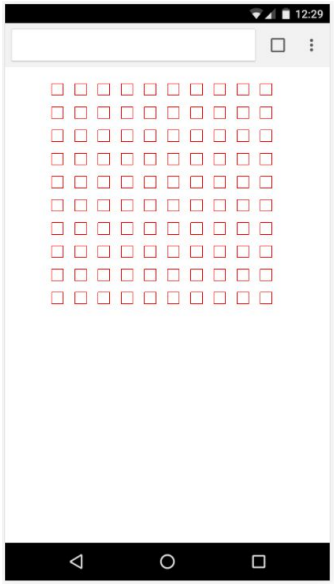
TD : (Partie 2)
Listeners

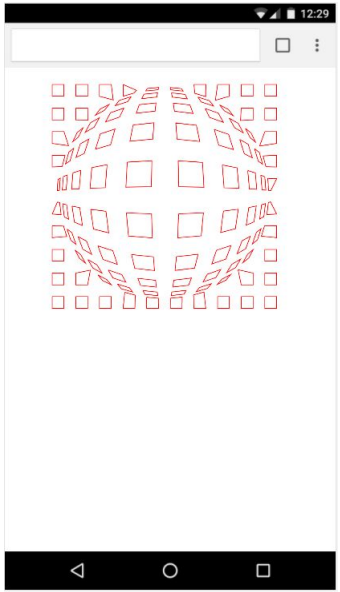
1

- Point & Polygon
- generatePolygons()



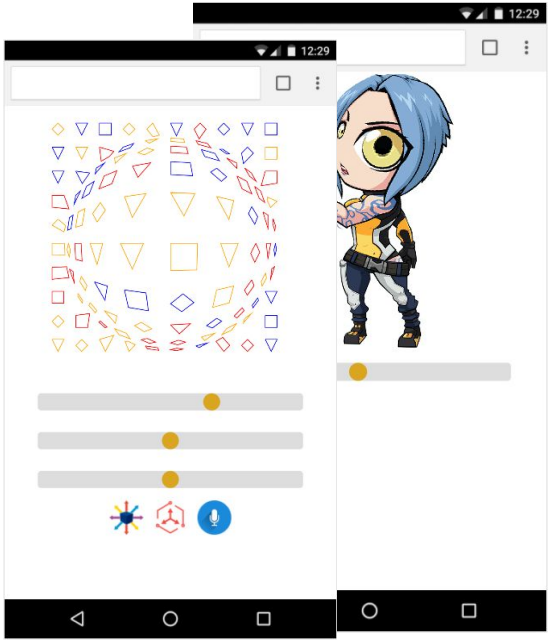
2

Fisheye deformation



3

Listeners




AH, THERE'S THE SIGNAL —
NOW WE CAN TAKE OFF.


AH, THERE'S THE SIGNAL — NOW WE CAN TAKE OFF.
Listener
Target
Event

Syntax

```
target.addEventListener(type, listener[, options]);
target.addEventListener(type, listener[, useCapture]);
```

Parameters

`type`

A string representing the event type to listen for.

`listener`

The object which receives a notification (an object that implements the `Event` interface) when an event of the specified type occurs. This must be an object implementing the `EventListener` interface, or a JavaScript function.

https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener

Exemples

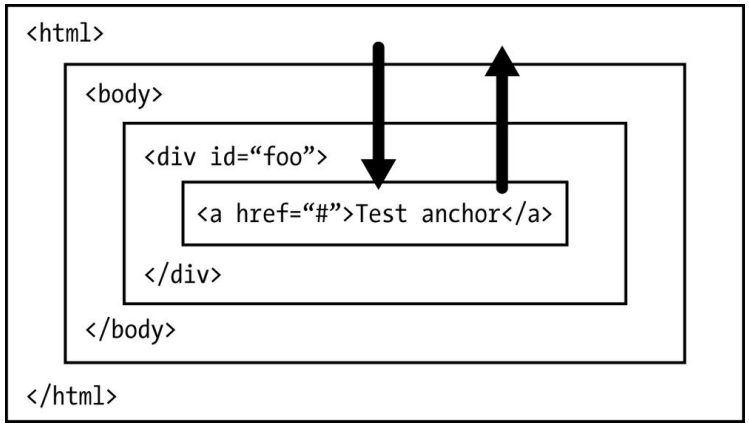
```
<script>
  document.addEventListener("DOMContentLoaded", function(event) {
    console.log("DOM fully loaded and parsed");
  });
</script>
```

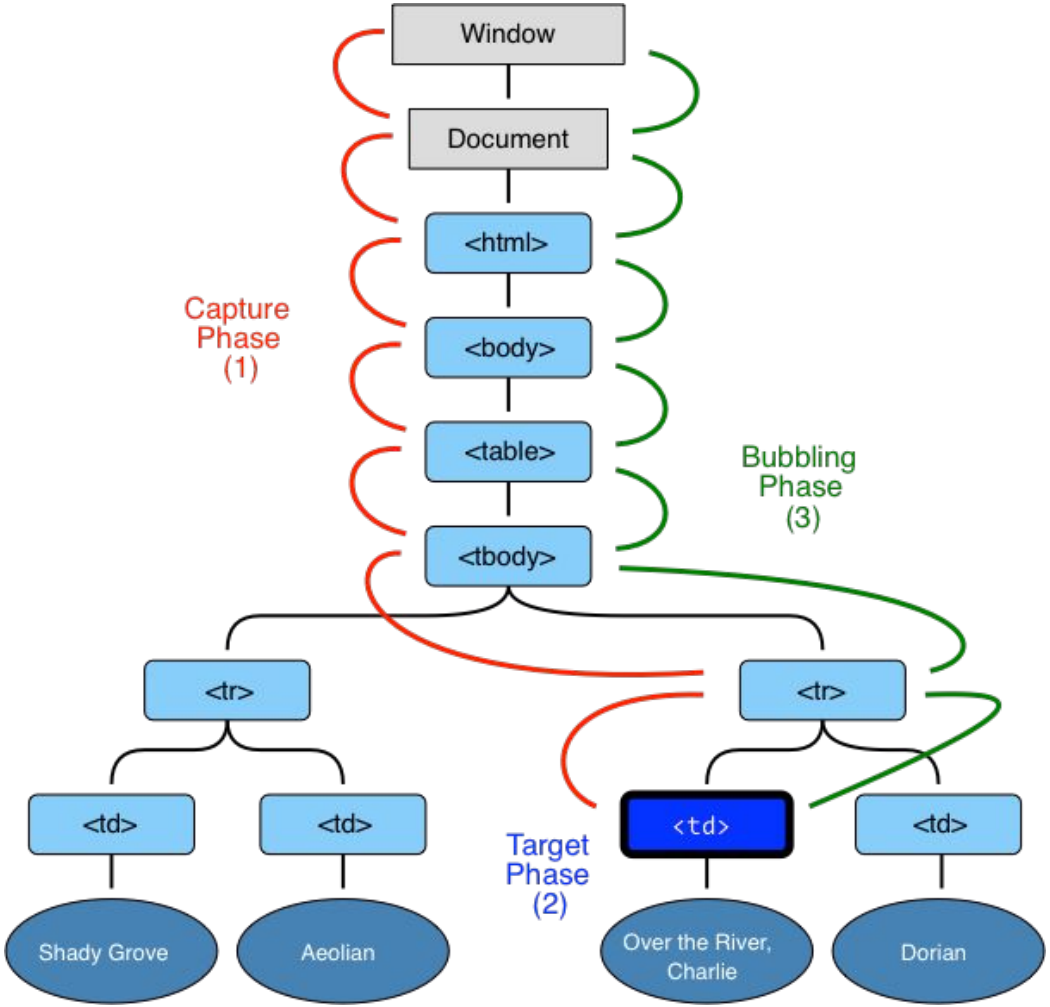
Exemples

```
<script>
  document.addEventListener("DOMContentLoaded", function(event) {
    console.log("DOM fully loaded and parsed");
  });
</script>
```

```
document.getElementById("myBtn").addEventListener("click", function(){
    this.style.backgroundColor = "red";
});
```

Et si nous avons plusieurs listeners, lequel se déclenche en premier ?



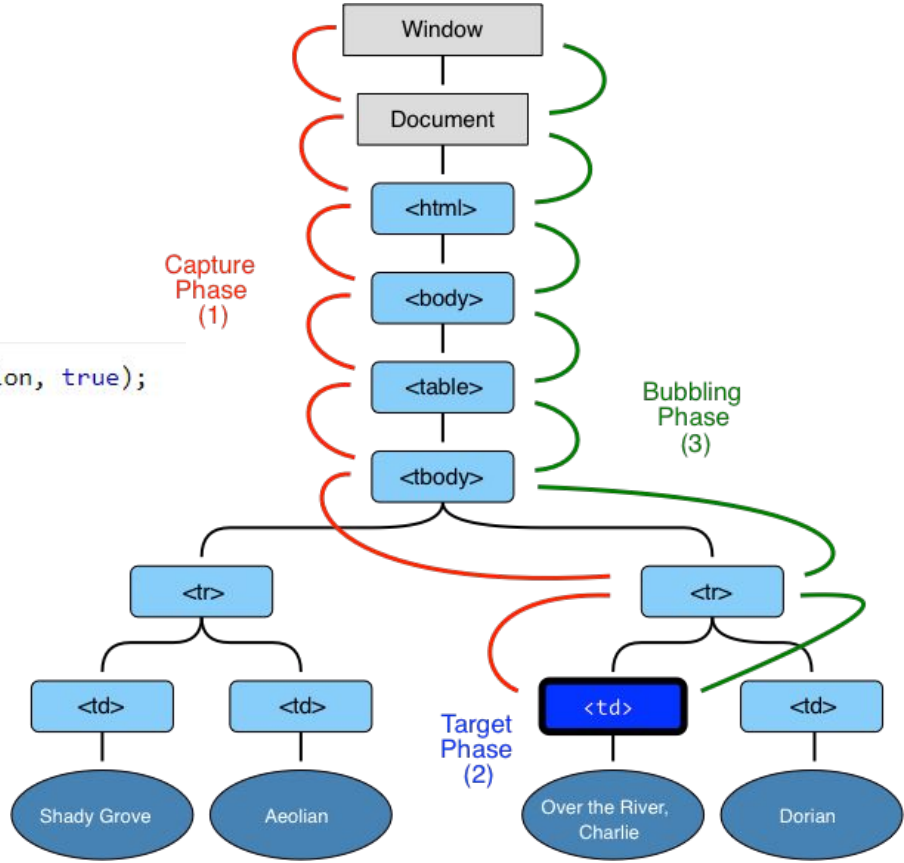

Window
Document
<html>
<body>
<table>
<tbody>
Capture Phase (1)
Bubbling Phase (3)
<tr>
<tr>
<td>
<td>
<td>
<td>
Target Phase (2)
Shady Grove
Aeolian
Over the River, Charlie
Dorian

UseCapture (3e paramètre) **:**

```
document.getElementById("myDiv").addEventListener("click", myFunction, true);
```

→ Déclenche le listener durant la *“capture phase”*

→ Par défaut est à false



Attention, les listeners peuvent provoquer des problèmes UI

Scroll performance

Best Practices

We've compiled some recommendations for modernizing your web app and avoiding performance pitfalls. These audits do not affect your score but are worth a look.

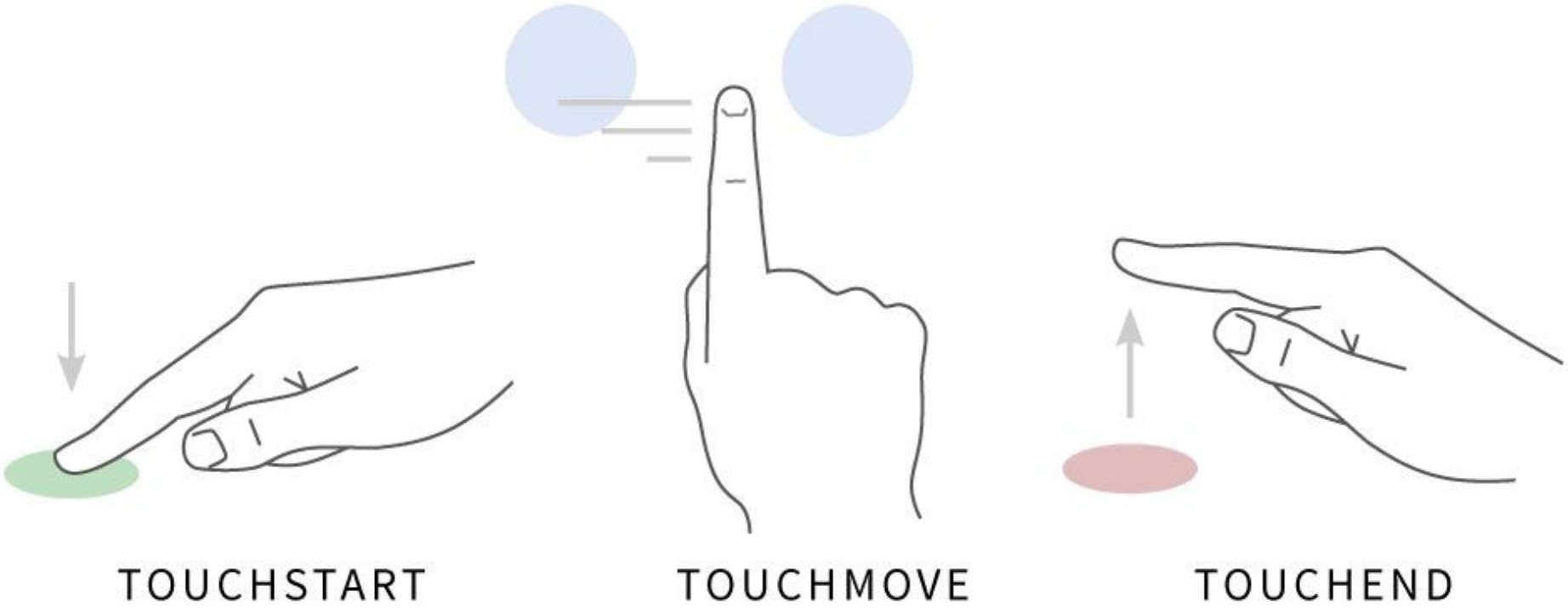
✖ Uses passive listeners to improve scrolling performance

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. Learn more.

▶ View Details

```
document.addEventListener('touchstart', handler, {passive: true});
```

Touch event

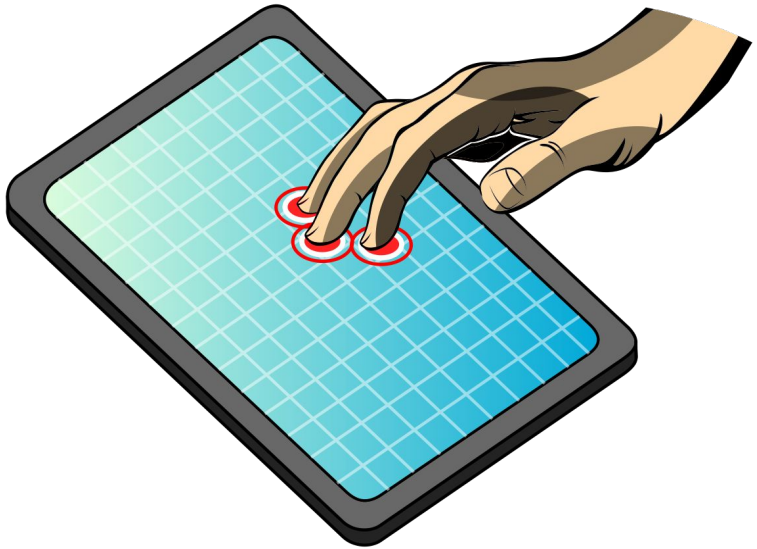


Touch Listener

```
const mobileTouchListener = function(e) {
    e.preventDefault();
    const i = e.touches.length - 1;
    const relativeX = e.touches[i].clientX + document.body.scrollLeft - this.offsetLeft;
    const relativeY = e.touches[i].clientY + document.body.scrollTop - this.offsetTop;
    ...
}

const desktopClickListener = function(e) {
    const relativeX = e.clientX + document.body.scrollLeft - this.offsetLeft;
    const relativeY = e.clientY + document.body.scrollTop - this.offsetTop;
    ...
}

canvas.addEventListener("touchstart", mobileTouchListener,{passive: false});
canvas.addEventListener("touchmove", mobileTouchListener);
canvas.addEventListener("mousemove", desktopClickListener,{passive: true});
```



Multitouch ?

Double tap ?

Long Press ?

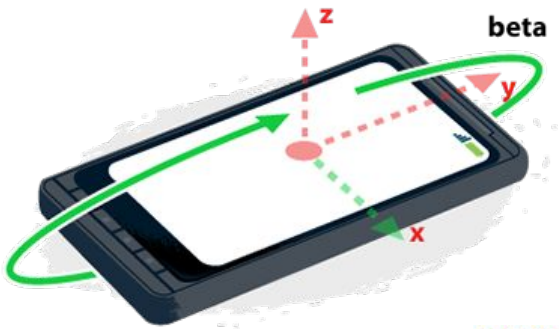
Touch Listener

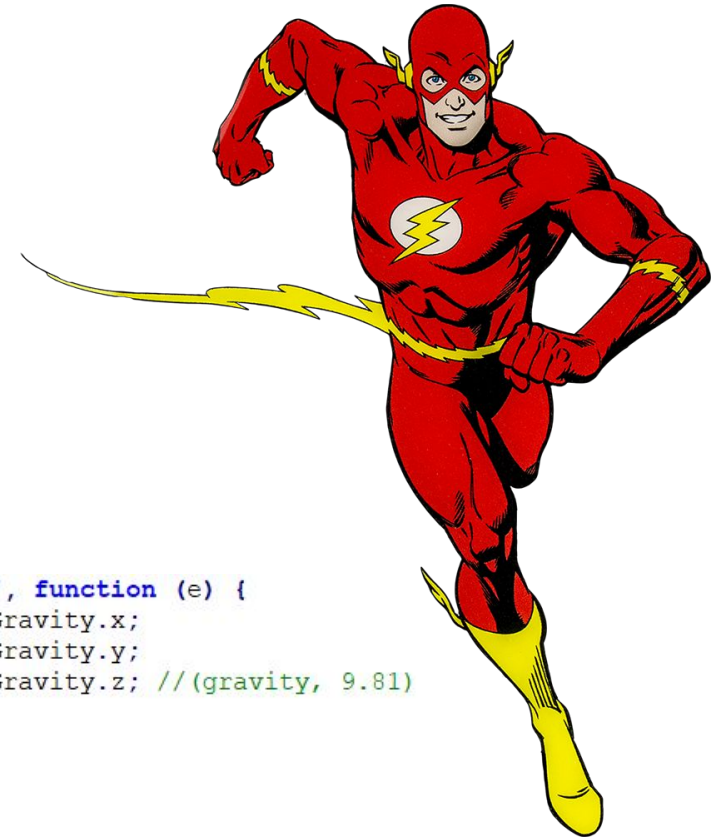
```
/**
 *  @param {Event} e
 */
onTouchStart (e) {
    this.timeoutLongPress = setTimeout(this.longPress, this.delayLongPress);
}
onTouchEnd(e) {
    clearTimeout(this.timeoutLongPress);
}
```

```
/**
 *  @param {Event} e
 */
onTouchEnd(e) {
    const currentTime = new Date().getTime();
    clearTimeout(this.timeout);
    // First tap
    if (this.doubleTapTime === 0) {
        this.doubleTapTime = currentTime;
        // timeout if no double tap after delayBeforeDoubleTap
        this.timeout = setTimeout(() => {
            this.doubleTapTime = 0;
            clearTimeout(this.timeout);
        }, this.delayBeforeDoubleTap);
    }
    // Double tap
    else if (currentTime - this.doubleTapTime < this.delayBeforeDoubleTap) {
        this.doubleTapTime = 0;
        this.doubleTap(e);
        e.preventDefault();
    }
}
```

Device orientation & motion

```
window.addEventListener("deviceorientation", function (e) {
    const alpha = e.alpha; // compass (north)
    const beta = e.beta; // angle x
    const gamma = e.gamma; // angle y
    ...
});
```





```
window.addEventListener("devicemotion", function (e) {
    const x = e.accelerationIncludingGravity.x;
    const y = e.accelerationIncludingGravity.y;
    const z = e.accelerationIncludingGravity.z; //(gravity, 9.81)
    ...
});
```

Web Speech API

```
/**
 *  VoiceAPI (work only on https:// or localhost (not IP adress)
 *  @class
 */
class VoiceAPI {
    constructor(callback) {
        this.callback = callback || function(){};
        this.recognition = new webkitSpeechRecognition();
        this.recognition.interimResults = false; // Live detection
        this.recognition.onresult = this.onResult.bind(this);
        this.recognition.onerror = this.onError;
    }
    start () {
        this.recognition.lang = "fr-FR";
        this.recognition.start();
    }
    /**
     *  @param {Event} e
     */
    onResult (e) {
        this.callback(e.results[0][0].transcript);
    }
    /**
     *  @param {Event} e
     */
    onError (e) {
        alert(e.error);
    }
}

const voice = new VoiceAPI((stringResult) => {
            ...
});
voice.start();
```

```html
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Hello World</title>
        <meta content='user-scalable=0' name='viewport'/>
        <style>
            * {
                padding: 0;
                margin: 0;
            }
        </style>
    </head>
    <body>
        <canvas id="myCanvas" width="500" height="500"></canvas>
        <script>"use strict";
            /**
             *  Entry point
             */
            window.addEventListener("DOMContentLoaded", function() {
                const canvas = document.getElementById("myCanvas");
                const ctx = canvas.getContext("2d");
                ...
            });

            /**
             *  Javascript Model
             */
            class Polygon {
                ...
            }
        </script>
    </body>
</html>
```

→ Contient les “**addEventListener**”

→ Les **listeners** (fonctions) peuvent être définis ailleurs

→ Le ***“javascript model”*** peut être maintenant défini dans un JS à part