Etanova Enterprise Solutions

Front End Development » 2017-10-21

http://www.etanova.com/technologies/front-end-development

Contents

HTML 5

Develop websites and web applications using modern HTML 5 technologies that will enhance user experience! Maximize web browser performance and create dynamic content that will be intuitive for website users.

Rich Internet Applications

A significant part of IT infrastructure can be moved to the front end with Rich Internet Applications (RIA) that deliver the features and functionality typically associated with platform dependant desktop software applications. RIA software runs inside the web browser and does not require the installation of additional software. Thus RIA powered web applications enable the portability and flexibility required by service oriented software development.

Web Browser Hardware Acceleration

With hardware accelerated HTML 5 features, web applications have the ability of emulating a desktop-like environment. Use new features including

- Canvas, SVG, and WebGL: for GPU accelerated graphic design and animations
- DOM Transitions and Transformations: for truly interactive and dynamic web content
- Scalable Vector Graphics: which save bandwidth and reduce infrastructure server resources.
- HTML 5 Elements: to take advantage of new web browser features without the overhead of a scripting language.

Semantic Web

Add semantic meaning to HTML development with semantic web in order to provide a tool for other websites or search engines to make sense of the information presented on a website. Using semantic web methodologies as a form of effective Search Engine Optimization will help increase a website's search rankings and popularity on the internet.

Backwards Compatibility

New features are great until they are tested out on an older incompatible web browsers that provide limited support. Web applications that are backwards compatible are able to provide useful content to legacy web browsers.

W3C Validated HTML

A website that looks great is not enough unless the code also looks great. W3C Validated HTML will ensure that HTML 5 code meets the right specifications and that it does not violate W3C guidelines. A W3C Validated website has a much better chance of being cross-platform and cross-browser compatible.

Cascading Style Sheets

Create beautiful and responsive graphical user interfaces for your websites, which will be accessible for users on any hardware device or software platform. Ensure dynamic web application interfaces operate quickly and efficiently by using optimized CSS that will maximize performance.

Dynamic StyleSheet Extensions (LESS or SASS)

Design interfaces with dynamic stylesheet language extensions such as LESS or SASS. Develop CSS code more rapidly using backend modules that detect changes in extended CSS files, pre-compile their code to regular CSS and then cache it for the front end web browser, so that it does not have to be reparsed with each request.

Mobile Compatible CSS

Mobile compatible CSS will enhance user web browsing experience on mobile devices such as phones or tablets. This is accomplished by using specialized CSS for mobile devices which will ensure mobile compatibility and reduce the amount of hardware resources required, which may slow down mobile devices with limited resources. With an increasing number of users using these devices, mobile friendly CSS is an integral part of any modern web application.

Responsive CSS Design

Website and web applications must respond to various platforms and screen sizes. Where screen size is a constraint, responsive CSS will effectively utilize available space by maximizing the visibility of application critical content. Responsive websites will display correctly on desktop computers, mobile devices and tablets.

Cross Browser Compatibility

CSS is a truly amazing technology, except for when it works only on a particular browser. With cross-browser compatible CSS, a website's interface will display consistently across all web browser platforms, including legacy versions of Internet Explorer.

W3C Validated CSS

Maximize website interface compatibility and performance by using stylesheets that have been validated against W3C web standards. Websites have a high probability of meeting cross platform compatibility requirements if they pass W3C CSS Validation.

Scripting

Web browsers were initially created to view static content. However modern websites are

dynamic and must respond to user interaction instantly, without taking users to a new web page every time an event occurs such as a mouse click. For this reason, front end scripting languages add a layer of dynamic interaction to websites and web applications. With effective scripting, it is possible to extend the functionality of web applications far beyond the original scope of HTML. Use scripting to develop Web 2.0 browser-based applications and dynamic widgets using JavaScript libraries and frameworks.

Native JavaScript

Develop cross-browser compatible code in native JavaScript. Modular JavaScript design patterns will allow for a large front end architecture that scales well. Use advanced prototypes, functional inheritance, delegations and other structural, behavioural and creational patterns. Avoid anti-patterns and misuse of JavaScript that may degrade a web applications performance.

jQuery Library

Develop Dynamic HTML with the jQuery library for optimized DOM traversal and manipulation. Use API functions such as event handling and animations to simplify tasks that would be too complex to write in cross browser compatible native JavaScript. Extend jQuery with custom web application plugins. In specialized situations, combine jQuery with native JavaScript for an effective solution.

AngularJS Framework

Leverage the power of AngularJS for large or enterprise level web applications. The framework will build rich client-side web applications by extending the limited HTML vocabulary so that it can be used for dynamic views. AngularJS offers many useful features such as: two way data binding, templates, server communication and a full testing environment.

Modernizr

When building modern web applications, use modernizr as an tool for detecting available browser features for the purpose of cross-browser compatibility. Using this tool allows websites and rich web applications to create feature-rich functionality without sacrificing accessibility. Modernizr will allow scripting languages to detect which features web browsers support so that applications can correctly handle missing features on legacy browsers.

Web Services

In order to add more functionality to the front end through scripting languages, effective communication is required between client and server nodes. Web services allow for data interchange between nodes through various methods. The appropriate method for communication depends on the web application scope and complexity.

REST

Exchange information between server and client nodes with HTTP RESTful data interchange for performance and scalability. REST creates web services based on URIs and HTTP, permitting many different data formats for request communication including XML and JSON. For security, REST web services may use SSL. REST is also cross-browser compatible for modern web browsers.

SOAP

Expose server side application logic to the web with SOAP for enterprise level features. SOAP offers WS-Security to verify node identities and provides data integrity and privacy services. Moreover, SOAP offers WS-AtomicTransaction for ACID transactions and WS-ReliableMessaging for node communication reliability.

AJAX

Communicate seamlessly between front end and backend nodes using AJAX. Web applications become more usable with AJAX as web browser users will not have their web page refreshed each time they issue a HTTP request. Avoid negative SEO implications by using AJAX only in the right context and not misusing the technology.

WebHooks

Accept HTTP requests as a URL route centralized service with web hooks. A web hook will act as a real-time API component for a web application. It will allow an application to trigger certain events, or to act as a service oriented application based on triggers sent as HTTP requests to particular URLs.

Web Sockets

Open a persistent connection between two nodes with web sockets, and reduce much of the overhead that can incur with AJAX in larger web applications. Full duplex web sockets allow two way messaging between nodes such that front end web clients can receive push notifications from the backend server. Push data to web browsers with various server side technologies such as Java Messaging Service (JMS), Windows Communication Foundation (WCF), Amazon Simple Notification Service (SNS), etc.

Front End Frameworks

Robust front end frameworks will provide websites with a toolkit of ready-to-use features that can setup website layouts and elements without coding everything from the ground up. HTML & CSS design templates are provided by frameworks to enhance typography, forms, buttons, navigation and interface components. Using frameworks can have many benefits:

- Rapid website prototyping

- Benefit from well tested code
- Scalability of a website
- Save development type

Foundation Framework

The Foundation Framework philosophy is rapid development of prototypes. Foundation uses the dynamic cascading stylesheets LESS and SASS to provide utility classes and features to developers. A main advantage to Foundation is that the framework does not impose any particular design patterns or templates. Rather, it allows developers to decide on beneficial features that can be integrated into websites or web applications.

Twitter Bootstrap Framework

The Twitter Bootstrap philosophy is a design-it-yourself approach by providing a toolbox of ready-to-use components. Websites built upon the Twitter Bootstrap Framework have the advantage of using many pre-stylized web elements and layouts.

Charting APIs

Display complex datasets in a way that makes sense to front end users through web based charts. Charting APIs offer many options to attach existing web application data to various charts, graphs and illustrations. Some advantages to using charting technologies include:

- Aggregate large amounts of data into charts providing concise and useful information.
- Display server side data in graphical format on the user web interface.
- Help users quickly make sense of complex information.
- Create better user interface design and increase the web browsing experience of end users.

D3 JS Charts

Bind data to a web application's Document Object Model and then apply data-driven transformations to the document, resulting in manipulation of a web application's interface based on some arbitrary data. D3 Charts has very little overhead and is extremely fast in comparison to other charting frameworks.

HighCharts

Create truly interactive charts that respond to mouse events such as hover and click. Dynamic charting allows users to modify series and points or modify axes by interacting with the chart. Highcharts is compatible with mobile phones and tablets, and also backwards compatible with legacy web browsers including back to Internet Explorer 6.

Dojo Toolkit

Maximize cross-browser compatibility with Dojo, which will attempt to use a graphic technology best compatible with the user's web browser. Dojo defaults to SVG, and then attempts to use Canvas, then VML then Silverlight in that order. Using Dojo is an ideal end charting toolkit to generate web application critical charts and diagrams.