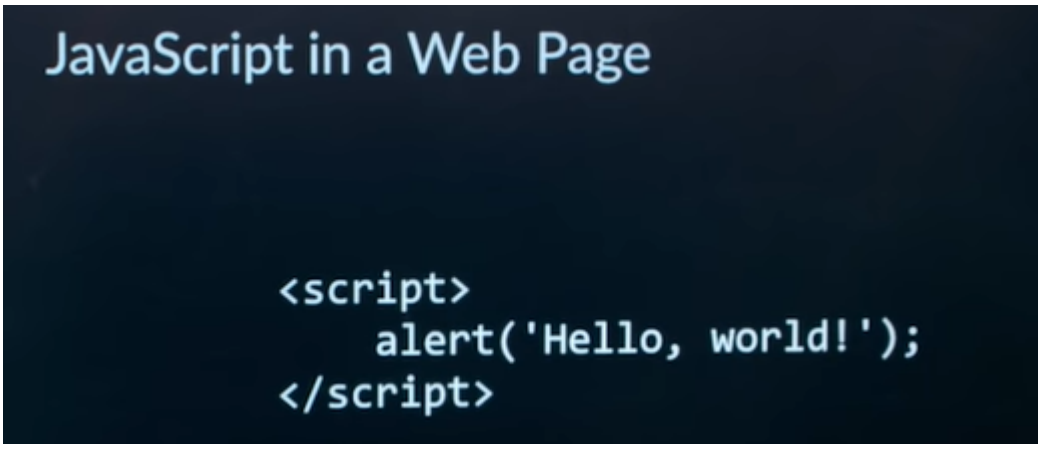
CS50's Web Programming with Python and JavaScript_Lecture 5_JS

April 9, 2023

0.0.1 JavaScript

0.0.2 ES6 – a version of JS

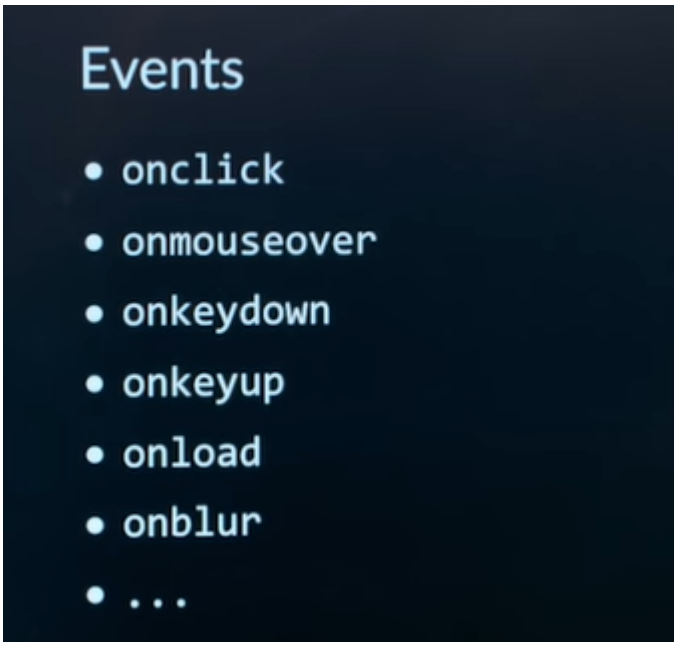


Show 'Hello' when clicked.

application.py | hello0.html | events.html ×

C: > Users > User > cs50WPPJ_5 > events.html > ...

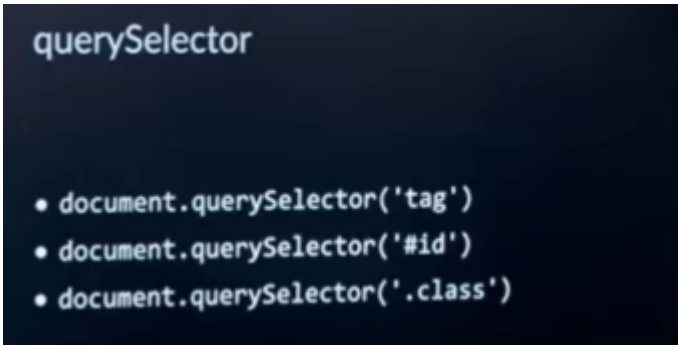
```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // Function to say hello
            function hello() {
                alert('Hello!');
            }

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1>Welcome!</h1>
        <button onclick="hello()">Click Here!</button>
    </body>
</html>
```



application.py | hello0.html | events.html | query.html

C: > Users > User > cs50WPPJ_5 > query.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // Function to change heading to say goodbye
            function hello() {
                document.querySelector('h1').innerHTML = 'Goodbye!';
            }

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1>Welcome!</h1>
        <button onclick="hello()">Click Here!</button>
    </body>
</html>
```



application.py | hello0.html | events.html | query.html | counter0.html

C: > Users > User > cs50WPPJ_5 > counter0.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            let counter = 0;

            function count() {
                counter++;
                document.querySelector('#counter').innerHTML = counter;
            }

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="counter">0</h1>
        <button onclick="count()">Click Here!</button>
    </body>
</html>
```

application.py | hello0.html | events.html | query.html | counter0.html | counter1.html

C: > Users > User > cs50WPPJ_5 > counter1.html > html > head

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            let counter = 0;

            function count() {
                counter++;
                document.querySelector('#counter').innerHTML = counter;

                if (counter % 10 === 0) {
                    alert(`Counter is at ${counter}!`);
                }
            }

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="counter">0</h1>
        <button onclick="count()">Click Here!</button>
    </body>
</html>
```

0.0.3 JS in a separate file

hello0.html | events.html | query.html | counter0.html | counter1.html | counter3.html

C: > Users > User > cs50WPPJ_5 > counter3.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script src="counter3.js"></script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="counter">0</h1>
        <button>Click Here!</button>
    </body>
</html>
```

0.0.4 the .js file

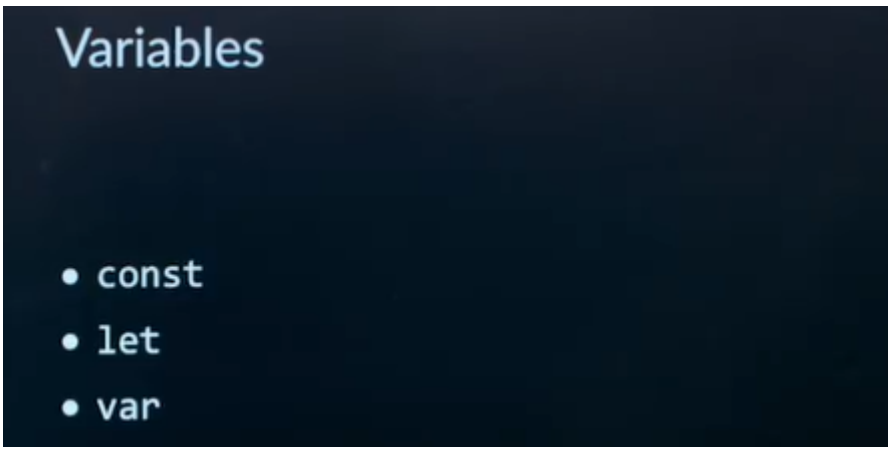
events.html query.html counter0.html counter1.html

C: > Users > User > cs50WPPJ_5 > JS counter3.js > ...

```
document.addEventListener('DOMContentLoaded', function() {
    document.querySelector('button').onclick = count;
});

let counter = 0;

function count() {
    counter++;
    document.querySelector('#counter').innerHTML = counter;

    if (counter % 10 === 0) {
        alert(`Counter is at ${counter}!`);
    }
}
```

Use of addEventListener() – a callback funtion

If there are multiple buttons, **querySelector()** will just select the first one.



let will exist in the innermost curly braces surrounding it. Outside of the curly brace where it was declared, it will not exist.

var will exist even in the outer curly braces.

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // This variable exists even outside the loop
            if (true) {
                var message = 'Hello!';
            }

            alert(message);
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1>Welcome!</h1>
    </body>
</html>
```

We will get an alert now.

Now, if we use **let** to define **message**, we will not see the alert.

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // This variable does not exist outside the loop
            if (true) {
                let message = 'Hello!';
            }

            alert(message);
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1>Welcome!</h1>
    </body>
</html>
```

Now, try to change **const** and it will not work.

counter1.html | counter3.html | counter3.js | variables0.html | variables1.html | variables2.html

C: > Users > User > cs50WPPJ_5 > variables2.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // The value of const variables cannot change
            const message = 'Hello!';
            message = 'Goodbye!';

            alert(message);
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1>Welcome!</h1>
    </body>
</html>
```

File | C:/Users/User/cs50WPPJ_5/variables2.html

iPad 768 × 1024 75% Online

Bookmark this tab

Elements Console Sources Network Performance

top Filter Default levels

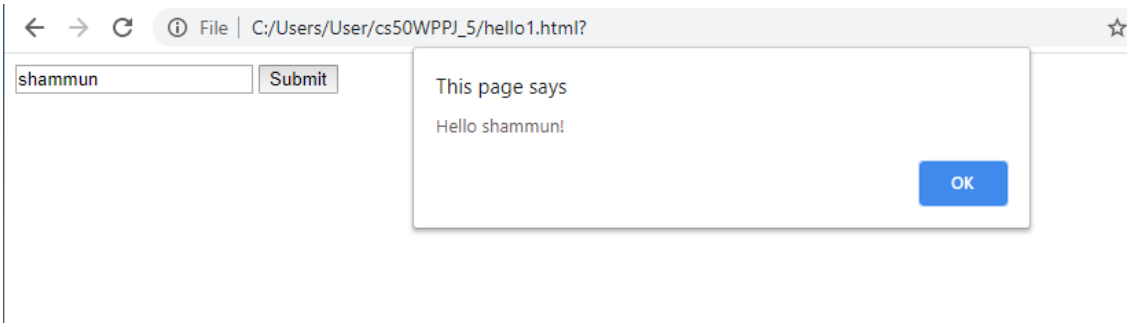
Uncaught TypeError: Assignment to constant variable. variables2.html:8
at variables2.html:8

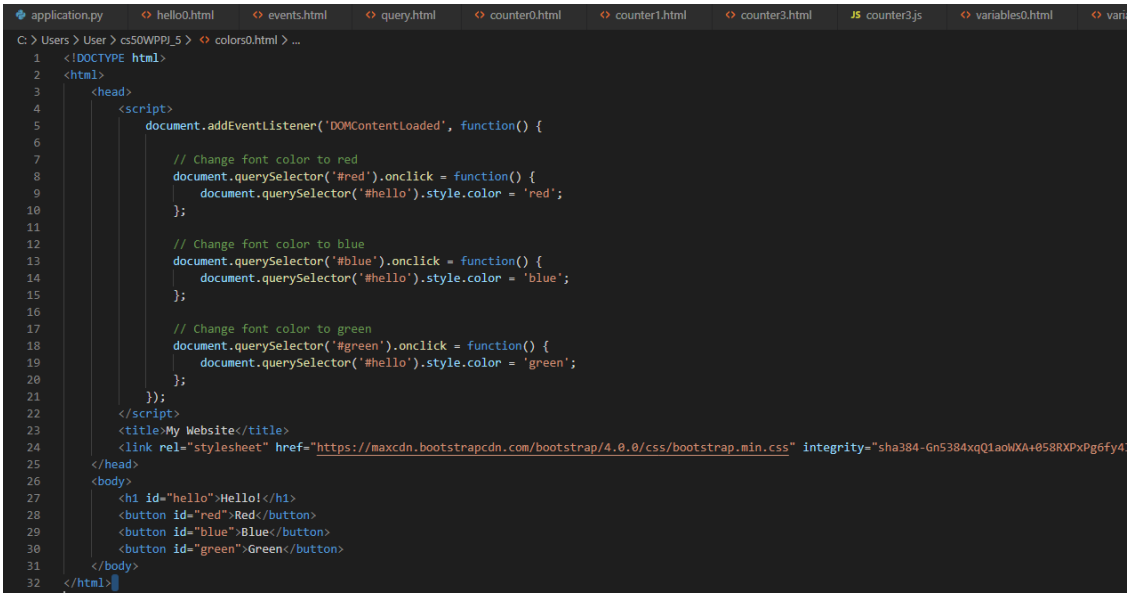
Welcome!

counter3.html | counter3.js | variables0.html | variables1.html | variables2.html | hello1.html

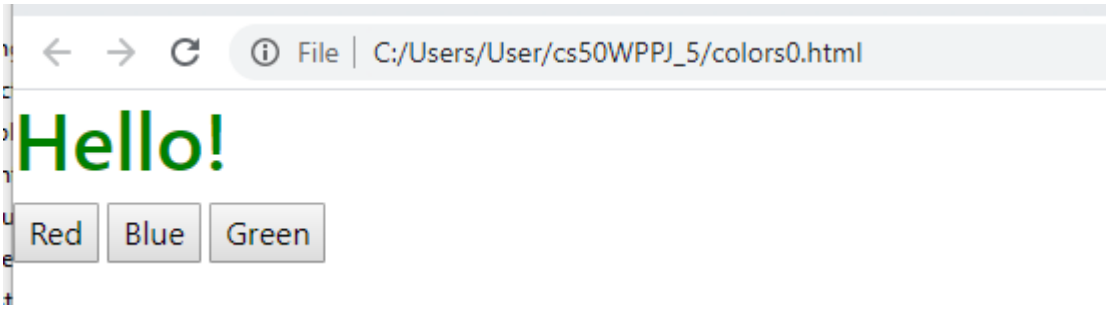
C: > Users > User > cs50WPPJ_5 > hello1.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', function() {
                document.querySelector('#form').onsubmit = function() {
                    const name = document.querySelector('#name').value;
                    alert(`Hello ${name}!`);
                };
            });
        </script>
        <title>My Website</title>
    </head>
    <body>
        <form id="form">
            <input id="name" autocomplete="off" autofocus placeholder="Name" type="text">
            <input type="submit">
        </form>
    </body>
</html>
```






```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', function() {

                // Change font color to red
                document.querySelector('#red').onclick = function() {
                    document.querySelector('#hello').style.color = 'red';
                };

                // Change font color to blue
                document.querySelector('#blue').onclick = function() {
                    document.querySelector('#hello').style.color = 'blue';
                };

                // Change font color to green
                document.querySelector('#green').onclick = function() {
                    document.querySelector('#hello').style.color = 'green';
                };
            });
        </script>
        <title>My Website</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4
    </head>
    <body>
        <h1 id="hello">Hello!</h1>
        <button id="red">Red</button>
        <button id="blue">Blue</button>
        <button id="green">Green</button>
    </body>
</html>
```



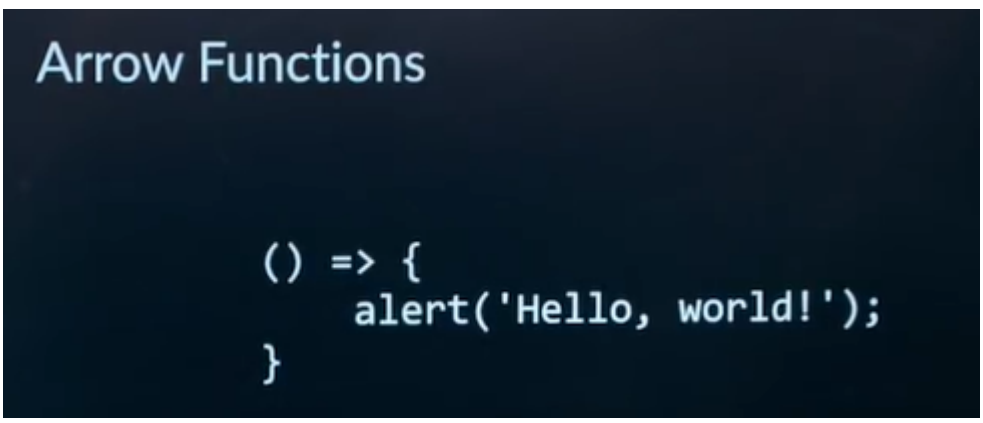
0.0.5 More efficient way

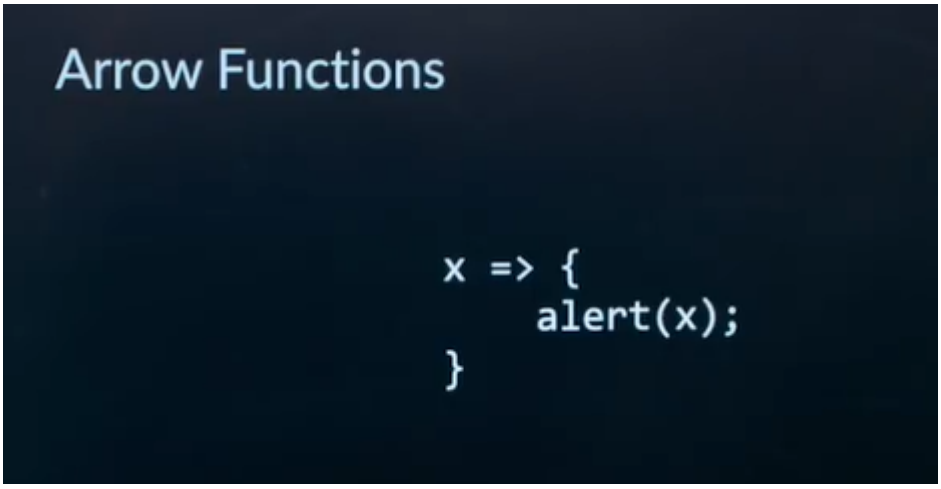
variables0.html | variables1.html | variables2.html | hello1.html | colors0.html | colors1.html

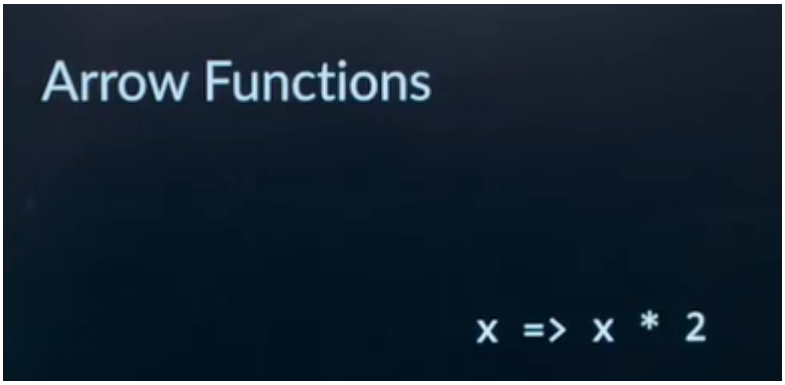
C: > Users > User > cs50WPPJ_5 > colors1.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', function() {

                // Have each button change the color of the heading
                document.querySelectorAll('.color-change').forEach(function(button) {
                    button.onclick = function() {
                        document.querySelector('#hello').style.color = button.dataset.color;
                    };
                });

            });
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="hello">Hello!</h1>
        <button class="color-change" data-color="red">Red</button>
        <button class="color-change" data-color="blue">Blue</button>
        <button class="color-change" data-color="green">Green</button>
    </body>
</html>
```

0.0.6 =>







The following code is the same as before using =>

variables1.html | variables2.html | hello1.html | colors0.html | colors1.html | colors2.html

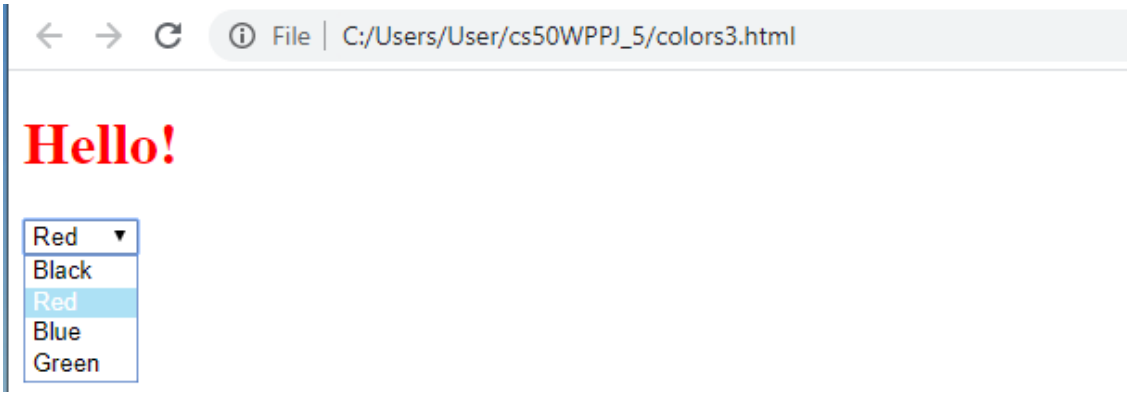
C: > Users > User > cs50WPPJ_5 > colors2.html > html > body > h1#hello

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', () => {

                // Have each button change the color of the heading
                document.querySelectorAll('.color-change').forEach(button => {
                    button.onclick = () => {
                        document.querySelector('#hello').style.color = button.dataset.color;
                    };
                });

            });
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="hello">Hello!</h1>
        <button class="color-change" data-color="red">Red</button>
        <button class="color-change" data-color="blue">Blue</button>
        <button class="color-change" data-color="green">Green</button>
    </body>
</html>
```

variables2.html | hello1.html | colors0.html | colors1.html | colors2.html | colors3.html

C: > Users > User > cs50WPPJ_5 > colors3.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', () => {

                // Change the color of the heading when dropdown changes
                document.querySelector('#color-change').onchange = function() {
                    document.querySelector('#hello').style.color = this.value;
                };

            });
        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="hello">Hello!</h1>
        <select id="color-change">
            <option value="black">Black</option>
            <option value="red">Red</option>
            <option value="blue">Blue</option>
            <option value="green">Green</option>
        </select>
    </body>
</html>
```



colors3.html | tasks0.html | colors0.html

C: > Users > User > cs50WPPJ_5 > tasks0.html > html > head > script > document.addEventListener('DOMContentLoad

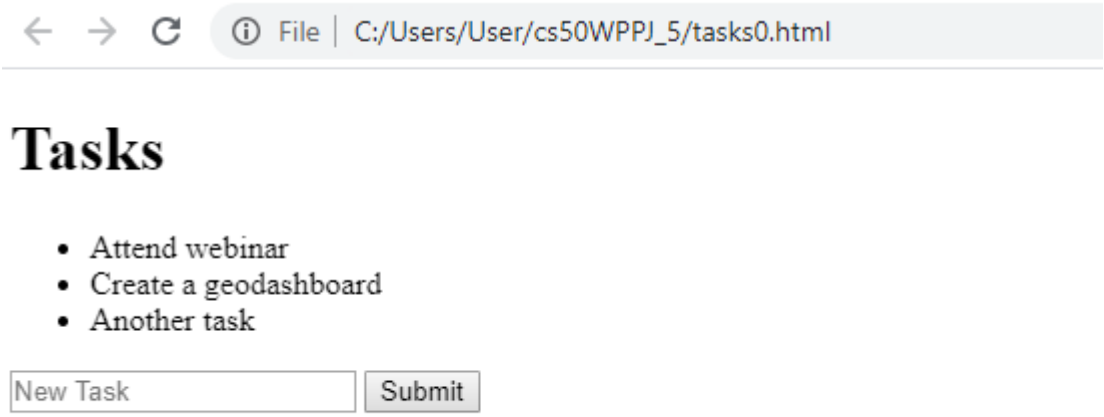
```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', () => {

                document.querySelector('#new-task').onsubmit = () => {

                    // Create new item for list
                    const li = document.createElement('li');
                    li.innerHTML = document.querySelector('#task').value;

                    // Add new item to task list
                    document.querySelector('#tasks').append(li);

                    // Clear input field
                    document.querySelector('#task').value = '';

                    // Stop form from submitting
                    return false;
                };

            });
        </script>
        <title>Tasks</title>
    </head>
    <body>
        <h1>Tasks</h1>
        <ul id="tasks">
        </ul>
        <form id="new-task">
            <input id="task" autocomplete="off" autofocus placeholder="New Task" type="text">
            <input type="submit">
        </form>
    </body>
</html>
```

File | C:/Users/User/cs50WPPJ_5/tasks0.html

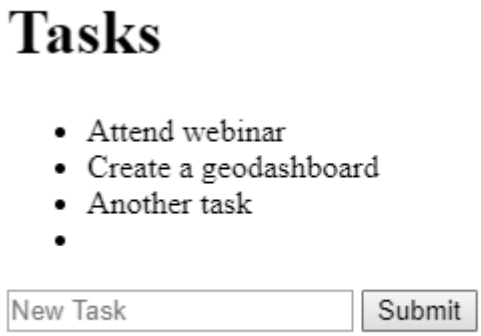
Tasks

- Attend webinar
- Create a geodashboard

Another task | Submit



The above code allows to submit blank task.



The following file (**tasks1.html**) prevents it.

Use **disabled** property.

colors3.html | tasks0.html | tasks1.html | colors0.html

C: > Users > User > cs50WPPJ_5 > tasks1.html > html > head > script > document.addEventListener('DOMContentLoaded') callback >

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', () => {

                // By default, submit button is disabled
                document.querySelector('#submit').disabled = true;

                // Enable button only if there is text in the input field
                document.querySelector('#task').onkeyup = () => {
                    document.querySelector('#submit').disabled = false;
                };

                document.querySelector('#new-task').onsubmit = () => {

                    // Create new item for list
                    const li = document.createElement('li');
                    li.innerHTML = document.querySelector('#task').value;

                    // Add new item to task list
                    document.querySelector('#tasks').append(li);

                    // Clear input field and disable button again
                    document.querySelector('#task').value = '';
                    document.querySelector('#submit').disabled = true;

                    // Stop form from submitting
                    return false;
                };

            });
        </script>
        <title>Tasks</title>
    </head>
    <body>
        <h1>Tasks</h1>
        <ul id="tasks">
        </ul>
        <form id="new-task">
            <input id="task" autocomplete="off" autofocus placeholder="New Task" type="text">
            <input id="submit" type="submit">
        </form>
    </body>
</html>
```

This code has some bugs, if we type something and then delete it, we can still submit empty task.

Check if the length of the task is > 0 or not.

colors3.html | tasks0.html | tasks1.html | tasks2.html ✕ | colors0.html

C: > Users > User > cs50WPPJ_5 > tasks2.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            document.addEventListener('DOMContentLoaded', () => {

                // By default, submit button is disabled
                document.querySelector('#submit').disabled = true;

                // Enable button only if there is text in the input field
                document.querySelector('#task').onkeyup = () => {
                    if (document.querySelector('#task').value.length > 0)
                        document.querySelector('#submit').disabled = false;
                    else
                        document.querySelector('#submit').disabled = true;
                };

                document.querySelector('#new-task').onsubmit = () => {

                    // Create new item for list
                    const li = document.createElement('li');
                    li.innerHTML = document.querySelector('#task').value;

                    // Add new item to task list
                    document.querySelector('#tasks').append(li);

                    // Clear input field and disable button again
                    document.querySelector('#task').value = '';
                    document.querySelector('#submit').disabled = true;

                    // Stop form from submitting
                    return false;
                };

            });
        </script>
        <title>Tasks</title>
    </head>
    <body>
        <h1>Tasks</h1>
        <ul id="tasks">
        </ul>
        <form id="new-task">
            <input id="task" autocomplete="off" autofocus placeholder="New Task" type="text">
            <input id="submit" type="submit">
        </form>
    </body>
</html>
```

0.0.7 Increments value from 0 in every single second and prints the value

colors3.html | tasks0.html | tasks1.html | tasks2.html | interval.html ✕ | colors0.html

C: > Users > User > cs50WPPJ_5 > interval.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            document.addEventListener('DOMContentLoaded', () => {
                setInterval(count, 1000);
            });

            let counter = 0;

            function count() {
                counter++;
                document.querySelector('#counter').innerHTML = counter;
            }

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="counter">0</h1>
    </body>
</html>
```

0.0.8 Local storage – retaining inside the local machine so that even if I close the file and open it later, I still can have the value from the last time

localStorage

```html
<!DOCTYPE html>
<html>
    <head>
        <script>

            // Set starting value of counter to 0
            if (!localStorage.getItem('counter'))
                localStorage.setItem('counter', 0);

            // Load current value of  counter
            document.addEventListener('DOMContentLoaded', () => {
                document.querySelector('#counter').innerHTML = localStorage.getItem('counter');

                // Count every time button is clicked
                document.querySelector('button').onclick = () => {
                    // Increment current counter
                    let counter = localStorage.getItem('counter');
                    counter++;

                    // Update counter
                    document.querySelector('#counter').innerHTML = counter;
                    localStorage.setItem('counter', counter);
                }
            });

        </script>
        <title>My Website</title>
    </head>
    <body>
        <h1 id="counter"></h1>
        <button>Click Here!</button>
    </body>
</html>
```

0.1 AJAX

Get data even without loading page

tasks0.html | tasks1.html | tasks2.html | interval.html | storage.html | application.py

C: > Users > User > cs50WPPJ_5 > currency > application.py > convert

```python
import requests

from flask import Flask, jsonify, render_template, request

app = Flask(__name__)


@app.route("/")
def index():
    return render_template("index.html")


@app.route("/convert", methods=["POST"])
def convert():

    # Query for currency exchange rate
    currency = request.form.get("currency")
    res = requests.get("https://api.fixer.io/latest", params={
        "base": "USD", "symbols": currency})

    # Make sure request succeeded
    if res.status_code != 200:
        return jsonify({"success": False})

    # Make sure currency is in response
    data = res.json()
    if currency not in data["rates"]:
        return jsonify({"success": False})

    return jsonify({"success": True, "rate": data["rates"][currency]})
```

tasks0.html | tasks1.html | tasks2.html | interval.html | storage.html | application.py | index.html | index.js

C: > Users > User > cs50WPPJ_5 > currency > templates > index.html > ...

```html
<!DOCTYPE html>
<html>
    <head>
        <script src="{{ url_for('static', filename='index.js') }}" type="text/javascript"></script>
        <title>Currency Converter</title>
    </head>
    <body>
        <form id="form">
            <input id="currency" autocomplete="off" autofocus placeholder="Currency" type="text">
            <input type="submit" value="Get Exchange Rate">
        </form>
        <br>
        <div id="result"></div>
    </body>
</html>
```

tasks0.html | tasks1.html | tasks2.html | interval.html | storage.html | application.py | index.html | index.js

C: > Users > User > cs50WPPJ_5 > currency > static > index.js > ...

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

    document.querySelector('#form').onsubmit = () => {

        // Initialize new request
        const request = new XMLHttpRequest();
        const currency = document.querySelector('#currency').value;
        request.open('POST', '/convert');

        // Callback function for when request completes
        request.onload = () => {

            // Extract JSON data from request
            const data = JSON.parse(request.responseText);

            // Update the result div
            if (data.success) {
                const contents = `1 USD is equal to ${data.rate} ${currency}.`
                document.querySelector('#result').innerHTML = contents;
            }
            else {
                document.querySelector('#result').innerHTML = 'There was an error.';
            }
        }

        // Add data to send with request
        const data = new FormData();
        data.append('currency', currency);

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

});
```

0.2 Web sockets

0.2.1 Socket.IO – a JS library

Real-time communication

0.2.2 Voting application

pip install flask_socketio

```python
import os
import requests

from flask import Flask, jsonify, render_template, request
from flask_socketio import SocketIO, emit

app = Flask(__name__)
app.config["SECRET_KEY"] = os.getenv("Drmhze6EPcv0fN_81Bj-nA")
socketio = SocketIO(app)

@app.route("/")
def index():
    return render_template("index.html")

@socketio.on("submit vote")
def vote(data):
    selection = data["selection"]
    emit("announce vote", {"selection": selection}, broadcast=True)
```

```html
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/socket.io/1.3.6/socket.io.min.js"></script
        <script src="{{ url_for('static', filename='index.js') }}"></script>
        <title>Vote</title>
    </head>
    <body>
        <ul id="votes">
        </ul>
        <hr>
        <button data-vote="yes">Yes</button>
        <button data-vote="no">No</button>
        <button data-vote="maybe">Maybe</button>
    </body>
</html>
```

```
ency   index.html C:\...\currency\...   index.js C:\...\currency\...   application.py C:\...\vote0   index.js C:\...\vote0\...
C: > Users > User > cs50WPPJ_5 > vote0 > static > index.js > ...
document.addEventListener('DOMContentLoaded', () => {

    // Connect to websocket
    var socket = io.connect(location.protocol + '//' + document.domain + ':' + location.port);

    // When connected, configure buttons
    socket.on('connect', () => {

        // Each button should emit a "submit vote" event
        document.querySelectorAll('button').forEach(button => {
            button.onclick = () => {
                const selection = button.dataset.vote;
                socket.emit('submit vote', {'selection': selection});
            };
        });
    });

    // When a new vote is announced, add to the unordered list
    socket.on('announce vote', data => {
        const li = document.createElement('li');
        li.innerHTML = `Vote recorded: ${data.selection}`;
        document.querySelector('#votes').append(li);
    });
});
```

0.2.3 Seeing vote totals and saving the vote even when we close and then reopen it

```
ency\...   application.py C:\...\vote0   index.js C:\...\vote0\...   application.py C:\...\vote1   index.html C:\...\vote1\...   index.js C:\...\
C: > Users > User > cs50WPPJ_5 > vote1 > application.py > ...
import os
import requests

from flask import Flask, jsonify, render_template, request
from flask_socketio import SocketIO, emit

app = Flask(__name__)
app.config["SECRET_KEY"] = os.getenv("SECRET_KEY")
socketio = SocketIO(app)

votes = {"yes": 0, "no": 0, "maybe": 0}


@app.route("/")
def index():
    return render_template("index.html", votes=votes)


@socketio.on("submit vote")
def vote(data):
    selection = data["selection"]
    votes[selection] += 1
    emit("vote totals", votes, broadcast=True)

```

```html
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/socket.io/1.3.6/socket.io.min.js"></script
        <script src="{{ url_for('static', filename='index.js') }}"></script>
        <title>Vote</title>
    </head>
    <body>
        <div>Yes Votes: <span id="yes">{{ votes["yes"] }}</span></div>
        <div>No Votes: <span id="no">{{ votes["no"] }}</span></div>
        <div>Maybe Votes: <span id="maybe">{{ votes["maybe"] }}<span></div>
        <hr>
        <button data-vote="yes">Yes</button>
        <button data-vote="no">No</button>
        <button data-vote="maybe">Maybe</button>
    </body>
</html>
```

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

    // Connect to websocket
    var socket = io.connect(location.protocol + '//' + document.domain + ':' + location.port);

    // When connected, configure buttons
    socket.on('connect', () => {

        // Each button should emit a "submit vote" event
        document.querySelectorAll('button').forEach(button => {
            button.onclick = () => {
                const selection = button.dataset.vote;
                socket.emit('submit vote', {'selection': selection});
            };
        });
    });

    // When a new vote is announced, add to the unordered list
    socket.on('vote totals', data => {
        document.querySelector('#yes').innerHTML = data.yes;
        document.querySelector('#no').innerHTML = data.no;
        document.querySelector('#maybe').innerHTML = data.maybe;
    });
});
```

[ ]: