Network

    How internet works

internet is a wire, servers connect to internet
server hard drives contains files(webpages)
ip address <DNS> website names
client - digital subscriber line - isp - webpage - server
computers break info into packets
routers in between diff network layers

webpage files - html, css, js, img, video
server stores files
web browser - renders files
html - defines contents of website, the foundation
css - static look, style
js - functionalities of webpages
http(hyper text transfer protocol)
https(hyper text transfer protocol secure) -
ssl - security sockets layer
tls - transport layer security
web browser ask server for specific html file, server sent back
web browser will then parse the html file, see if additional info
Domain name resolution - DNS(domain name system)
server stores webpages and know the ip address but not domain name

    CND(content delivery network)

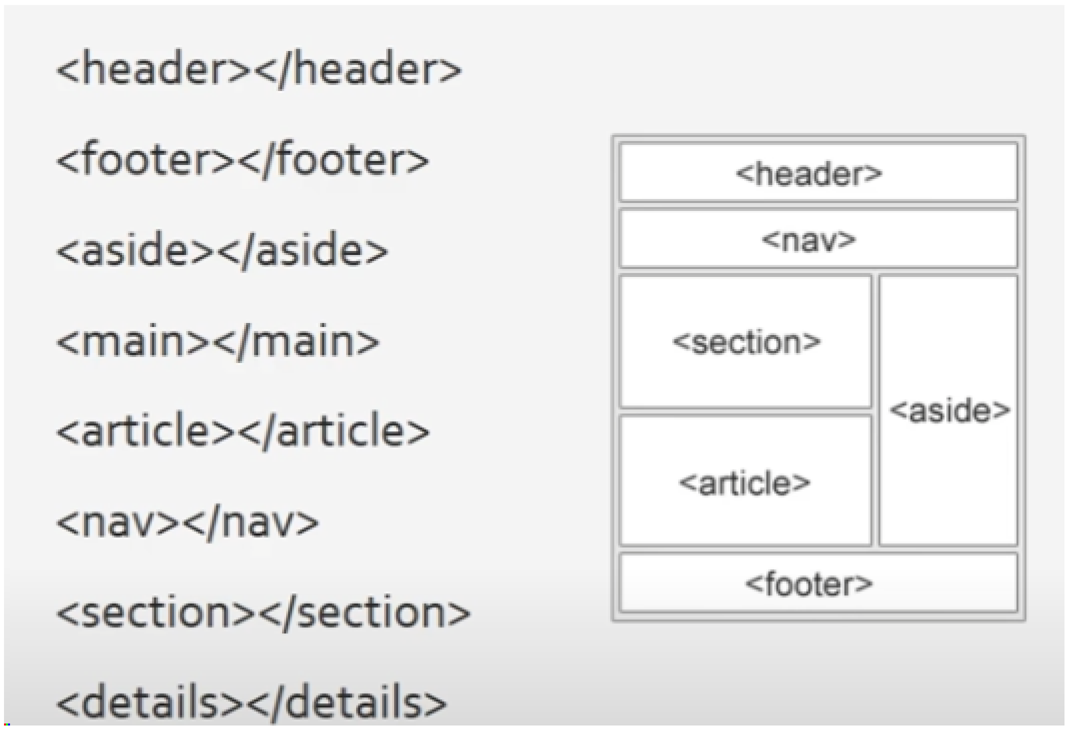
late 90s, speed up the static html content for users
now, whenever http traffic is served
bring contents closer to user, deploys servers, better performance, reduce load time
pop, point of presence, server location, edge servers - reverse proxy
static content cache
modern cdns can change format
DNS-BASED routing - each pop has own ip address
anycast - same ip address
ssl - security sockets layer
tls - transport layer security
tls handshake - a process that establishes secure connect,
between client and server by exchanging messages to verify each other's identities and agree on encryption keys
terminates tls connect at edge server, reduce user latency to establish

modern application dynamic uncacheable contents
modern cdn, security(ddos) and availability (highly distributed)

    http requests
Api - application programming interface - point of contact, allow connection between computers or programs - deliver requests and response - enable connections using set of rules
REST(REpresentation State Transfer)
create read update delete
http://example.com/fruits
GET - retrieves a list of the resource, acts on the entire resource - list of fruits
-not secured, sent in the form, visible in url, sql injection
POST - create a new resource, submit data to server
PUT - update the resource
DELETE - delete the resource
http requests/responses cycle:
loading pages, form submit,
ajax calls(asynchronous javascript and XML), asynchronous request, update parts of the page without a full reload.
every http request is stateless, independent

header and body
method - path - protocol
response header - server, set cookie, content type, content-length, date
request header - cookies, authorization, user-agent, content type, length
general header - request url, request method, status code, remote address, referrer policy
content type - html, text, json
status codes
1xx informational, request received / processing
2xx success, successfully received, understood and accepted
3xx redirect, further action must be taken / redirect
4xx client error, request does not have what it needs
5xx server error, server failed to fulfill an apparent valid request
200 - ok,everything is fine, request and respose
201 - ok created, fine, something is created
301 - moved to new url, redirection
304 - not modified(cached version),resource not updated
400 - bad request, not sending correct data

```
401 - unauthorized, miss token, can’t get in
403 - server refused to process the requests
404 - not found, something doesn’t exist
500 internal server error, anything, something on the server side

HTML
<> angle brackets
/ forward slash
<!DOCTYPE html>
<head>
      <title></title>
</head>
<body>
      <h1></h1> start tag, end tag
      <p></p>
      <br />
      <strong>
      <em>
      <a href=”http://google.com” target=”_blank”>
      Open google in new tab
      Or we can also do <a href=”local.html”>
</body>
```



```
<header></header>
<footer></footer>
<aside></aside>
<main></main>
<article></article>
<nav></nav>
<section></section>
<details></details>
```

```
Control-u to see html page being parsed
```

F12 to console

Shift + alt + a to multi line comment

Control + / to current line comment

Lorem

Padding,border,margin

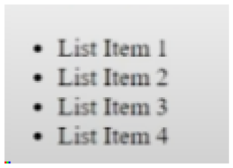
Inline elements - do not start new line, take only necessary width

Block elements - start on a new line, take full width available

Block level:div,h1 - h6, p, form

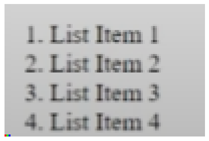
Inline level: <span>, <img>, <a>

Ul and li, unordered list



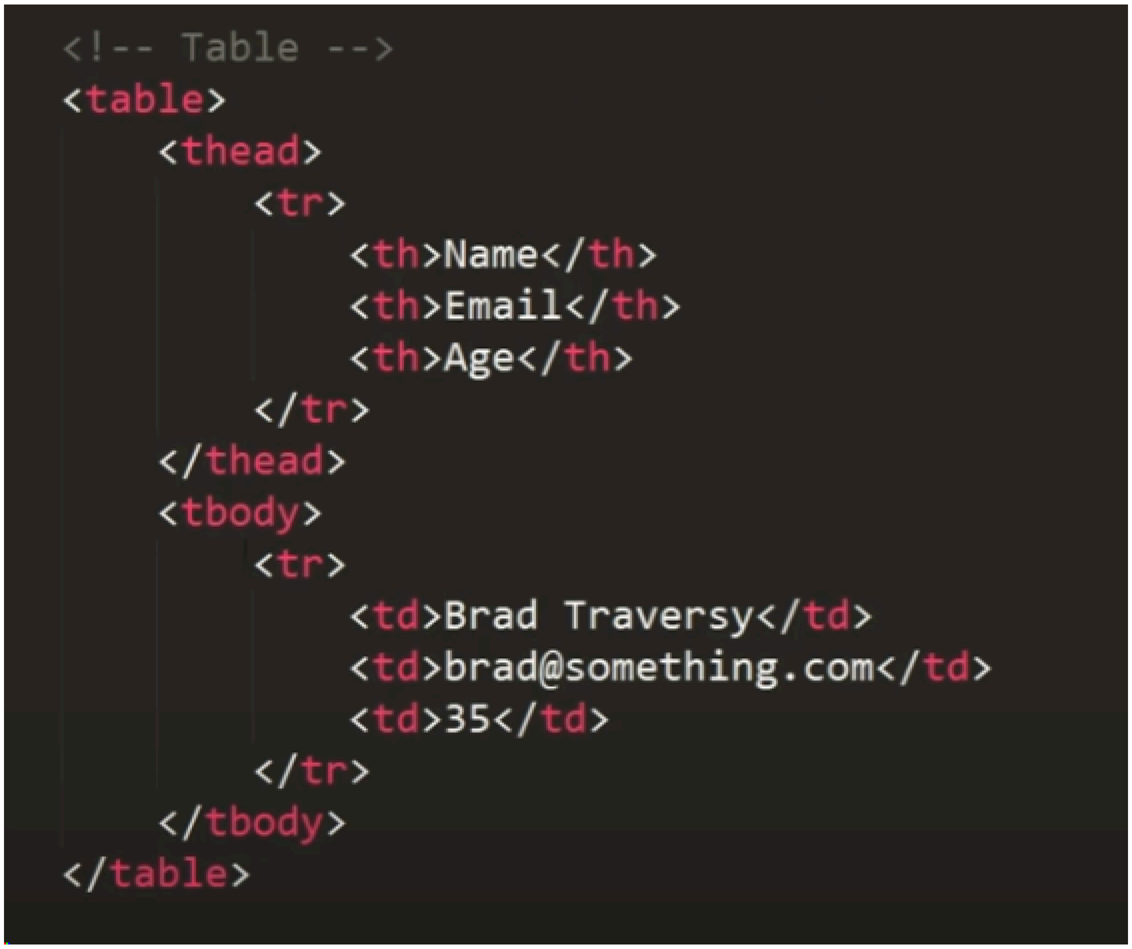
- List Item 1
- List Item 2
- List Item 3
- List Item 4

Ol and li, ordered list



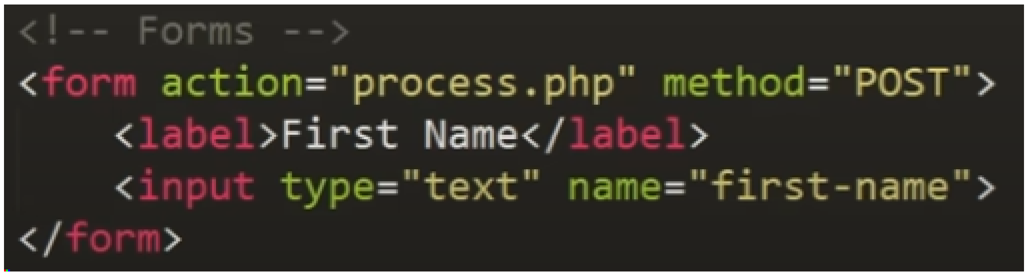
1. List Item 1
2. List Item 2
3. List Item 3
4. List Item 4

Table, thead, tbody, tr(table row), th(column headers), td(data)

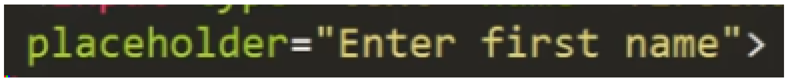


```html
<!-- Table -->
<table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Email</th>
            <th>Age</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Brad Traversy</td>
            <td>brad@something.com</td>
            <td>35</td>
        </tr>
    </tbody>
</table>
```

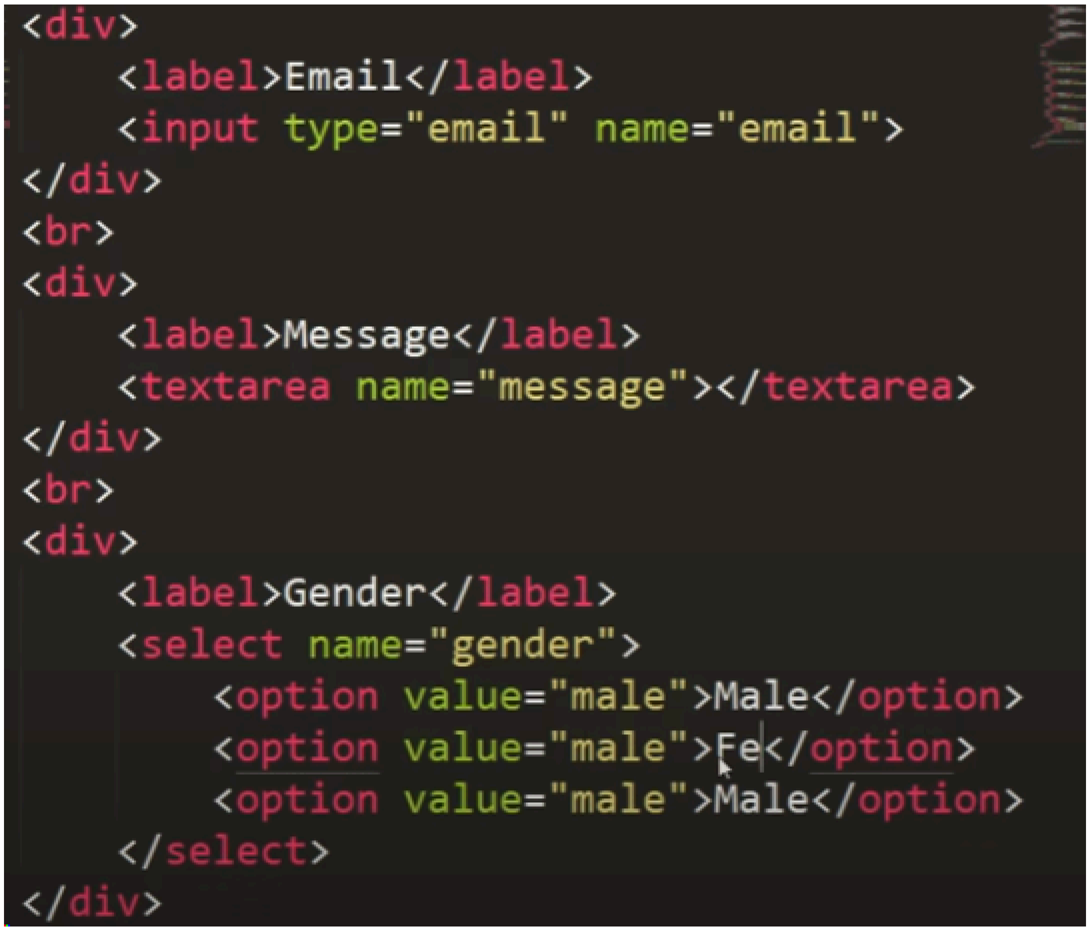
Form, label, input



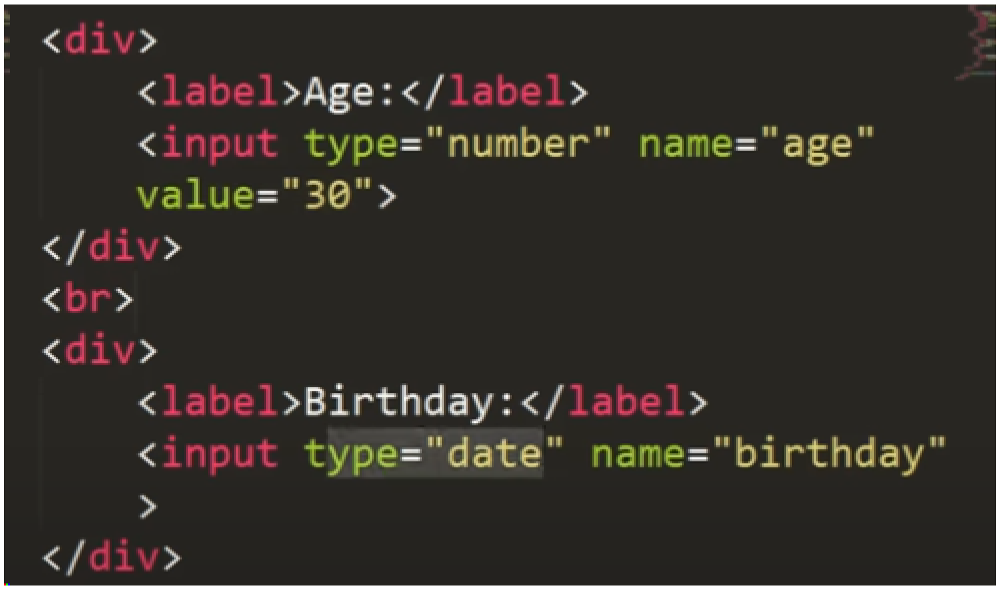
```html
<!-- Forms -->
<form action="process.php" method="POST">
    <label>First Name</label>
    <input type="text" name="first-name">
</form>
```



```html
placeholder="Enter first name">
```

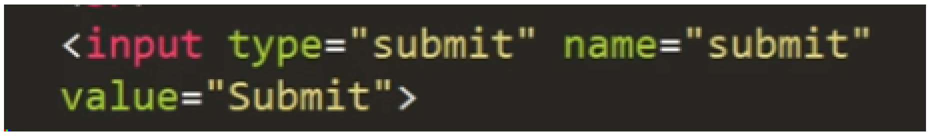


```html
<div>
    <label>Email</label>
    <input type="email" name="email">
</div>
<br>
<div>
    <label>Message</label>
    <textarea name="message"></textarea>
</div>
<br>
<div>
    <label>Gender</label>
    <select name="gender">
        <option value="male">Male</option>
        <option value="male">Fe</option>
        <option value="male">Male</option>
    </select>
</div>
```



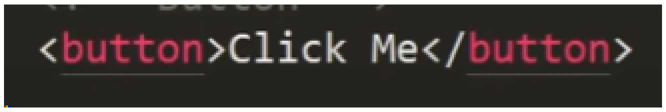
```html
<div>
    <label>Age:</label>
    <input type="number" name="age"
    value="30">
</div>
<br>
<div>
    <label>Birthday:</label>
    <input type="date" name="birthday"
    >
</div>
```

submit botton inside form

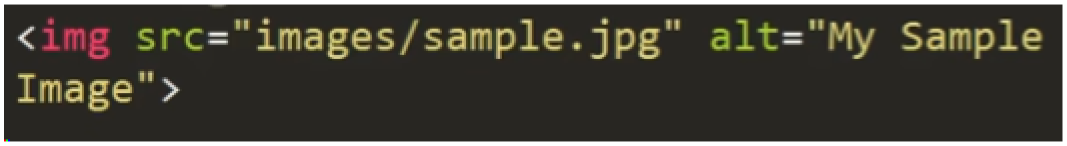


```html
<input type="submit" name="submit"
value="Submit">
```

Button Outside form

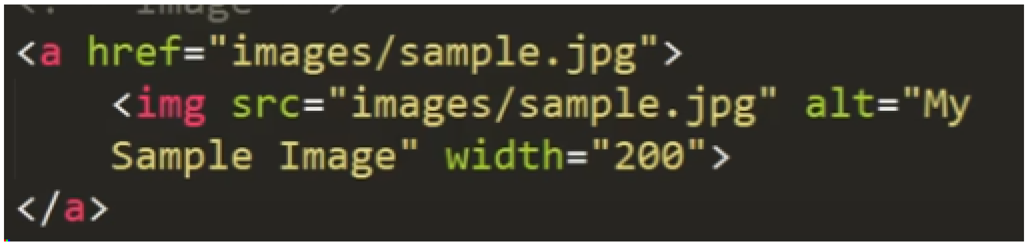


```html
<button>Click Me</button>
```

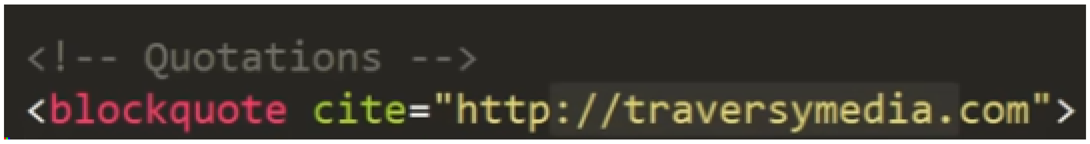


```html
<img src="images/sample.jpg" alt="My Sample
Image">
```

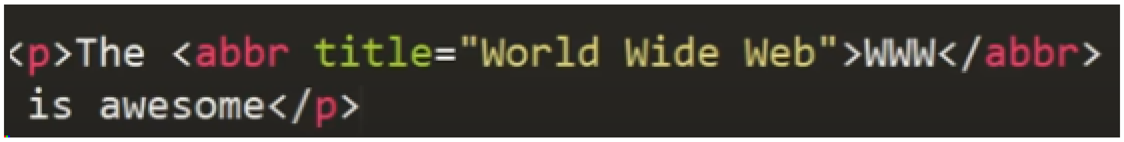
Now if I click image it will open up the image



```html
<a href="images/sample.jpg">
    <img src="images/sample.jpg" alt="My
    Sample Image" width="200">
</a>
```

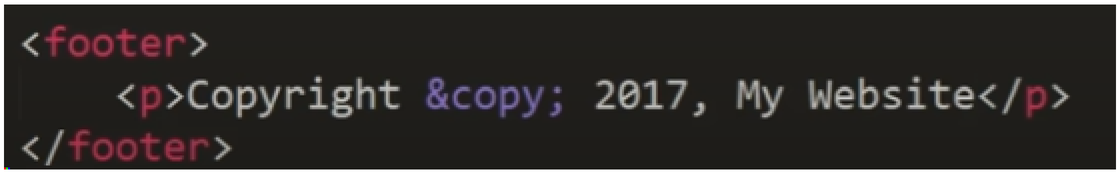


```html
<!-- Quotations -->
<blockquote cite="http://traversymedia.com">
```



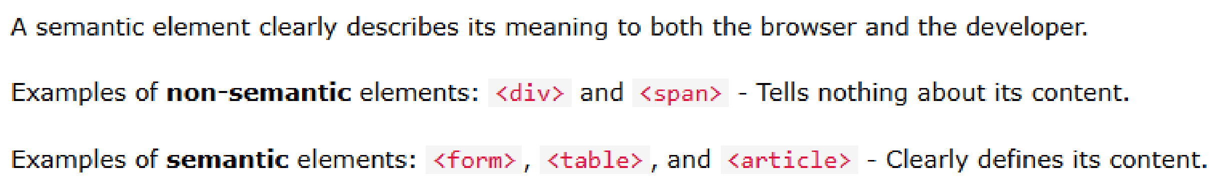
```html
<p>The <abbr title="World Wide Web">WWW</abbr>
 is awesome</p>
```

<cite></cite>



```html
<footer>
    <p>Copyright &copy; 2017, My Website</p>
</footer>
```

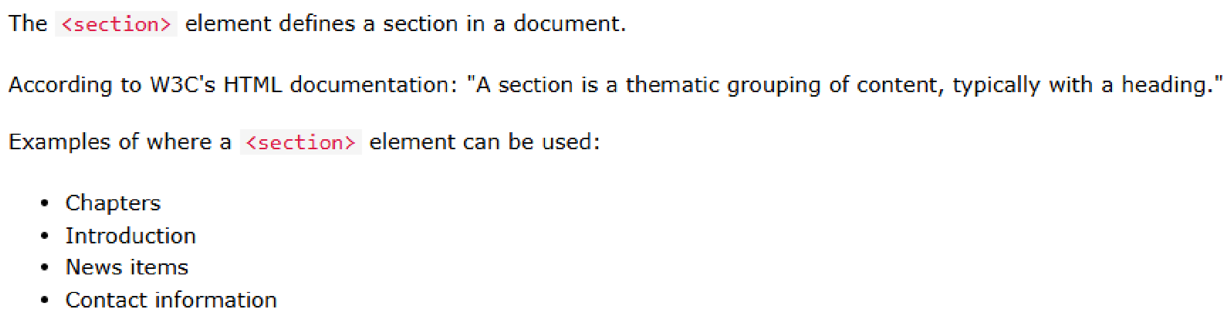
Semantic web allows data to be shared and reused across applications, enterprises, and communities.



A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

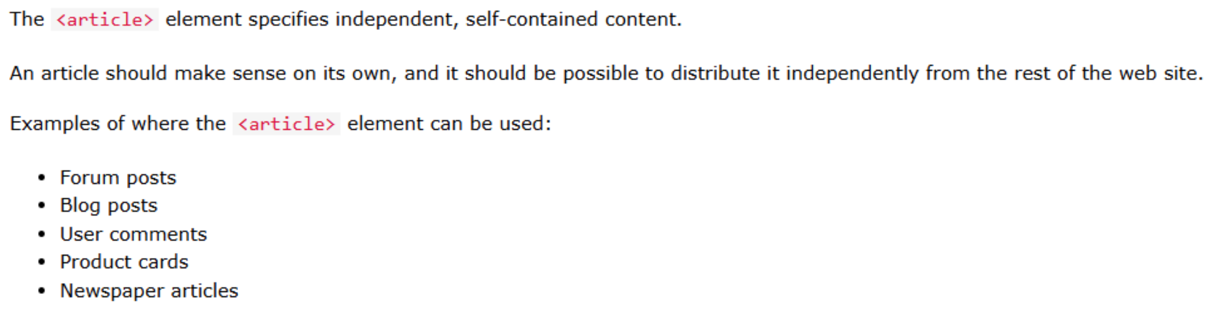


The `<section>` element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

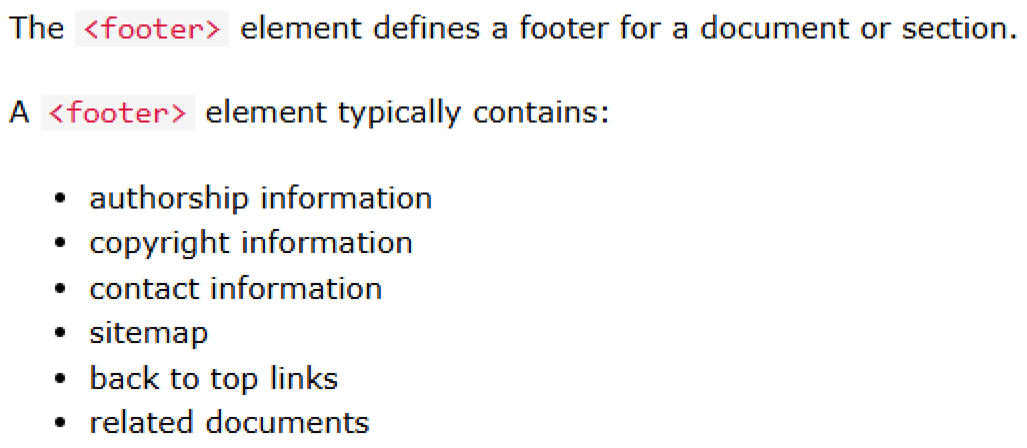


The `<article>` element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the `<article>` element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles



The `<footer>` element defines a footer for a document or section.

A `<footer>` element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

- **know how to look up attributes on MDN**

DOM - a programming interface represents web documents as a tree structure of nodes and objects.

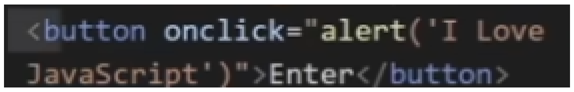
DOM(document object model) manipulation - a javascript features that allows developers to change the structure, content, and style of a web page.

Selectors - the selector api, selest / retrieve html element within a docu using javascript

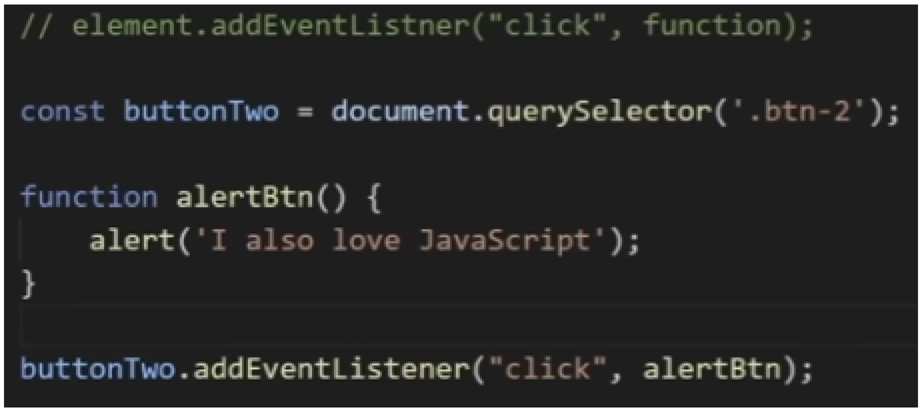
document.GetElementById() - unique identifier, GetElementByClassName, getElemensByTagName, querySelector - return first element that matches the selector, querySlectorAll,

Event listeners - set up a function that will be called whenever the specific event is delivered tot he target. click, mouseover,scroll

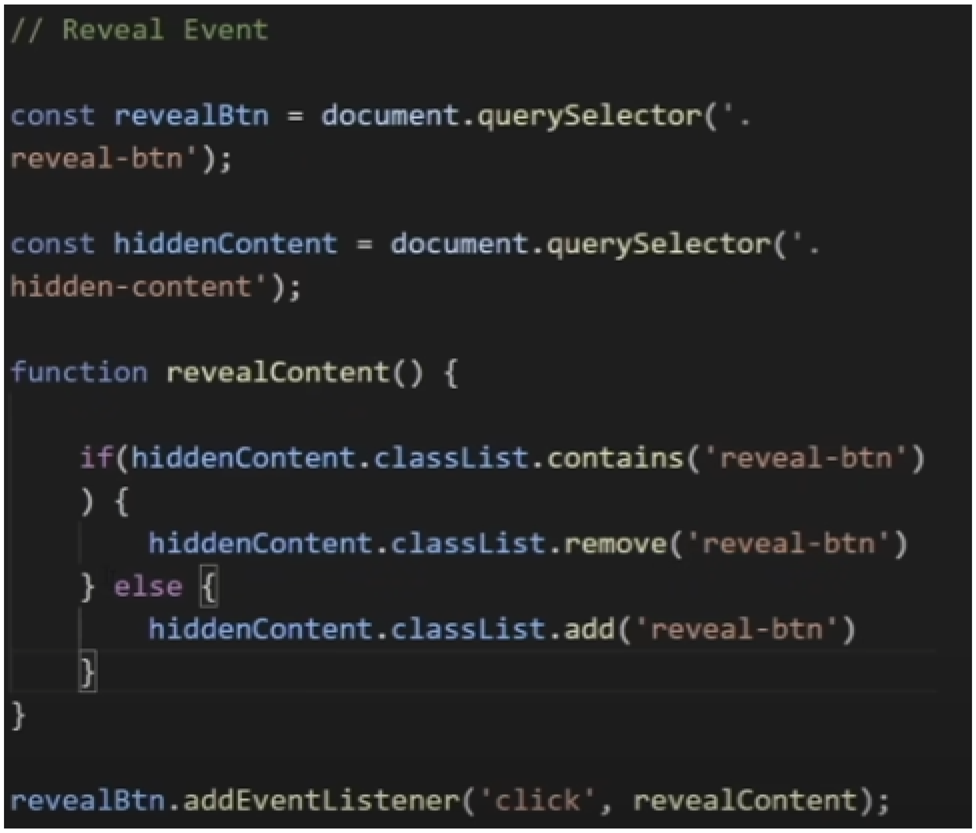
Direct into the html or use element.addEventListener(),



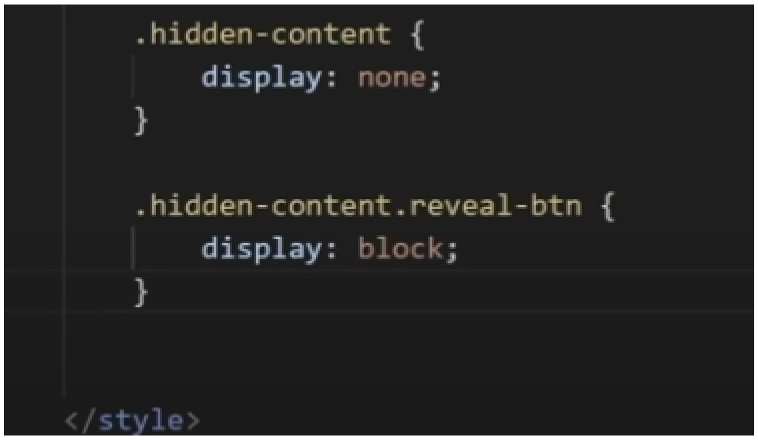
```
<button onclick="alert('I Love
JavaScript')">Enter</button>
```



```
// element.addEventListner("click", function);

const buttonTwo = document.querySelector('.btn-2');

function alertBtn() {
    alert('I also love JavaScript');
}

buttonTwo.addEventListener("click", alertBtn);
```



```
// Reveal Event

const revealBtn = document.querySelector('.
reveal-btn');

const hiddenContent = document.querySelector('.
hidden-content');

function revealContent() {

    if(hiddenContent.classList.contains('reveal-btn')
    ) {
        hiddenContent.classList.remove('reveal-btn')
    } else {
        hiddenContent.classList.add('reveal-btn')
    }
}

revealBtn.addEventListener('click', revealContent);
```



```
    .hidden-content {
        display: none;
    }

    .hidden-content.reveal-btn {
        display: block;
    }


</style>
```

Event Propagation - process by which an event travels through the DOM when an event occurs, how event travel through the dom tree to it's element

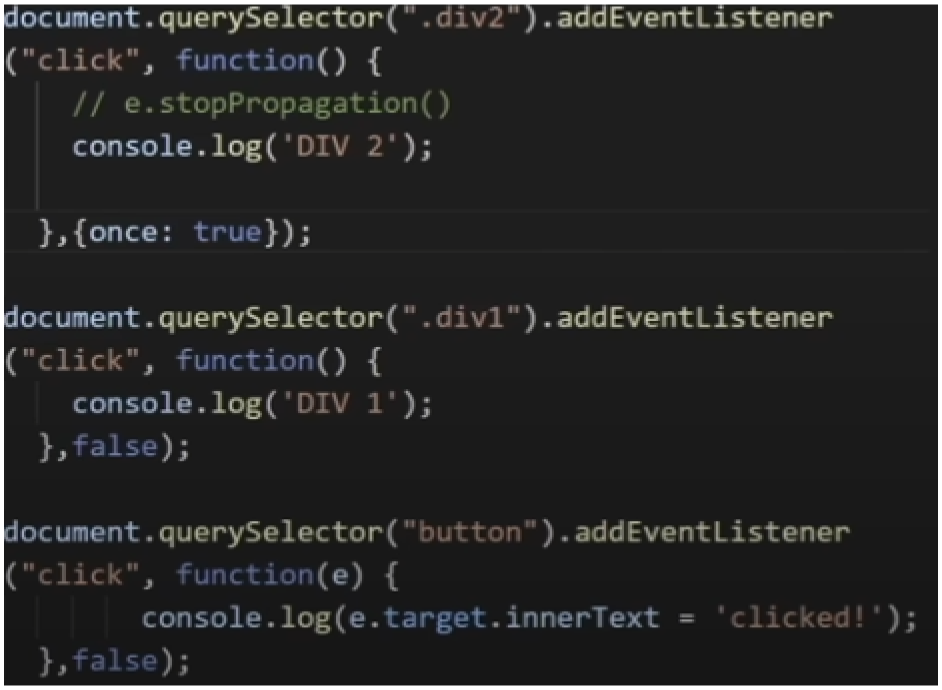
Event capturing - starts at root of the DOM and travels down to the tagret

target

Event bubbling - starts at target and travels back up to the root of DOM e represent the event object

e.stopPropagation(),

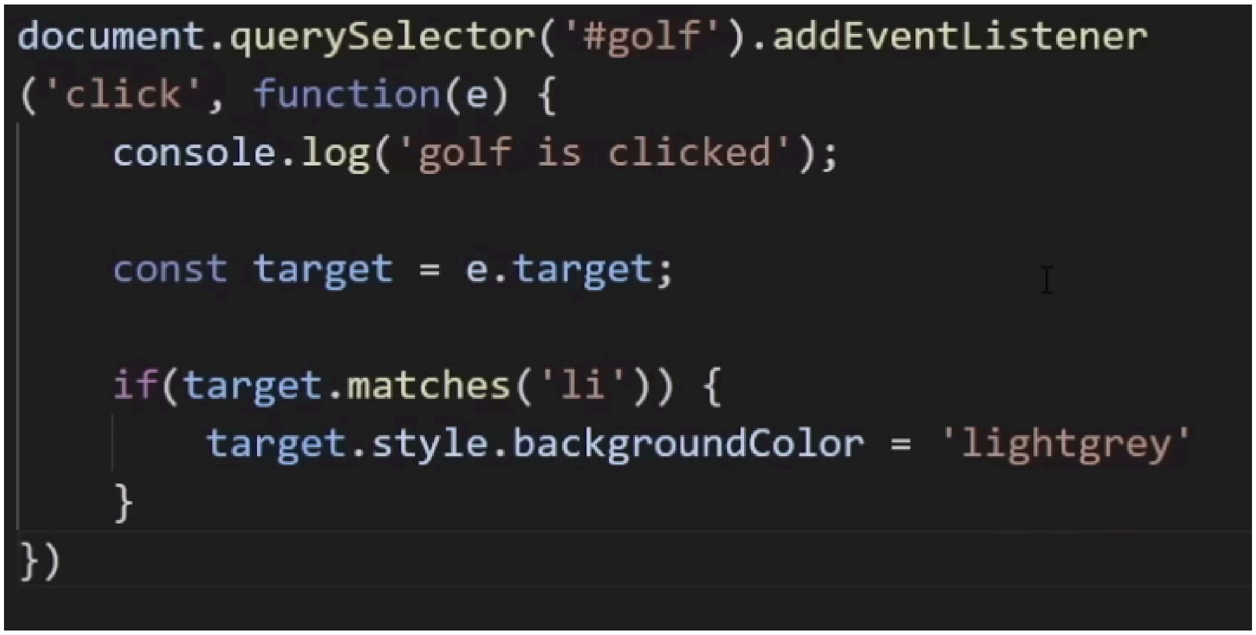
e.preventDefault() - the default event actions will not occur. Click on a submit botton, prevent it from submitting a form; click on a link, prevent the link from following the url;

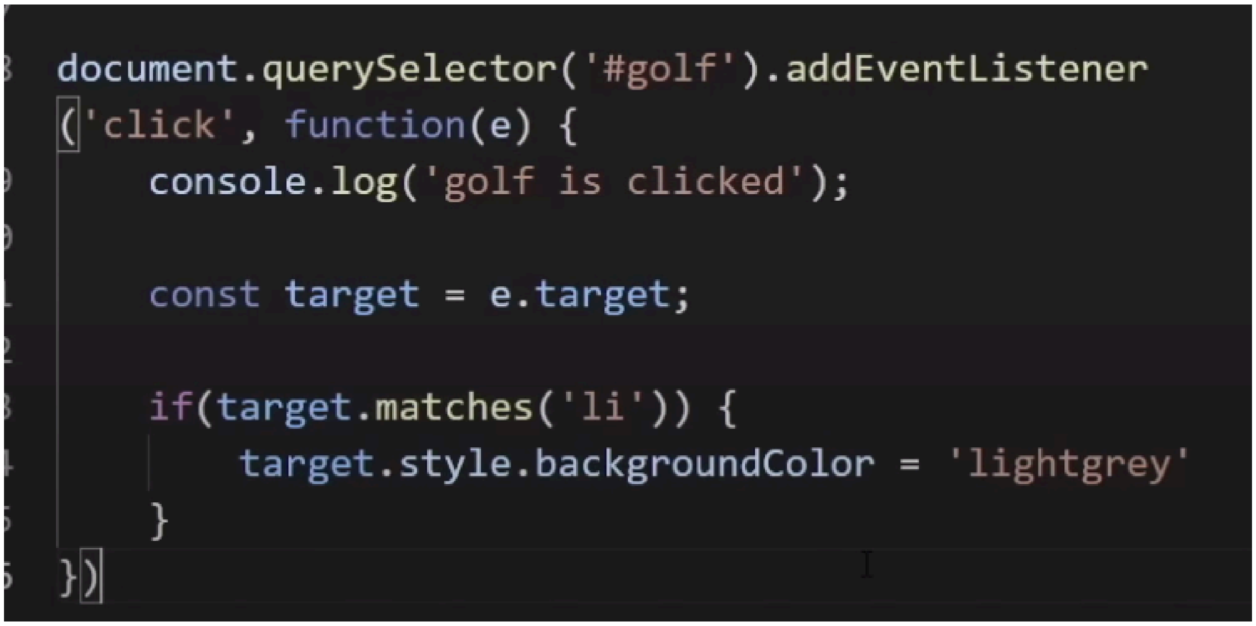


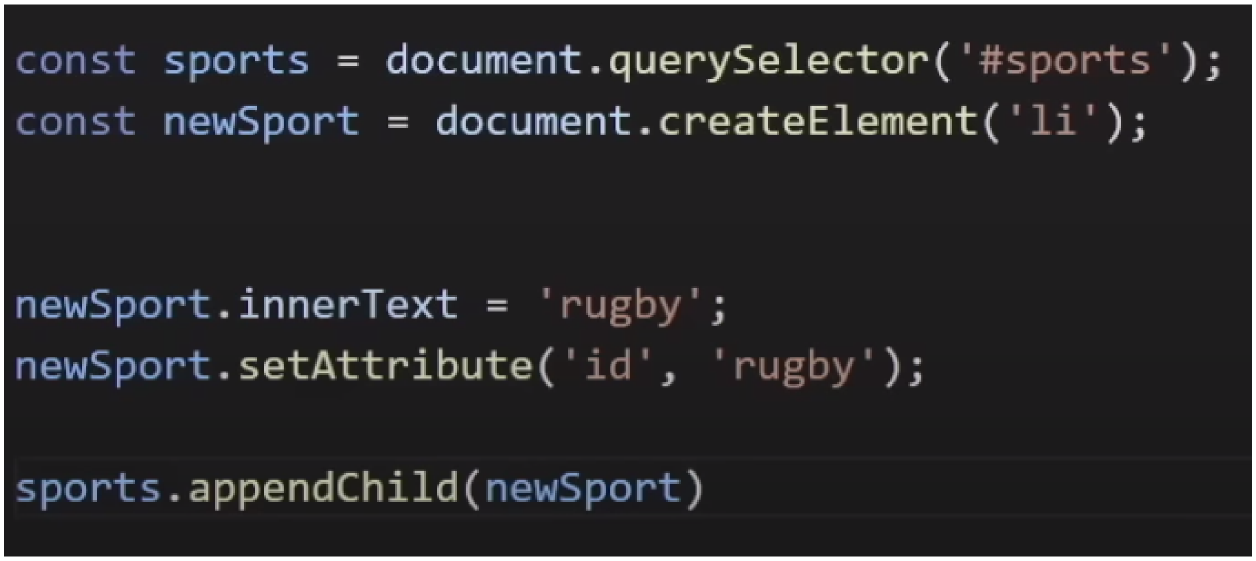
```
document.querySelector(".div2").addEventListener
("click", function() {
    // e.stopPropagation()
    console.log('DIV 2');

  },{once: true});

document.querySelector(".div1").addEventListener
("click", function() {
    console.log('DIV 1');
  },false);

document.querySelector("button").addEventListener
("click", function(e) {
        console.log(e.target.innerText = 'clicked!');
  },false);
```

This will skip div2 once, from element to root.

Event Delegation - attach / append a SINGLE event listener to a parent element that adds it to all of it’s present AND future descendants that match a selector. Improve performance and write less code.


```
document.querySelector('#golf').addEventListener
('click', function(e) {
    console.log('golf is clicked');

    const target = e.target;

    if(target.matches('li')) {
        target.style.backgroundColor = 'lightgrey'
    }
})
```


```
document.querySelector('#golf').addEventListener
('click', function(e) {
    console.log('golf is clicked');

    const target = e.target;

    if(target.matches('li')) {
        target.style.backgroundColor = 'lightgrey'
    }
})
```



```
const sports = document.querySelector('#sports');
const newSport = document.createElement('li');


newSport.innerText = 'rugby';
newSport.setAttribute('id', 'rugby');

sports.appendChild(newSport)
```

Browser - 3 ways to store data in user’s browser, single user and specific browser

Size, duration, acessibility

Local storage - 10MB,all tabs, don’t expire unless delete(persistent), store in browser

Session storage - 5MB singlle tab, close tab - expires, store in browser

Cookies - small storage 4KB , all tabs, set expiration, serves as a medium for communication between client and server. client send data to server

Accessibility - the practice of making information, products , and environments usable by as many people as possible, including those with disabilities.

In web development means enabling as many people as possible to use websites.

Example: Color accessibility, Audio, hearing

Screen reader, Virtual hierarchy

When information and communication technologies are inaccessible, people with disabilities are denied equitable access to education, employment, and involvement in society.

it also benefits businesses by reaching a wider audience and minimizing legal risks associated with inaccessible design.