LEARN WEB

ale66

github.com/ale66/learn-web

LAUNCHING JS

REVIEW

- JS is the programming language that runs ‘inside’ web pages
- it can access and change all details of the page at *rendering time*
- access is through the DOM notation:
  `document.GetElamentById('demo').innerHtml`

Let’s review how JS changes the page, one element at a time

CHANGE TEXT

```
document.getElementById("demo").innerHTML = 'My 2nd JS experiment';
```

Element `demo` could be an `h1` or a `p` or a `div` etc.

CHANGE STYLE

```
document.getElementById('myPara').style.fontSize = "30px";
```

Now text is bigger

REDEFINE ELEMENTS OF THE PAGE WITH JS

```js
const my_restyled_para = document.getElementById('myPara');

my_restyled_para.textContent = "My third JS experimen!"

my_restyled_para.style.fontSize = "30px";
```

use the VS Code command-completion facility to know what parameters are available

```js
document.getElementById("demo").textContent = 'My 2nd JS experiment';
```

only changes text, no HTML

```js
document.getElementById("demo").innerHTML = 'My 2nd JS <it>experiment</it>';
```

Can insert HTML and force a re-rendering of the page

CHANGE ANY PROPERTY OF THE NAMED TAG

```html
<img id="myImage" src="./imgs/winter.jpg" width="200" alt="Start pic">
```

As season change, we change the images.

```javascript
document.getElementById('myImage').src = "./imgs/summer.jpg"
```

Important: the new file must be reachable from the page, or we will spoil the existing rendering

JS CONTROLS THE PAGE

JS functions can *get* the document they’re in

With JS commands of type

```
document.func()
```

we apply function `func()` to the whole document

The browser will run `func()` when encounters it in the process of rendering the page or of running a JS function call

JS PAGE ACTIVATION

JS code can change a hidden parameter, common to all tags, that controls *visibility*

Hence, we control which parts are shown at each stage of the visit

An extreme case: `document.write()` will wipe out the page and rewrite it from scratch; changes are irreversible

```
window.alert('A message to be seen')
```

a gentle way to get users' attention

useful in *debugging* code:

when the behaviour differs from expected, we can execute step-by-step and check what values are stored into variables

`window.print()` is for printing on paper 😄

here `window`, `window.document` and `document` are interchangeable

For reference: document-level functions at w3schools

STARTING JS

- today, pages are often created *client-side:*
    - the browser gets a simple page + lots of JS code
    - it runs the code to obtain the finished page
- in fact, JS execution can account for *layers* of personalisation (geo, tiee and cookies) that can only be descided at rendering time

- JS code can be put in several places inside the page
- placement may negatively affect readability of the page
- it also changes what is available to JS

TWO WAYS TO START

JS functions execute

- automatically, when the browser loads and renders the page
- in response to user’s input, given through buttons, menus etc.

We mostly study JS through the latter

JS: WHERE TO PUT IT?

a short list of commands, separated by `;` cna be placed almost everywhere

please the w3schools.com/js/js_whereto.asp

Cleanest solution:

- put JS functions inside .js files
- keep those in a local folder called, e.g., `JS` of `Scrips`

A PLATFORM FOR LEARNING

Learning programming languages is much like learning foreign human languages: theory and practice

Python: interpreter, VS Code with lots of extensions, code profiler, problem-solvinng challenges etc.

JS: VS Code, refresh page in the browser, not much else…

JS does not have an input/output behaviour *per se*

w3schools.com/js/js_output.asp

THE BROWSER CONSOLE

Firefox: Ctrl + Shift + k

Chromium: Ctrl + Shift + j

CHALLENGE

can you personalise the title of your page?

I.e., write a JS function which changes the title of the page, e.g., inserting the date when the page has been loaded?

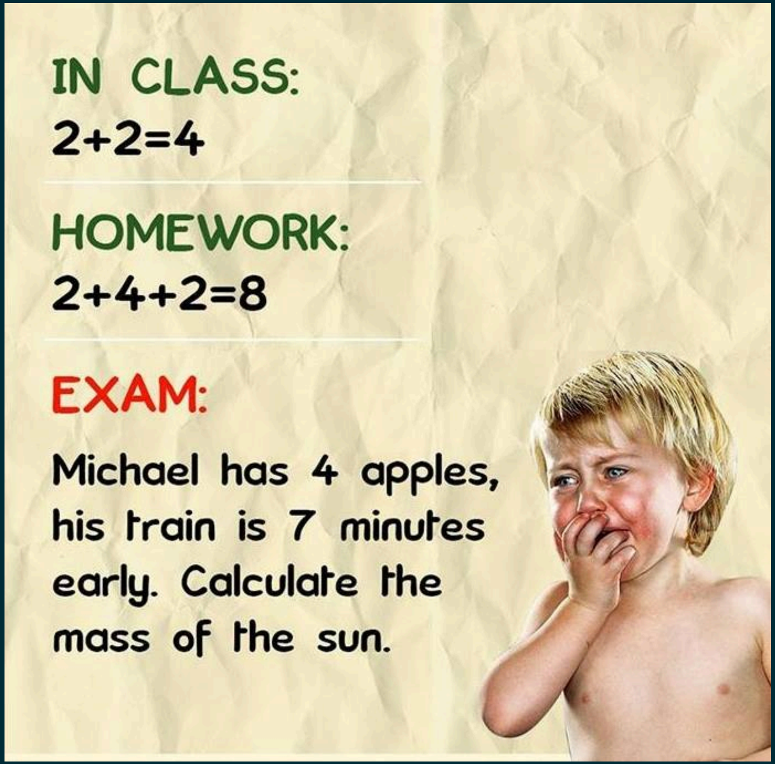
Hint: study the title function

Thanks to the DOM we can write a 1-line function then invoke it in the `<script>` part at the bottom.

EXERCISE:

MA COM is outsourced to the new *University of S. Babila*

Challenge: can you change all the links in a page from the base `www.unimi.it` to `www.unibabila.it`?


IN CLASS:
2+2=4
HOMEWORK:
2+4+2=8
EXAM:
Michael has 4 apples, his train is 7 minutes early. Calculate the mass of the sun.

to get ready for the challenge cover the following units first:

getElementById()

links()

LISTENERS

USER ACTIONS TRIGGER JS REACTIONS

There is a set of advanced HTML tags that are for user interaction:

- bottom
- input fields
- forms
- pull-down menus
- lists (see example in this section)

User interaction on any of these can be connected to invocation of a JS function

LISTENERS

```
<script>
    //'change' is managed by the browser;
    // every time the selected value changes, it invokes the 'react' function
    // notice the lack of parentheses for react: tricky setup!
    document.getElementById('marksSelector').addEventListener('change', react);
</script>
```

`change` is the event that triggers execution of function `react()`, here without `()`

Other notable events are `click`, `mouseover`...

ADVANCED:

getElementsByClassName()

requires handling a **collection** of HTML elements. Proceed if you are familiar with lists or arrays in any programming language.

FURTHER TOPICS

Analise the WaterCSS CSS generator.

Use Markdown to format your instructions.

A NOTE ABOUT LEARNING

Please do not

- watch YouTube videos as they might slow down/comfuse/hinder your learning
- prompt ChatGPT and similar as you (probably) won’t learn anything

Please do use *active learning* web sites such as

- wdpg.io
- w3schools.com/html
- w3schools.com/css
- w3schools.com/js