Have you ever visited a website with

a map video or

social media posts contained in the webpage.

Now that you're learning about HTML,

you may have wondered how that's done.

Well, it's done using something called an iframe.

An iframe is HTML element that allows you to

place or embed content from

another website into a webpage.

It's defined using the iframe HTML tag.

How it works is that it embeds

another browsing instance inside the page.

What this means is that it runs the embedded webpage,

similar to how a webpage runs

in another tab of your web browser.

Therefore, it's running the HTML,

CSS, and JavaScript of the embedded webpage.

An iframe is often used to display adverts,

but it can also be used to

embed content from another website,

such as a social media post.

The content that's embedded is

defined using the src attribute.

The value of the attribute is the URL of the content.

It's also possible to set the width and

height of an iframe

using the width and height attributes.

While iframe is very useful.

It security has been a concern since

its inception because it's vulnerable

to malicious code and injection.

As previously mentioned,

a webpage can run JavaScript code.

It's important to ensure that you

trust the website you're embedding into your own.

Fortunately, there are some attributes that can be

applied to limit the behavior of the iframe.

The first attribute is the allow attribute,

which limits which browser

features the iframe can access.

There are many possible values

that can be set on the allow attribute.

For now, let's examine how to disable

camera and microphone access in an iframe.

On your iframe element,

you add the allow attribute and set its value to

camera none and microphone none. Now that was easy.

The second attribute to improve

security is the sandbox attribute.

Similar to the allow attribute,

there are many values that can be set.

The sandbox attribute limits behavior within the iframe,

such as preventing files from being

downloaded and preventing pop-up windows.

For now, the important thing to note is that when

the sandbox attribute is used with an empty value,

all sandbox restrictions will apply to the iframe.

Individual sandbox restrictions can

be removed by adding attributes.

For example, if you want to allow

file downloads in the iframe,

you would allow the allow downloads value.

While these restrictions help to keep your users secure,

there may still be security vulnerabilities in

your web browser that the embedded web page can exploit.

Therefore, it's always best to be cautious when using

iframes and ensure that you

trust the website that you are embedding.

You are a developer at the Little Lemon Tree company and your team leader has asked you to embed a map in the web page you are developing. You have learned that there are various important security issues when using iframes. Select the ways that these issues can be addresed. Check all that apply.



Recommend that iframes are not used.



You can use the allow attribute to limit which browser features the iframe can access.

Correct

That's correct! The allow attribute can be set, for example, to not allow the iframe camera or microphone access.



The sandbox attribute can prevent files from being downloaded.

Correct

That's correct! The sandbox attribute can also prevent popup windows.

**iFrame sandbox cheat sheet**

The *<iframe>* is the inline frame element that embeds an HTML page into another page. Apart from the global attributes, which can be a part of the *iframe*, major element-specific attributes are listed below.

**allow**

It specifies what features or permissions are available to the frame, for instance, access to the microphone, camera, other APIs and so on. For example:

* *allow="fullscreen”* the fullscreen mode can be activated
* *allow=“geolocation”* lets you access the user’s location

To specify more than one feature, a semicolon-separator should be used between features. For example, the following would allow using the camera and the microphone:

*<iframe src="https://example.com/…" allow="camera; microphone"> </iframe>*

**name**

The name for the *<iframe>*. For example: *<iframe name = "My Frame" width="400" height="300"></iframe>*

**height**

It specifies the height of the frame. The default value is 150, in terms of CSS pixels. width

**width**

Specifies the width of the frame, in terms of CSS pixels. The default value is 300 pixels.

**referrerpolicy**

A referrer is the HTTP header that lets the page know who is loading it. This attribute indicates which referrer information to send when loading the frame resource. The common values are:

* *no-referrer* The referrer header will not be sent.
* *origin* The referrer will be limited to the origin of the referring page
* *strict-origin* The origin of the document is sent as the referrer only when using the same protocol security level (HTTPS to HTTPS)

**sandbox**

To enforce greater security, a sandbox applies extra restrictions to the content in the *<iframe>*. To lift particular restrictions, an attribute value (permission token) is used. The common permission tokens are listed below:

* *allow-downloads* Allows the user to download an item
* *allow-forms* Allows the user to submit forms
* *allow-modals* The resource can open modal windows
* *allow-orientation-lock* Lets the resource lock the screen orientation
* *allow-popups* Allows popups to open
* *allow-presentation* Allows a presentation session to start
* *allow-scripts* Lets the resource run scripts without creating popup windows

Note that when the value of this attribute is empty, all restrictions are applied. To apply more than one permission, use a space-separated list. For example, the following would allow form submission and scripts while keeping other restrictions active:

<iframe src="my\_iframe\_sandbox.html" sandbox="allow-forms allow-scripts">

</iframe>

## src

The URL of the page to embed in the <iframe>. Using the value about:blank would embed an empty page.

## srcdoc

Let's you specify the inline HTML to embed in the <iframe>. When defined, this attribute would override the src attribute. For instance, the following code will display "My inline html" in the frame, instead of loading my\_iframe\_src.html.

<iframe src="my\_iframe\_src.html" srcdoc="<p>My inline html</p>" >

</iframe>

## loading

This attribute let's you specify if the iframe should be loaded immediately when the web page loads (eager) or loaded when iframe is displayed in the browser (lazy). This allows you to defer loading iframe content if it is further down your web page and outside of the display area when the page is initially loaded.

<iframe src="my\_iframe\_src.html" loading="lazy" >

</iframe>

## title

This attribute let's you add a description to the iframe for accessibility purposes. The value of this attribute should accurately describe the iframe's content.

<iframe src="history.html" title="An embedded document about the history of my family" >

</iframe>

The Little Lemon restaurant has partnered with

another restaurant for cross promotional purposes.

The other restaurant wants to be

able to change the images in

the campaign throughout the coming year

based on their current promotions.

To support this, I've been asked to setup

an iFrame on the main page of the website.

The other restaurants specifically asked for

an iFrame to be used because it

means that they can update their promotional image on

the Little Lemon website

without needing to contact me first.

In this video, I'm going to demonstrate how to do that.

I've opened the index.html in Visual Studio Code.

First, I add the iFrame element.

The other restaurant didn't provide

the URL for the promotional image.

For now, I'll embed

a placeholder image that they did provide.

To do this, I add an SRC attribute to

the iFrame element and set its value to placeholder.png.

The image will always be

320 pixels by 128 pixels in size.

I'll need to set the iFrame to the same size.

I add the width attribute to

the iFrame element and set its value to 320.

I then add the height attribute and set its value to 128.

I saved the file and open it in the web browser.

The image is correctly placed on the webpage.

When the other restaurant provides

the final URL of the advertisement,

I can update the SRC attribute of the element.

However, the Little Lemon restaurant has expressed

security concerns about embedding content,

especially now that they have

online ordering and customers

interacting with your website.

Their concerns are valid as there are

many security vulnerabilities with

embedding content from another website.

To address these concerns,

I've decided to sandbox and limit

the capabilities of the iFrame element.

I opened my index.html file again.

I then add the sandbox attribute to the iFrame element.

This will prevent any JavaScript files running within

the iFrame and restrict

many browser capabilities within the iFrame context,

such as submitting forms and allowing pop-up windows.

Next, I'll allow attribute.

The allow attribute accepts

multiple values and the values

are separated by semi-colons.

I want to disable several

browser features for the iFrame.

These include payment, microphone, and camera access.

In the allow attribute,

I first add the word payment,

followed by a space character then a single quote,

the word none, and another

single quote to disabled payments.

I then add a semicolon.

Next, I add the word camera,

and again a space followed by

the word none in single quotes.

This will disable camera axis,

then add another semicolon.

Finally, I add the word microphone,

and again add a space followed by

the word none in single quotes.

This will disable microphone access.

I save the file and open it in the web browser.

Great, the iFrame still works and it's now more secure.

The iFrame I created is now secure things.

The restrictions I've placed on it.

First, I added the sandbox attribute to restrict

certain browser capabilities and then I use

the allow attribute to disable some

potentially unsecured browser features.

Now that they know it's secure,

the other restaurant is going to

implement the same code on their website.

Hopefully this will lead to a lot more business

for both restaurants.

A small business is planning to embed advertisements on their website and you have to create an iframe to set up an image on their website. However, there are some security concerns when it comes to embedded content. What can you do to minimize those concerns? Select all that apply.



Add the src attribute.



Add the sandbox attribute to the iframe element.

Correct

That's correct! This will prevent JavaScript files from running within the iframe and it restricts browser capabilities within the iframe context.



Use the allow attribute.

Correct

That's correct! Using the allow attribute means that you can disable browser features for the iframe like payment, microphone and camera access.