

FRONT-END WEB DEVELOPMENT

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

There are several tools available that can be used to develop the front end of a website, and understanding which tools are best fit for specific tasks marks the difference between developing a hacked site and a well designed, scalable site.

HyperText Markup language (HTML)

Hyper Text Markup Language is the backbone of any website development process, without which a web page doesn't exist. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. After the development of the HTML there are many versions which came in the market of World Wide Web. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation. This version contains new and efficient ways of handling elements such as video and audio files. HTML5 is now very popular among front-end web developers. The HTML5 has lots of features in comparison to the older versions. After the development of HTML, a revolution came in the field of internet. Now, world started communicating by using the single language i.e.: HTML. The term HTML is made up of two parts. The first one is Hyper Text and the second one is Markup Language.

Style

CSS, cascading style sheets, a core functionality of front-end development, the styles that lay out the page and give it both its unique visual flair and a clear, user-friendly view to allow readers, who never linger on pages we would like to think they do, some help to read or skim the contents quickly.

Design means both how something looks and how something is structured, and in a good design, both come together.

An important aspect of styling is checking across several browsers and to write concise, terse code that is specific yet generic at the same time and displays well in as many renderers as possible, which leads me to the next point.

Programming

Although by this I mean mostly Javascript, this could apply to ActionScript, PHP or any other popular web languages developed for the front-end. Javascript has fully grown up from inline commands embedded in html to full-blown asynchronous applications executed on the fly on the browser as unobtrusive rich functionality.

The widespread usage of js libraries such as jQuery or MooTools has produced a plethora, some would even say an excess, of visual effects that turn web pages into a more three-dimensional immersive experience. A notoriously untyped language with a bad reputation that baffles programmers and scares scripters, javascript was developed specifically for the Web and, like it or not, it is here to stay.

Usability

Information architecture has blossomed in the past few years but as the people who build the site interact with the clients, the graphic designers, the backend developers and product managers, front-end last-stop position should mean flagging up details, suggesting improvements as well as taking part in usability testing.

Depending on the size of the team and allocated budget, sometimes a front-end developer is both designer, QA of the backend bugs, usability and accessibility tester and ia, which isn't an enviable position. A front-end developer probably looks at more websites and evaluates how they look and work more than other team members as a translator is more naturally interested in words and grammar than other people.

Performance

To build even faster sites, your markup, styles and javascript should be both scalable and nimble. A growing discipline, with the foreseeable rising costs of energy in the years to come and the problems caused by scaling performance in rich content sites with customised content (the web 2.0 meme), it is in the best interest for companies to reduce their page size footprint as much as possible to avoid rising bandwidth costs.

At the same time, the shift from desktop to online rich applications means that a heavier load on the browser should be alleviated with page reductions elsewhere and constant monitoring of new features.

Cross-browser, cross-platform, cross-device functionality

The browser on your computer is to remain the most advanced and feature-rich client application to access the web for a long time, but that doesn't mean that 'snacking' with mobile browsing, or netbooks, etc, isn't a rapidly growing in presence and importance.

Since the browser wars between Netscape and Internet Explorer on PCs, much has happened. Nowadays, browsers compete with each other for page-rendering speed, plugins and add-ons to achieve both a lean and comprehensive browser experience. As applications move to the cloud, gmail being a popular example, the browser becomes the OS, which puts further emphasis on coding, styling and programming for as many clients as possible. As Douglas Crockford [said](#) Browsers are the most hostile software development environment imaginable.

UX designer (user experience designer)

UX designers are primarily concerned with **how the product feels**. A given design problem has no single right answer. UX designers explore many different approaches to solving a specific user problem. The broad responsibility of a UX designer is to ensure that the product logically flows from one step to the next. One way that a UX designer might do this is by conducting in-person user tests to observe one's behavior. By identifying verbal and non-verbal stumbling blocks, they refine and iterate to create the "best" user experience. An example project is creating a delightful onboarding flow for a new user.

UI designer (user interface designer)

Unlike UX designers who are concerned with the overall feel of the product, user interface designers are particular about **how the product is laid out**. They are in charge of designing each screen or page with which a user interacts and ensuring that the UI visually communicates the path that a UX designer has laid out. For example, a UI designer creating an analytics dashboard might front load the most important content at the top, or decide whether a slider or a control knob makes the most intuitive sense to adjust a graph. UI designers are also typically responsible for creating a cohesive style guide and ensuring that a consistent design language is applied across the product. Maintaining consistency in visual elements and defining behavior such as how to display error or warning states fall under the purview of a UI designer.

In conclusion

Thucydides wrote 'Knowledge without understanding is useless'. Since its inception, the web has been primarily about information but this data-driven trend can just drown people with a flood of disconnected, random info factoids that few can grasp and even fewer be interested in.

The goal of a front end developer is to create clear, easy, fast pages and interfaces that will make people understand and care about the information, by putting it in context, expose its legitimacy or lack thereof, and reveal their implicit or explicit interconnection.

Front-end is not just a pretty face, it's the friendly, forward-looking interface of web development.

REFERENCES

1. Codesido I. What is front-end development? [Электронный ресурс] / Ivan Codesido // The Guardian. – 2009. – Режим доступа до ресурсу: <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>.