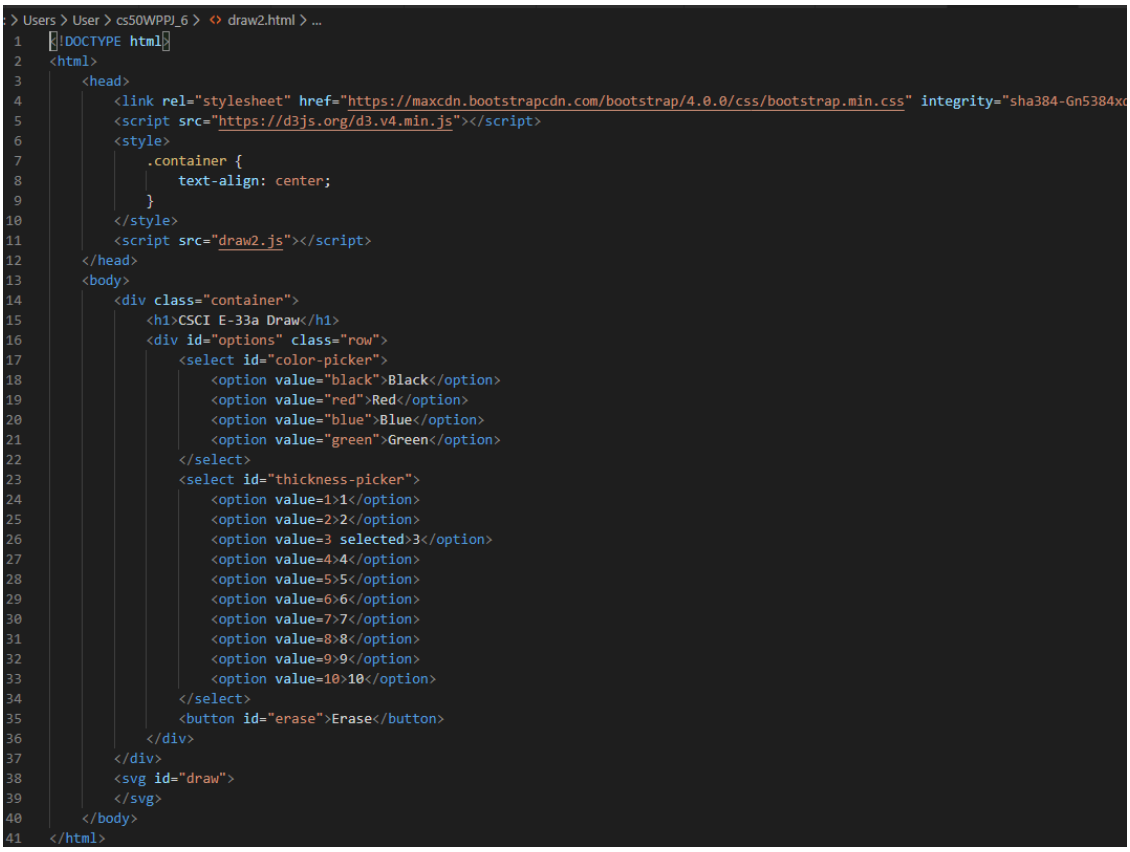
C: > Users > User > cs50WPPJ_6 > draw0.html > html > script > draw_point

```
<!DOCTYPE html>
<html>
    <head>
        <script src="https://d3js.org/d3.v4.min.js"></script>
    </head>
    <body>
        <svg id="svg" style="width:100%; height:800px"/>
    </body>
    <script>

        const svg = d3.select('#svg');

        function draw_point() {
            const coords = d3.mouse(this);

            svg.append('circle')
               .attr('cx', coords[0])
               .attr('cy', coords[1])
               .attr('r', 5)
               .style('fill', 'black');
        };

        svg.on('mousemove', draw_point);

    </script>
</html>
```

0.3.5 Now, make a site where only if we left click mouse and drag, the drawing occurs

```
    <script src="https://d3js.org/d3.v4.min.js"></script>
</head>
<body>
    <svg id="svg" style="width:100%; height:800px"/>
</body>
<script>

    const svg = d3.select('#svg');
    let drawing = false;

    function draw_point() {
        if (!drawing)
            return;

        const coords = d3.mouse(this);

        svg.append('circle')
           .attr('cx', coords[0])
           .attr('cy', coords[1])
           .attr('r', 5)
           .style('fill', 'black');
    };

    svg.on('mousedown', () => {
        drawing = true;
    });

    svg.on('mouseup', () => {
        drawing = false;
    });

    svg.on('mousemove', draw_point);
```

The problem with this and the previous app is that, if we move fast, there will be gaps between the points. What we can do is store the previous point and connect the present point with the previous one with a straight line.


Users > User > cs50WPPJ_6 > draw2.html > ...
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384x
<script src="https://d3js.org/d3.v4.min.js"></script>
<style>
.container {
text-align: center;
}
</style>
<script src="draw2.js"></script>
</head>
<body>

<h1>CSCI E-33a Draw</h1>

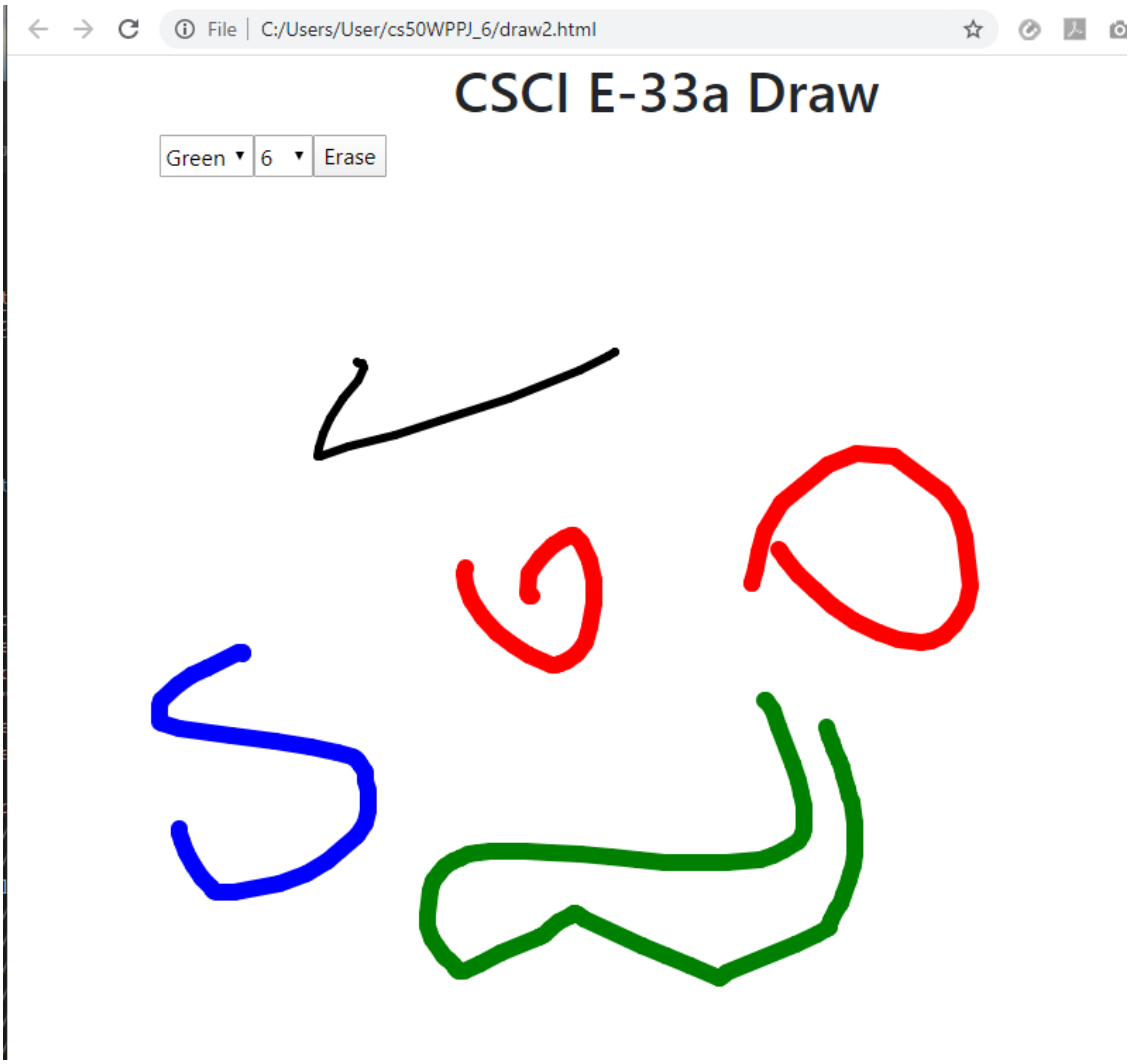
<select id="color-picker">
<option value="black">Black</option>
<option value="red">Red</option>
<option value="blue">Blue</option>
<option value="green">Green</option>
</select>
<select id="thickness-picker">
<option value=1>1</option>
<option value=2>2</option>
<option value=3 selected>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
</select>
<button id="erase">Erase</button>

<svg id="draw">
</svg>
</body>
</html>

C: > Users > User > cs50WPPJ_6 > JS draw2.js > document.addEventListener('DOMContentLoaded') callba

```
document.addEventListener('DOMContentLoaded', () => {

    // state
    let draw = false;

    // elements
    let points = [];
    let lines = [];
    let svg = null;

    function render() {

        // create the selection area
        svg = d3.select('#draw')
                .attr('height', window.innerHeight)
                .attr('width', window.innerWidth);

        svg.on('mousedown', function() {
            draw = true;
            const coords = d3.mouse(this);
            draw_point(coords[0], coords[1], false);
        });

        svg.on('mouseup', () =>{
            draw = false;
        });

        svg.on('mousemove', function() {
            if (!draw)
                return;
            const coords = d3.mouse(this);
            draw_point(coords[0], coords[1], true);
        });

        document.querySelector('#erase').onclick = () => {
            for (let i = 0; i < points.length; i++)
                points[i].remove();
            for (let i = 0; i < lines.length; i++)
                lines[i].remove();
            points = [];
            lines = [];
        }

    }
```

```
    function draw_point(x, y, connect) {

        const color = document.querySelector('#color-picker').value;
        const thickness = document.querySelector('#thickness-picker').value;

        if (connect) {
            const last_point = points[points.length - 1];
            const line = svg.append('line')
                            .attr('x1', last_point.attr('cx'))
                            .attr('y1', last_point.attr('cy'))
                            .attr('x2', x)
                            .attr('y2', y)
                            .attr('stroke-width', thickness * 2)
                            .style('stroke', color);
            lines.push(line);
        }

        const point = svg.append('circle')
                         .attr('cx', x)
                         .attr('cy', y)
                         .attr('r', thickness)
                         .style('fill', color);
        points.push(point);
    }

    render();
});
```


File | C:/Users/User/cs50WPPJ_6/draw2.html
CSCI E-33a Draw
Green
6
Erase

[ ]: