**Basics**

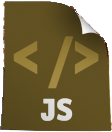
**What’s in a webapp?**



**hypertext markup language** contains text and links to other pages



**cascading style sheets** defines the look and feel



**Javascript** adds interactivity

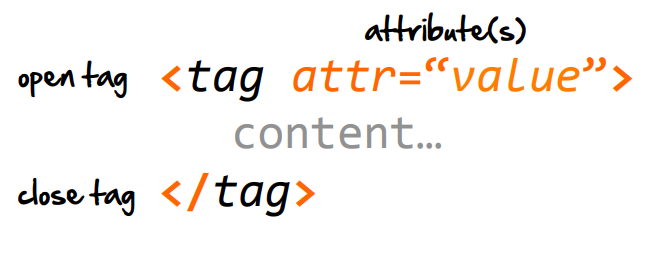
**HTML**

**HTML structure**



**HTML tags**

**Tag structure**



**HTML tags**

**Typography**



text formatting



Organization



Note: view all the tags at: <http://www.w3schools.com/tags/>

**CSS**

* inline styling

<p style=“font-size:14px;”>…</p>

hard to maintain , HTML/CSS coupled

* block styling

<style>

p { font-size: 14px; }

</style>

HTML/CSS coupled

* separate file

p { font-size: 14px; }

Write once for whole site HTML & CSS decoupled

CSS rule syntax

selector {

property: value;

property: value;

…

}

Properties



Selectors

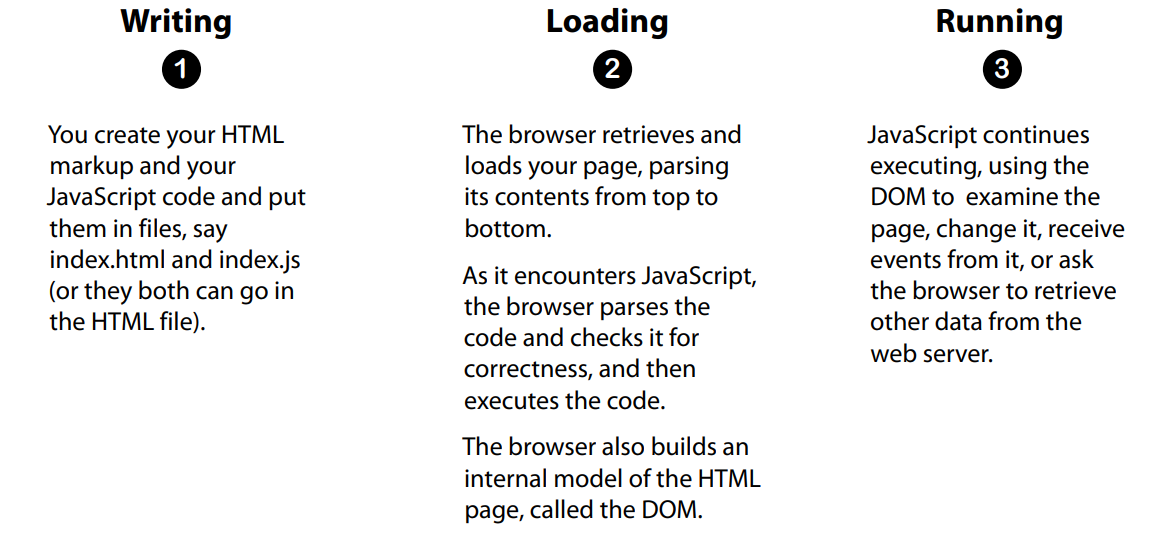
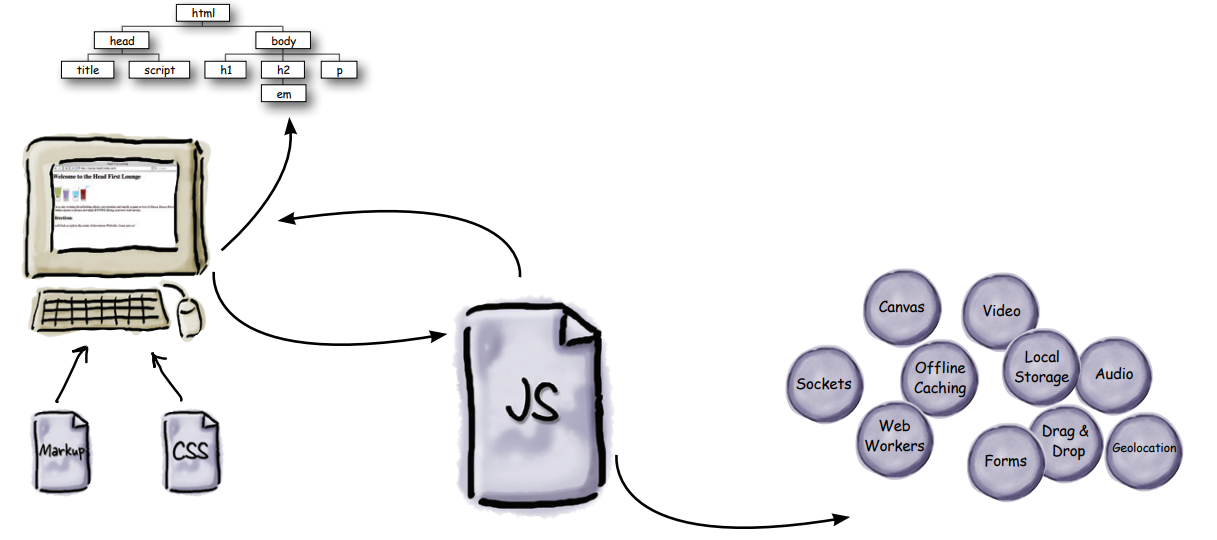


How HTML and CSS look together (how we use them)



**Java Script**

HTML started as mere markup language which can contain text and link to other pages. Later on it added support for java script. If know HTML and CSS you can create some great-looking pages, but they’re still just pages. When you add behavior with JavaScript, you can create an interactive experience; or, even better, you can create full blown web applications

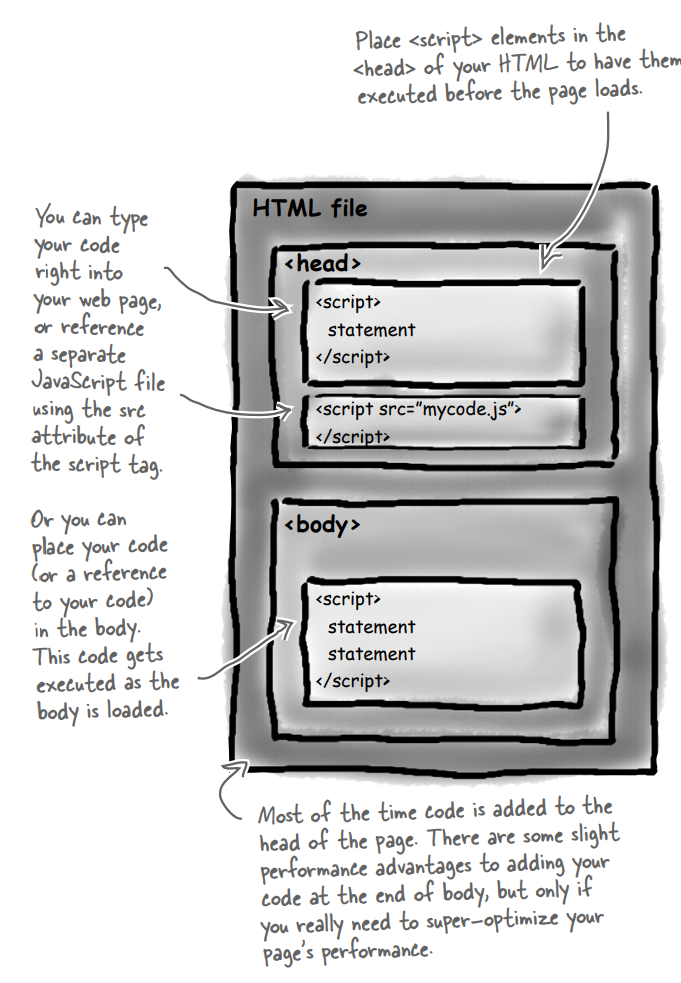


**Variable** is a storage for value, which has name can be retrieved for future usage.

**Method** is collection statement which perform specific task. Method has name and parameter can be passed with that. Called anytime.

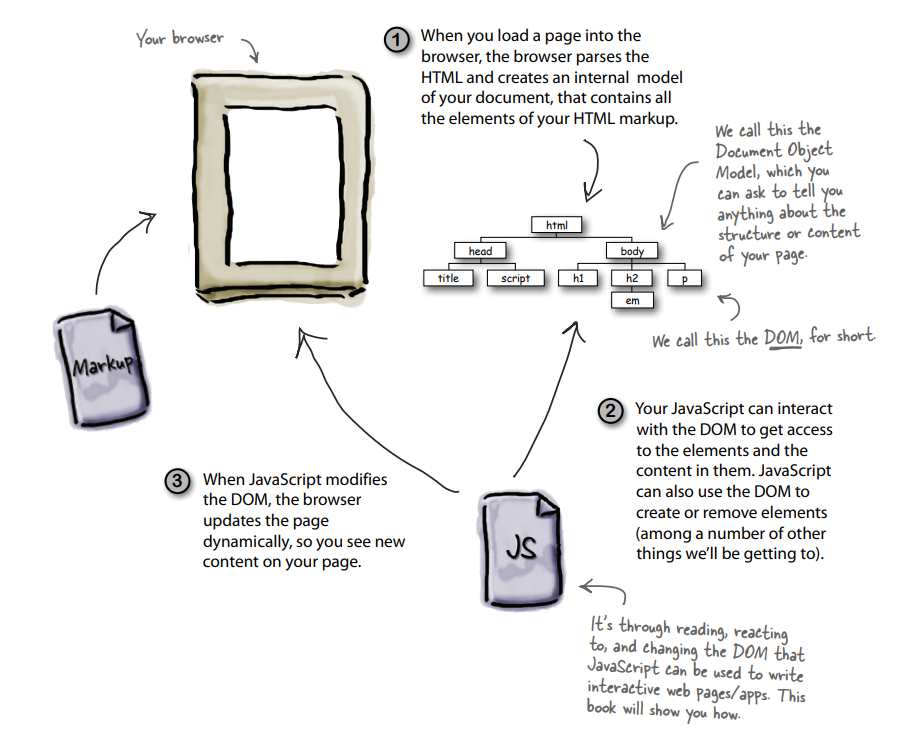
**Where to add JS**

.

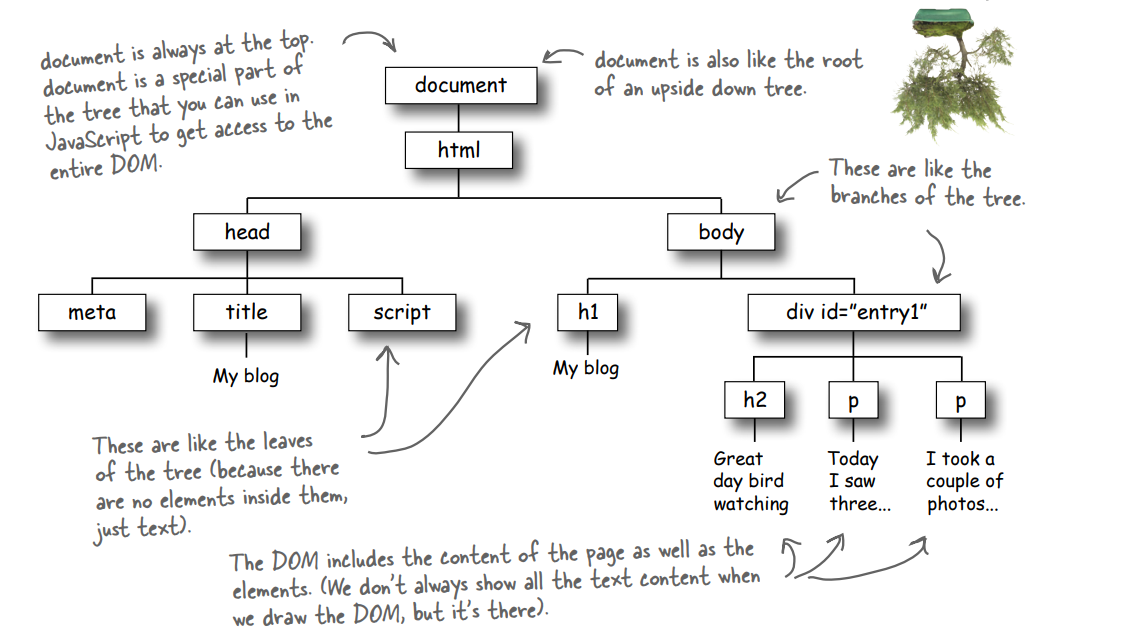


**How JavaScript interacts with your page**

JavaScript and HTML are two different things. HTML is markup and JavaScript is code. So how do you get JavaScript to interact with the markup in your page? You use the Document Object Model.



How DOM looks



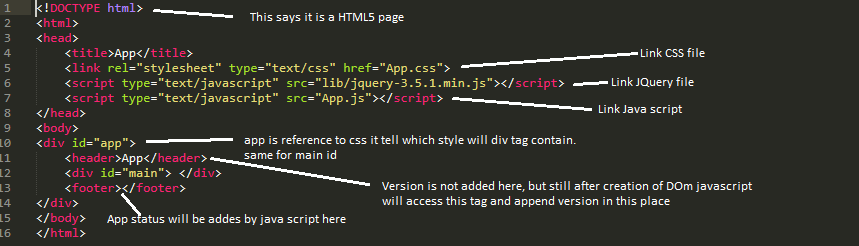
**Template**

Why we are making template, as you write more and more applications, you begin to notice that you are doing the same basic things over and over every time you get started, so it makes sense to create an application template to get started up quickly without reinventing the wheel every time.

You can understand what minimum basic things are in web app.

HTML

Img1

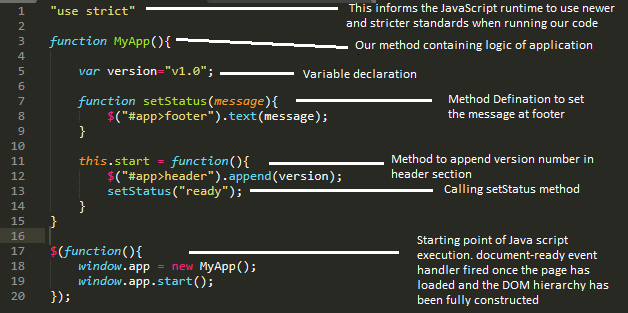


CSS

Refer CSS part from Img5

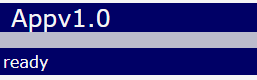
Java script

Img2



Template output

Img3



**JavaScript to update the DOM with example**

Let’s just create an HTML5 document with a list element to hold the task list:

HTML

Img4



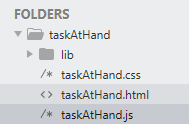
CSS and Java script

Img5



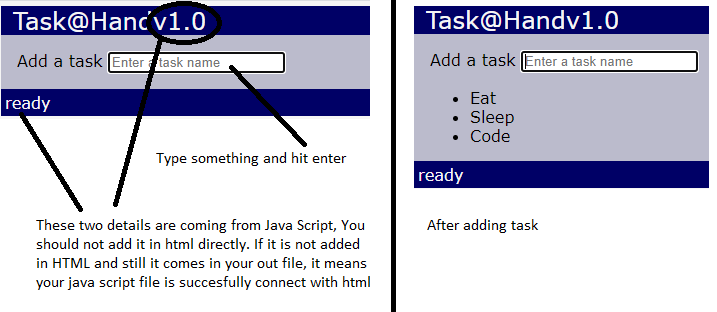
JS explained in Img7

Folder structure Img5



After completing above steps we should be able to see this result.

Output Img6



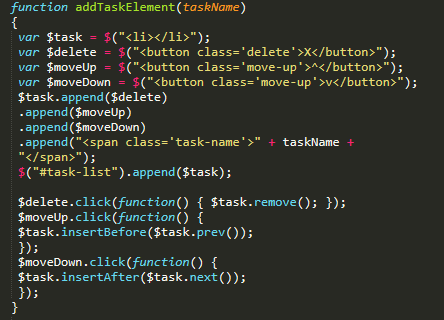
**Delete Element**

Only java script will change



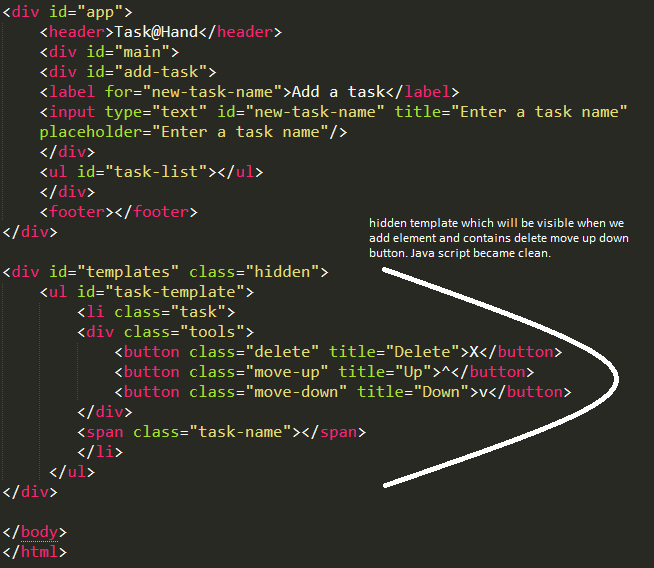
Moving tasks within the list

Change addTaskElement method

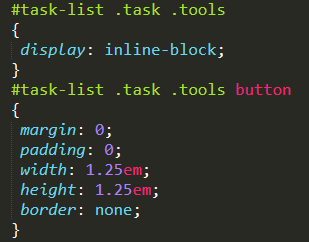


HTML templates

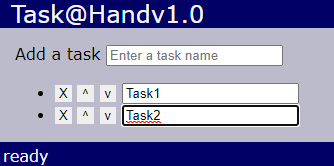
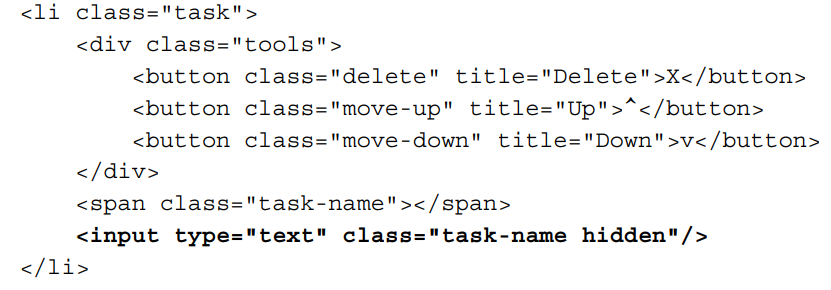
As you can see, things are getting a little messy in our addTaskElement() method. We are creating a bunch of elements programmatically in JavaScript and manually adding them to the DOM. we'll create an HTML template that we can reuse to easily create new tasks.



Add this to old CSS



**Editing a task in the list**



Thank you for reading