Life is Possible - 生命教育 手機程式工作坊

Lecture 05 - Interact JS in HTML

Menu

- Using JS in html
- DOM control
- Making a mini quiz game

Recall

What we have learn in previous lessons?

```
let a = 10;

if(a === 10){
  console.log("Hello mate")
}

function calculateAgeStatus(age){
    console.log(`You are in ${age} right?`)
}

calculateAgeStatus(16)
```

Recall Summary

- Run a `app.js`
- Print variables with `console.log`
- Data Types `string`, `number`, `boolean`
- Arithmetic Operators `++`, `*`, `/` ...
- Assignment Operators `=`, `+=`, `*=`, `/=` ...
- Define variables `let`, `const`, `var`
- Conditional Statements `if`, `else`
- Compare Statements `==`, `===`, `<=` ...
- Functions `function add(a,b){ return a + b }`

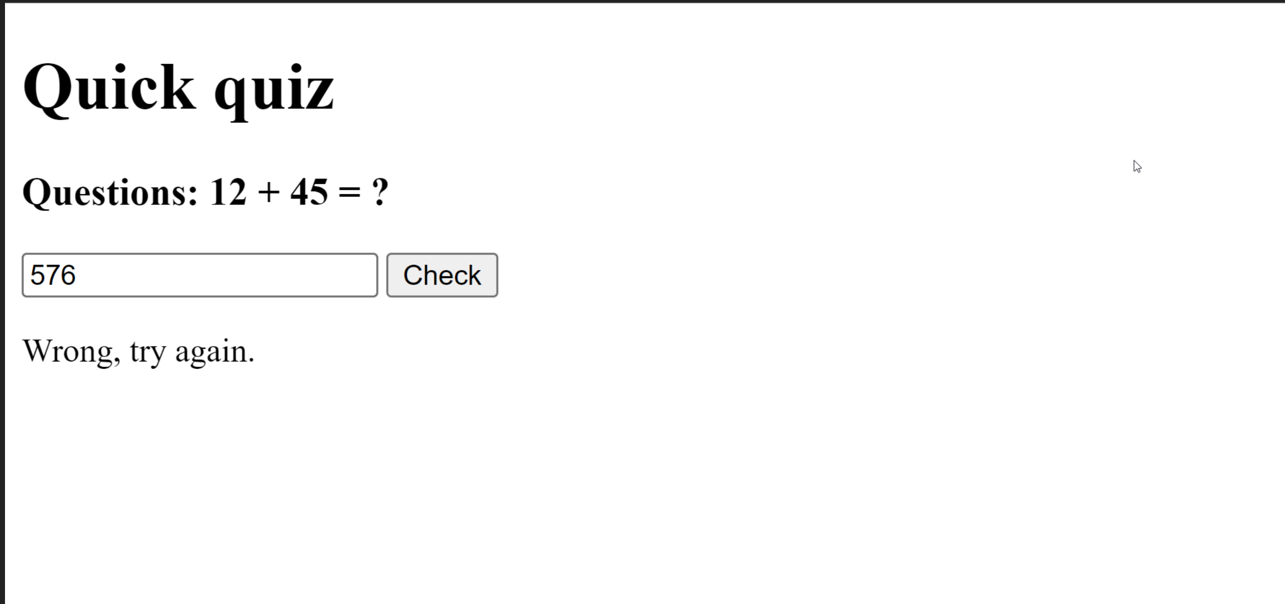
But

What if javascript in HTML?

Quick test

We have a html file for a quick quiz

How can we know the user input the correct ans?



Quick quiz

Questions: 12 + 45 = ?

Wrong, try again.

Using HTML ?

Opps, HTML is only for elements

Using CSS ?

Nope. CSS is only for style

Using Javascript ?

Correct, but why?

We need Javascript in HTML

`Javascript` is the brain in HTML
It helps to do all the calculate / logics rendering job

Example of: Is something clicked? Calculate the sum of user inputs.

Add `script` in html

In html, we would add a `<script>` tag for using javascript.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>
    <h1> Yo all </h1>
    <script> <!-- Add me for using script -->
        console.log("Hello all")
    </script>
</body>
</html>
```

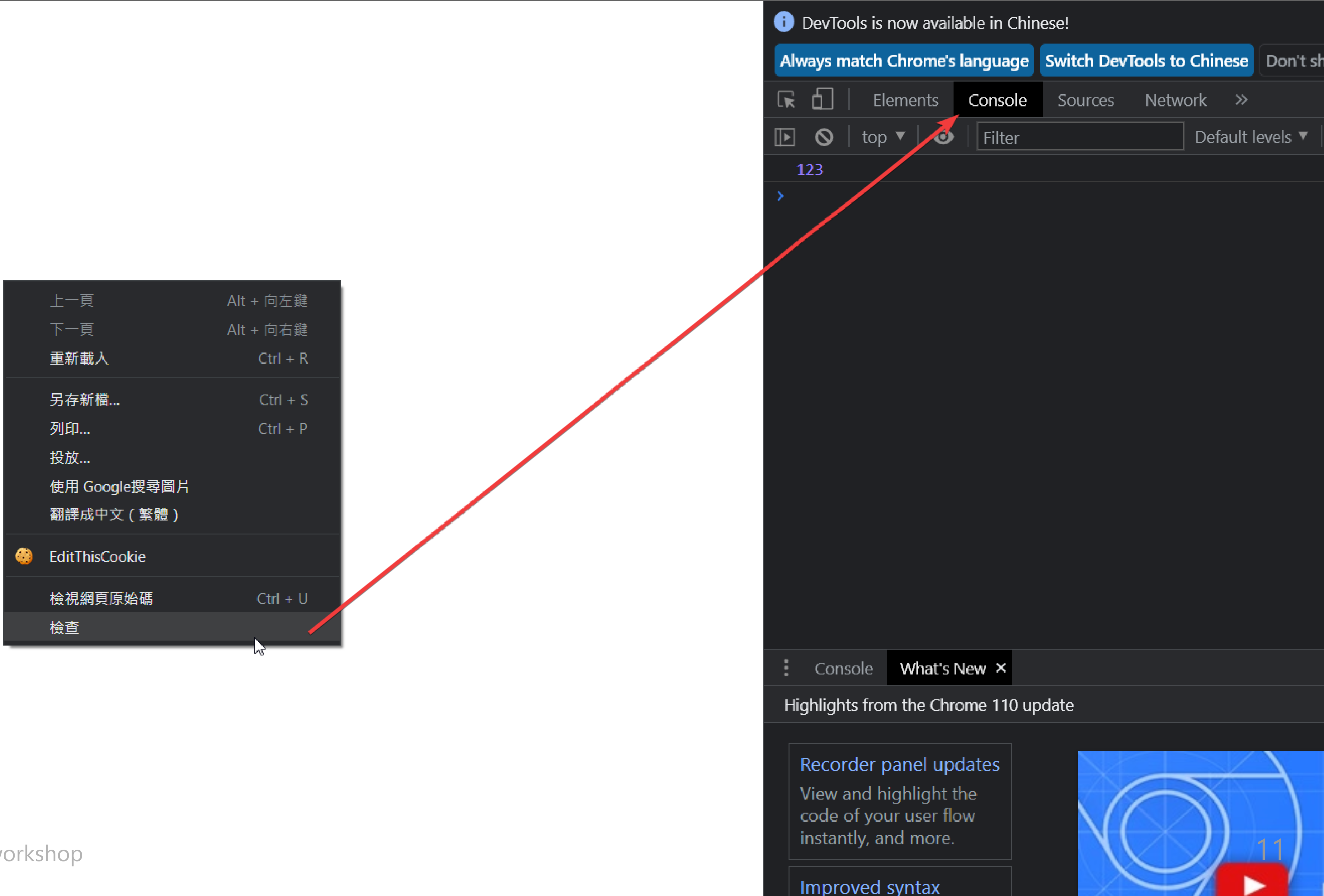
Writing script in html

```
<script>
      let a = 100;
      let b = 23;

      let c = a + b;

      console.log(c); // where is it logged to?
</script>
```

`Right click` in your html OR press `F12` to open the `DevTools`
Then find the `Console` in the right top sections.



DevTools is now available in Chinese!

```
123
>
```

Highlights from the Chrome 110 update

Recorder panel updates

View and highlight the code of your user flow instantly, and more.

Improved syntax

Control the Dom with `document.querySelector()`

Basic Dom control in web js

Let’s assume we have the HTML like this

`index.html`

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3 id="msg">Hello mate</h3>
    <h3 class="yolo">yolo hi</h3>
</body>
</html>
```

Get the DOM elements

In general, we have these function to get the regarding elements. Both function are build in for web.

```javascript
// getElementBy
const msgBox = document.getElementById('msg'); // Is a id
const yoloBox = document.getElementsByClassName("yolo"); // Is a array
```

```javascript
// querySelector
const msgBoxQu = document.querySelector('#msg'); // Is a id
const yoloBoxQu = document.querySelector(".yolo"); // Not a array

// Since msg is a id, so we need to add #
// Since yolo is a class, so we need to add .
```

Those function can help us to get the regarding elements.

Add interactions to boxes

In general, we can add event to the DOM elements:

```
const msgBox = document.getElementById('msg');

// When the msgBox box clicked, a message box will be pop up
msgBox.addEventListener("click", function() {
    alert("Hello mate")
});
```

index.html with const

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3 id="msg">Hello mate</h3>

    <script>
        const msgBox = document.getElementById('msg');

        // When the msgBox box clicked, a message box will be pop up
        msgBox.addEventListener("click", function() {
            alert("Hello mate")
        });
    </script>
</body>
</html>
```

index.html with document

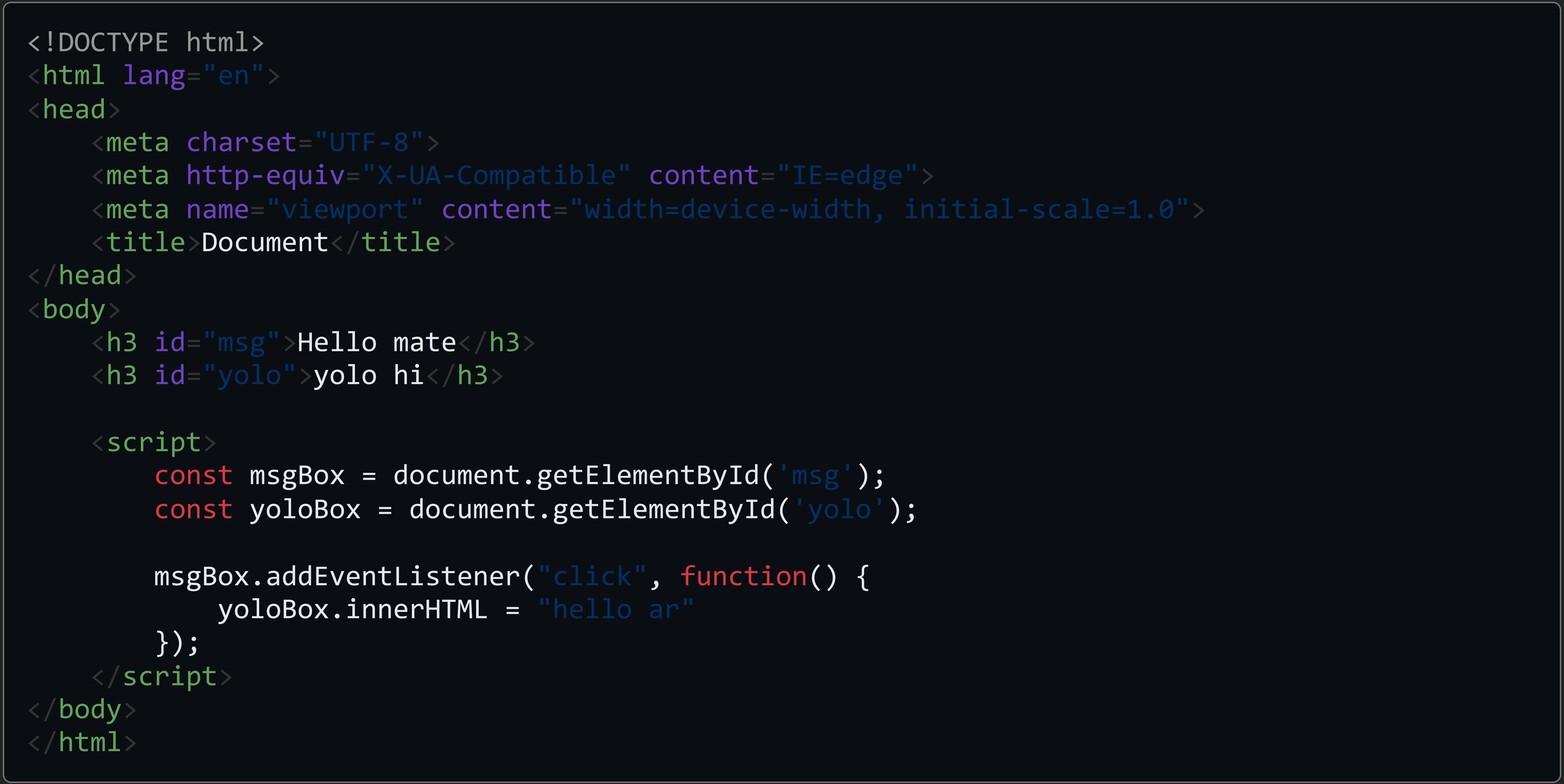
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3 id="msg">Hello mate</h3>

    <script>
        // When the msgBox box clicked, a message box will be pop up
        document.getElementById('msg').addEventListener("click", function() {
            alert("Hello mate")
        });
    </script>
</body>
</html>
```

Dom value control

You can control the messgae of the elements by `innerHTML` too.

index.html with const



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3 id="msg">Hello mate</h3>
    <h3 id="yolo">yolo hi</h3>

    <script>
        const msgBox = document.getElementById('msg');
        const yoloBox = document.getElementById('yolo');

        msgBox.addEventListener("click", function() {
            yoloBox.innerHTML = "hello ar"
        });
    </script>
</body>
</html>
```

index.html with document

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3 id="msg">Hello mate</h3>
    <h3 id="yolo">yolo hi</h3>

    <script>
        document.getElementById('msg').addEventListener("click", function() {
            document.getElementById('yolo').innerHTML = "hello ar"
        });
    </script>
</body>
</html>
```

More references on `innerHTML`

counter.html

```html
<body>
    <h3 id="counter">0</h3>
    <button id="addCount">add num</button>

    <script>
        const counterNumber = document.getElementById('counter');
        const addCount = document.getElementById('addCount');

        addCount.addEventListener("click", function() {
            let originalNUmber = parseInt(counterNumber.innerHTML)
            originalNUmber ++;

            counterNumber.innerHTML = originalNUmber
        });
    </script>
</body>
```

Break

Control with input

In HTML, we have a tag `<input>` which allow user to input stuff

> The HTML `<input>` element is used to create interactive controls for web-based forms in order to accept data from the user.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

Control with `<input>`

The input tag is used to get the user enter data and send back to server / local checking.

For examples, we use the `<input>` tags like this

```html
<input id="textInput" type="text">
<input id="numberInput" type="number">
<input
    type="file" id="fileInput"
    id="avatar" name="avatar"
    accept="image/png, image/jpeg"
>
```

Control with `<input type="???">`

Following type are supported

- `text`, `number`, `date`
- `password`, `email`
- `checkbox`, `radio`
- `color`, `url`
- `week`, `month`, `time`, `datetime-local`, `date`
- `tel`, `range`
- `submit`, `reset`

And more ...

Get `<input>` value

To get the input value in javascript, you need to assign a id / calss to an input tag first.

`index.html`

```html
<input id="textInput" type="text" value="hello">
```

Then, in javascript, we have to do it with `.value`

`index.js`

```javascript
const textInput = document.getElementById("textInput");
console.log(textInput.value);
```

Get `<input>` value when typing

To detect the user input event, we have to use `addEventListener` with `input` to listen the type event.

`index.html`

```html
<input id="textInput" type="text">
```

`index.js`

```javascript
const textInput = document.getElementById("textInput");

textInput.addEventListener("input", () => {
    console.log(textInput.value)
})
```

Get multi `<input>` value when typing

To detect different input, assign DIFFERENT id to each `<input >` elements for it.

```html
<input id="textInput" type="text">
<input id="englishInput" type="text">
```

`index.js`

```javascript
const textInput = document.getElementById("textInput");
const englishInput = document.getElementById("englishInput");

textInput.addEventListener("input", () => {
    console.log(textInput.value)
})

englishInput.addEventListener("input", () => {
    console.log(englishInput.value)
})
```

break

Lab 1: Math quiz

Write a math quiz site that each time will generate a different math question and answer. And each time will regarding the user input to check if the answer match the sum.

Examples:

- This time is 10 + 23 = ?
- Other time is 10 + 36 = ?
- Another time is 10 + 2 = ?

You can references to `quiz.html` for this lab

Tips 1: Using `Math.floor( Math.random() * 30)` to generate a random integer number.

Lab 1: Math quiz Tips

`tips1.js`

```
let rngNum = Math.floor( Math.random() * 30) + 1;
```

`tips2.html`

```
<h3>Questions: 10 + <p id="rngNumberBox"></p> = ?</h3>
```

Lab 1.1: Bonus Math quiz

If you are doing quick in lab 1, try to adjust the questions to this:

Questions: 10 + x = 45, find x
Questions: 14 + x = 32, find x

Generate a site that require user to input `x`, and check if the `x` can fullfill the equations.

The answer should random too.

End