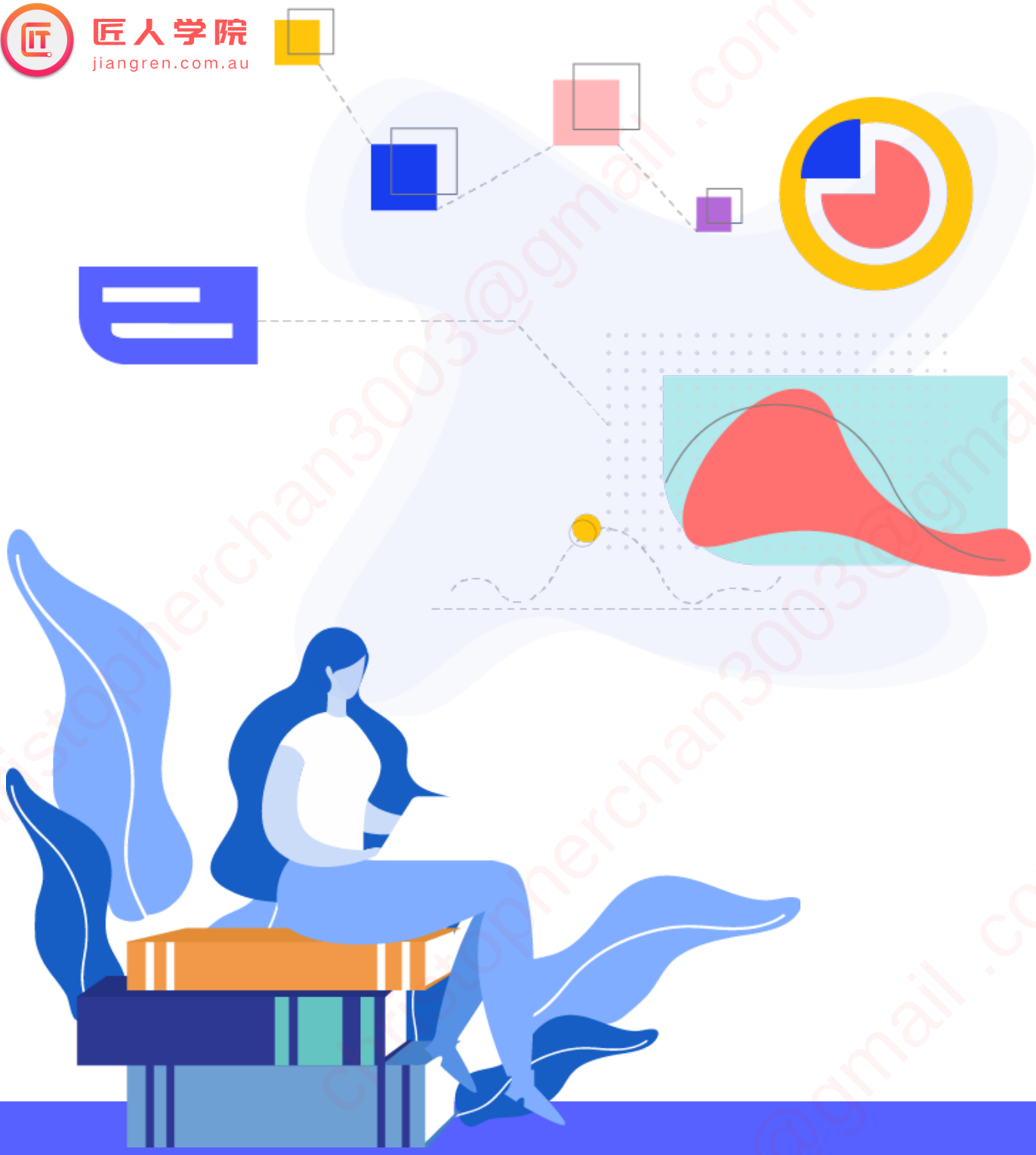
匠人学院
jiangren.com.au



HTML面试宝典

面试题

1.What does a DOCTYPE do?

DOCTYPE is an abbreviation for “document type". It is a declaration used in HTML to distinguish between standards mode and quirks mode. Its presence tells the browser to render the web page in standards mode.
Moral of the story - just add <!DOCTYPE html> at the start of your page.

2.Describe the difference between a cookie, sessionStorage and localStorage.

All the above-mentioned technologies are key-value storage mechanisms on the client side. They are only able to store values as strings.

	cookie	localStorage	sessionStorage
Initiator	Client or server. Server can use Set-Cookieheader	Client	Client
Expiry	Manually set	Forever	On tab close
Persistent across browser sessions	Depends on whether expiration is set	Yes	No
Have domain associated	Yes	No	No
Sent to server with every HTTP request	Cookies are automatically being sent via Cookie header	No	No
Capacity (per domain)	4kb	5MB	5MB
Accessibility	Any window	Any window	Same tab

3. Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>? Do you know any exceptions?

Placing <link>s in the <head>
Putting <link>s in the head is part of the specification. Besides that, placing at the top allows the page to render progressively which improves the user experience. The problem with putting stylesheets near the bottom of the document is that it prohibits progressive rendering in many browsers, including Internet Explorer. Some browsers block rendering to avoid having to repaint elements of the page if their styles change. The user is stuck viewing a blank white page. It prevents the flash of unstyled contents.

Placing <script>s just before </body>
<script>s block HTML parsing while they are being downloaded and executed. Downloading the scripts at the bottom will allow the HTML to be parsed and displayed to the user first.
An exception for positioning of <script>s at the bottom is when your script contains document.write(), but these days it's not a good practice to use document.write(). Also, placing <script>s at the bottom means that the browser cannot start downloading the scripts until the entire document is parsed. One possible workaround is to put <script> in the <head> and use the defer attribute.

4.span vs div

- div is a block element and span is inline element.
- Extra: It is illegal to put block element inside inline element. div can have a p tag and a p tag can have a span. However, span can't have a div or p tag inside.

div, section & article

- <section>, group of content inside is related to a single theme, and should appear as an entry in an outline of the page. It’s a chunk of related content, like a subsection of a long article, a major part of the page (eg the news section on the homepage), or a page in a webapp’s tabbed interface. A section normally has a heading (title) and maybe a footer too.
- <article>, represents a complete, or self-contained, composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user- submitted comment, an interactive widget or gadget, or any other independent item of content.
- , on the other hand, does not convey any meaning, aside from any found in its class, lang and title attributes

div, section & article

- Extra: Authors are strongly encouraged to view the div element as an element of last resort, for when no other element is suitable. Use of more appropriate elements instead of the div element leads to better accessibility for readers and easier maintainability for authors.

面试题

5.How do you serve a page with content in multiple languages?

use i18n

6. What is semantic HTML?

- Semantic HTML, or "semantically-correct HTML", is HTML where the tags used to structure content are selected and applied appropriately to the meaning of the content.
- for example, (for bold), and (for italic) should never be used, because they’re to do with formatting, not with the meaning or structure of the content.
- Instead, use the replacements and (meaning emphasis), which by default will turn text bold and italic (but don’t have to do so in all browsers), while adding meaning to the structure of the content.

7.Why you would like to use semantic tag?

- Search Engine Optimization, accessibility, repurposing, light code.
- Many visually impaired person rely on browser speech and semantic tag helps to interpret page content clearly.
- Search engine needs to understand page content to rank and semantic tag helps.

8.How do you serve a page with content in multiple languages?

- The question is a little vague, I will assume that it is asking about the most common case, which is how to serve a page with content available in multiple languages, but the content within the page should be displayed only in one consistent language.
- When an HTTP request is made to a server, the requesting user agent usually sends information about language preferences, such as in the Accept-Language header. The server can then use this information to return a version of the document in the appropriate language if such an alternative is available. The returned HTML document should also declare the lang attribute in the <html> tag, such as <html lang="en">...</html>.
- In the back end, the HTML markup will contain i18n placeholders and content for the specific language stored in YML or JSON formats. The server then dynamically generates the HTML page with content in that particular language, usually with the help of a back end framework.

9.What kind of things must you be wary of when designing or developing for multilingual sites?

- Use lang attribute in your HTML.
- Directing users to their native language - Allow a user to change his country/language easily without hassle.
- Text in images is not a scalable approach - Placing text in an image is still a popular way to get good-looking, non-system fonts to display on any computer. However, to translate image text, each string of text will need to have it's a separate image created for each language. Anything more than a handful of replacements like this can quickly get out of control.
- Restrictive words/sentence length - Some content can be longer when written in another language. Be wary of layout or overflow issues in the design. It's best to avoid designing where the amount of text would make or break a design. Character counts come into play with things like headlines, labels, and buttons. They are less of an issue with free- flowing text such as body text or comments.
- Be mindful of how colors are perceived - Colors are perceived differently across languages and cultures. The design should use color appropriately.
- Formatting dates and currencies - Calendar dates are sometimes presented in different ways. Eg. "May 31, 2012" in the U.S. vs. "31 May 2012" in parts of Europe.
- Do not concatenate translated strings - Do not do anything like "The date today is " + date. It will break in languages with different word order. Use a template string with parameters substitution for each language instead. For example, look at the following two sentences in English and Chinese respectively: I will travel on {% date %} and {% date %} 我会出发. Note that the position of the variable is different due to grammar rules of the language.
- Language reading direction - In English, we read from left-to-right, top-to-bottom, in traditional Japanese, text is read up-to-down, right-to-left.

10.What are data- attributes good for?

Before JavaScript frameworks became popular, front end developers used data-attributes to store extra data within the DOM itself, without other hacks such as non-standard attributes, extra properties on the DOM. It is intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements.
These days, using data- attributes is not encouraged. One reason is that users can modify the data attribute easily by using inspect element in the browser. The data model is better stored within JavaScript itself and stay updated with the DOM via data binding possibly through a library or a framework.

11.Consider HTML5 as an open web platform. What are the building blocks of HTML5?

• Semantics - Allowing you to describe more precisely what your content is.
• Connectivity - Allowing you to communicate with the server in new and innovative ways.
• Offline and storage - Allowing webpages to store data on the client- side locally and operate offline more efficiently.
• Multimedia - Making video and audio first-class citizens in the Open Web.
• 2D/3D graphics and effects - Allowing a much more diverse range of presentation options.
• Performance and integration - Providing greater speed optimization and better usage of computer hardware.
• Device access - Allowing for the usage of various input and output devices.
• Styling - Letting authors write more sophisticated themes.

12. Describe the difference between <script>, <script async> and <script defer>.

- <script> - HTML parsing is blocked, the script is fetched and executed immediately, HTML parsing resumes after the script is executed.
- <script async> The script will be fetched in parallel to HTML parsing and executed as soon as it is available (potentially before HTML parsing completes). Use async when the script is independent of any other scripts on the page, for example, analytics.
- <script defer> - The script will be fetched in parallel to HTML parsing and executed when the page has finished parsing. If there are multiple of them, each deferred script is executed in the order they were encountered in the document. If a script relies on a fully- parsed DOM, the defer attribute will be useful in ensuring that the HTML is fully parsed before executing. There's not much difference in putting a normal <script> at the end of <body>. A deferred script must not contain document.write.

Note: The async and defer attributes are ignored for scripts that have no src attribute.

13.What is progressive rendering?

Progressive rendering is the name given to techniques used to improve the performance of a webpage (in particular, improve perceived load time) to render content for display as quickly as possible.
It used to be much more prevalent in the days before broadband internet but it is still used in modern development as mobile data connections are becoming increasingly popular (and unreliable)!

Examples of such techniques:

- Lazy loading of images - Images on the page are not loaded all at once. JavaScript will be used to load an image when the user scrolls into the part of the page that displays the image.
- Prioritizing visible content (or above-the-fold rendering) - Include only the minimum CSS/ content/scripts necessary for the amount of page that would be rendered in the users browser first to display as quickly as possible, you can then use deferred scripts or listen for the DOMContentLoaded/load event to load in other resources and content.
- Async HTML fragments - Flushing parts of the HTML to the browser as the page is constructed on the back end. More details on the technique can be found here.

14. Why you would use a srcset attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute.

You would use the srcset attribute when you want to serve different images to users depending on their device display width - serve higher quality images to devices with retina display enhances the user experience while serving lower resolution images to low-end devices increase performance and decrease data wastage (because serving a larger image will not have any visible difference). For example: <img srcset="small.jpg 500w, medium.jpg 1000w, large.jpg 2000w" src="..." alt="">tells the browser to display the small, medium or large .jpg graphic depending on the client's resolution. The first value is the image name and the second is the width of the image in pixels. For a device width of 320px, the following calculations are made:

500 / 320 = 1.5625
1000 / 320 = 3.125
2000 / 320 = 6.25

If the client's resolution is 1x, 1.5625 is the closest, and 500w corresponding to small.jpg will be selected by the browser.
If the resolution is retina (2x), the browser will use the closest resolution above the minimum. Meaning it will not choose the 500w (1.5625) because it is greater than 1 and the image might look bad. The browser would then choose the image with a resulting ratio closer to 2 which is 1000w (3.125).
srcsets solve the problem whereby you want to serve smaller image files to narrow screen devices, as they don't need huge images like desktop displays do — and also optionally that you want to serve different resolution images to high density/low-density screens.

面试题

15. Have you used different HTML templating languages before?

Yes, Pug (formerly Jade), ERB, Slim, Handlebars, Jinja, Liquid, just to name a few. In my opinion, they are more or less the same and provide similar functionality of escaping content and helpful filters for manipulating the data to be displayed. Most templating engines will also allow you to inject your own filters in the event you need custom processing before display.