→ HTML API (Geo Location — Web Worker
— Drag & Drop — SSE
Web Storage

Tuesday

1) Geolocation API

* Used → locate → user's posistion — (C) Not available
  - user approved
  - Since ↓ priv → compromise

→ get geographical posistion of user.

→ getCurrentPosistion() ∿) Return user's posistion
  - (↓) Lat longitude Hude
  - 2 Parameter
  - Success call — One handle errors — para meter (in) specified → to function
  - If → return → coordinate object → to function

Eg:

```
<1.DOCTYPE html>
<html>
  <body>
    <button id="btn" onclick="getLocc()">Pos</button>
    <p id="p"></p>
    <script>
      var p = document.getElementById("p")
      var btn = document.getElementById("btn")
      function getLocc(){
        if (navigator.geolocation){
          navigator.geolocation.getCurrentPosistion(show
          Loc,
          showErr)
        }
        else {
          p="NA";
        }
      function showLoc (pos){
        alert("Latitude:" +pos.coords.da+itude + "In longi")
                         + pos.coords.longitude )
      }
      function showErr (error){
        switch (error.code){
          case error.PERMISSION - DENIED:
            p.innerHTML = "user denied"
            break;
```

```
case error.POSISTION -UNAVAILABLE'.
p.innerHTML e "Location NA"
break';
case error.TIMEOUT';
p.innerHTML : "timeout"
break';
case error.UNKNOWN-ERROR';
p.innerHTML ="Unknown"
break';
}
}
</script>
<body/>
</html>
```

getCurrentPosistion()

return → object on success
if available
Latitude
Longitude   others
Accuracy

(like gps har)
user moves

↑
at
↑
↑ updated pos.

> Watch Posistion() '! Return → Current → user
Pos
↳ Continues ↱ return

> clearWatch() " → stop

> Property of Coords. ─ (Latitude   (Accuracy   (altitude
(longitude   (Attitude   Accuracy
(speed)   ← time   (heading
stamp

2) Drag & Drop API → Common feature (When → object
→ getting user experience   grab → drag
(default   diff
location

1) Set Draggable Attribute ~ True ~ element
draggable

2) On Drag Start → Calls function, drag (event) that specifies
what data to be dragged.

3) On Drag Over ~ 'Specify where → dragged data → dropped

4) On Drop ~ 'When ~ Dragged Data is → Event occurs

FEBRUARY   2020
Su Mo Tu We Th Fr Sa
                    1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

5) If draggable → set → false → can't drag image.
6) data Transfer 'setData() ~ set Data Type
value → dragged
data
7) ev → event

8) On Drag Start → function drag(ev) {
    ev. dataTransfer . setData("text",
    ev.target.id);
}
(key)

9) on drag Over ∩ Default → | Data/Element (cannot dropped → in other elements |

To Allow ← we prevent → default → element handling

done by → | event . preventDefault() |

10) On Drag Start → on Element to be dragged
    use it

On Drag over → Use it on to which to be dropped

11) data Transfer .get Data() ∩ get Dragged Data
    Return ∩ Any Data → was → set that
    Set Data ← the Same ← to type
    ()

12) ev . target (id) → Target Event ∩ In this is → Image
    data → key (Step 8)
    refers

Eg or

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        border: 1px solid black;
        padding: 5%;
        width: 200px;
        height: 200px;
        display: block;
        position: relative;
      }
    </style>
  </head>
  <body>
```

```html
<img id="draga" src="ch.jpg" width="450px"
     height="150px" draggable="true"
     ondragstart="drag(event)">

<div id="up" ondragover="allowDrop(event)" ondrop="drop(event)">
</div>

<script>
    var p = document.getElementById("p")

    var bt

    function drag(ev) {
        //alert(ev)      // [object DragEvent]
        ev.dataTransfer.setData("text", ev.target.id);
    }
    function allowDrop(ev) {
        ev.preventDefault();
    }
    function drop(ev) {
        ev.preventDefault();
        var data = ev.dataTransfer.getData("text");
        alert(data);   //draga
        ev.target.dataTransfer.appendChild(document.
                                getElementById(data));
    }
</script>
</body>
</html>
```
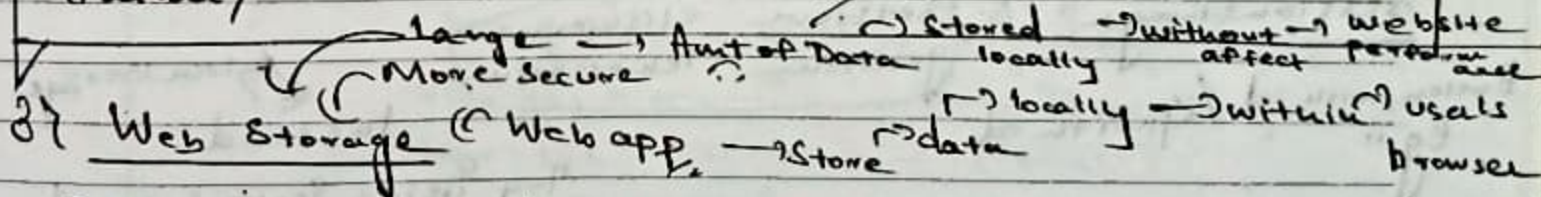
large → Amt of Data → Stored → without → website
More Secure ?   locally   affect perform
                                                          ance
3) Web Storage (Web app → Store → locally → within → users
                               → Store   data   browser

                          include
Before u In → every → Server → data → Store
                        request   Cookies

Storage Limit → At least → 5MB (for larger)

Info. → Never → Transferred to server.
                                                              → data
It → per origin [per domain]
                         protocol   → same
All pages → from → one → Store /server
                        origin

Web Storage → 2 objects → Storing Data   Client Gone

	Store Data
Window . local Storage	No expiration Date
Window . Session Storage	for 1 Session (Data → when → Browser Lost → Tab Closed)

¬) Browser Support

¬)
```
if (typeof (Storage) !== "undefined") {
    // Code
    } else {
    // Sry NA
    }
```

¬) localStorage
```
¬) localStorage . setItem (name, value)
¬) local Storage . getItem ()
    @1
   localStorage . getItem (name);
```

localStorage . clickCount

```
localStorage . removeItem (name) ✓
    or
localStorage . removeItem ()
```

¬) SessionStorage   S. setItem   clickcount
                        getItem   removeItem   localStorage
Button with id / <p with id   instead
Eg "   Try using SessionStorage
                                            ios

```
<script>
function clickCounter {
    if (typeof (Storage) !== "undefined") {
        if (localStorage . clickcount) {
            localStorage . clickcount = Number
                (local . clickcount) +1;
        } else {
            localStorage . clickcount = 1;
        }
        document . get-ed ("p") . innerHTML = local . clickcount;
                                                        storage
    } else {
        ...
<script>
```

{18 4} Web worker

Page ← performance
← affect

→ It is a JS running in background without

→ Eg: Click
Button → If that → Takes → No of in completing → task
Button seconds

So until → particular
task → completed → not
able → other in → HTML
to work
do ↓
any get
stuck

Browser
Support →

```
if (typeOf (Worker) ! == "undefined"){
    //code
} else {
    //sorry
}
```

→ Eg: Below Program Shows Working of without web worker

```
<! DOCTYPE html>
<html>
    <body>
        <button id="btn1" onclick="looper">Big Loop </button>
        <button id="btn2" onclick="SayHi">Hello </button>
        <p id="p"> </p>
        <script>
            var i, j;
            var p = doc...getElementById("p");
            var bt1 = _____("btn1");
            var bt2 = _____("btn2");
            function loop(){
                for( i=0; i < 590000000; i+=1){
                    j=i;
                }
                alert("Completed :" +j);
            }
            function SayHi{
                alert("Hello");
            }
        </script>
    </body>
</html>
```

completed
this
↓
after
↓
display

o/p

[Big Loop] [Hello]
→ OK

Click
Takes Time

Click
this

FEBRUARY   2020
Su Mo Tu We Th Fr Sa
                      1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

18

FEBRUARY 2020

document → Window ← → objects
parent → ↓ No Access
→ External files

With Web worker

- **Post Message()** → used → post → back to → HTML
  message page

- Not ← used → Simple ; More → CPU Intensive
  Scripts Tasks

- terminate() → Terminate
  ↓ → and → free browser
  WebWorker computer
  resources -

- onmessage → When Message → received → through
  ↓
  Event
  Source

- Eg:
  Canvas.html | bloop.js

```
<script>
Var p = doc... id("p");

var b+1 = doc.. id("b+n1");
Var b12 = doc.- id("b+n2");

function looper {
  if (typeof (Worker)! ==
                "undefined")
                  {
  Var worker = new Worker
            ("bloop.js");

worker.onmessage = function (event)
            {
  alert("Completed: "+event.
                  data);
  };
  }else{
  p.innerHTML = "Na";
  }}
function sayHi() {
  alert("Hello");
}
</script>
```

```
Var i, j;

for(i=0; i<90000000;
            {i+=1} {

j=1;
}
postMessage(j);


— x —


Rw
```

FEBRUARY 2020 → Server Sent Event API
→ allow → web page → get updates → from server
automatically

19

- Web page → had → ask → any → updates ; With SSE → comes
                                  available

- eg:" Sports Result Stock
      News Feed Price updates

- EventSource object → used → receive → SSE Notifications

  ┌───┐
  │ Var Source = new EventSource ("a...") ; │
  └───┘

- Each time → update received → onMessage → occurs.

  id ← element ← Into received ← puts
    with data

- Browser support ──────→ if (typeof (EventSource)!= "undef")
                          { code }
                          else { // NA }

- Eg:" PHP file → XAMPP server HTML PHP (comes → with)
      → Go to Drive → Xampp → htdocs → Create
              (folder) (folder) folder
                                    (SSEapi)

- Set () content-Type → header → text/event-stream
- Specify → page → not cache
- o/p → data → to send → Start with (data:)
- flush → o/p → Data → Back → web page

  demo.html | demo.php

FEBRUARY 2020
Su Mo Tu We Th Fr Sa
                1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

```
{!DOCTYPE html}
{html}
{body}
  <div id="result"></div>
{script}
if (typeof (EventSource)! = "")
Var Source = new EventSource ("demo.php");
Source.onmessage = function (event){
res.innerHTML += event.data+"<br>"
{/script} {/body}
```

```
<?php
header ("Content-Type: text/event-stream");
header ('Cache-control');
$time = date ("r");
echo "data: $time\n\n";
flush();
?>
```