

MTA Exam 98-375: HTML5 Application Development Fundamentals

About This Exam

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the ***foundational knowledge*** needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

This exam is designed to provide candidates with an assessment of their knowledge of *fundamental* HTML5 application development concepts. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams.

Audience Profile

Candidates for this exam are seeking to prove core HTML5 client application development skills that will run on today's touch-enabled devices (PCs, tablets, and phones). Although HTML is often thought of as a web technology that is rendered in a browser to produce a UI, this exam focuses on using HTML5, CSS3, and JavaScript to develop client applications. Before taking this exam, candidates should have solid ***foundational knowledge*** of the topics outlined in the preparation guide, including CSS and JavaScript. It is recommended that candidates be familiar with the concepts of and have some hands-on experience with the related technologies either by taking relevant training courses or by working with tutorials and samples available on MSDN and in Microsoft Visual Studio.

Objective Domain

1. Manage the Application Life Cycle

1.1. Understand the platform fundamentals.

This objective may include but is not limited to: packaging and the runtime environment: app package, app container, credentials/permission sets, host process, leveraging existing HTML5 skills and content for slate/tablet applications

1.2. Manage the state of an application.

This objective may include but is not limited to: manage session state, app state, and persist state information; understand states of an application

1.3. Debug and test an HTML5-based touch-enabled application.

1.4. Publish an application to a store.

This objective may include but is not limited to: Windows Store; third-party stores

2. Build the User Interface by Using HTML5

2.1. Choose and configure HTML5 tags to display text content.

2.2. Choose and configure HTML5 tags to display graphics.

This objective may include but is not limited to: when, why, and how to use Canvas; when, why, and how to use SVG

2.3. Choose and configure HTML5 tags to play media.

This objective may include but is not limited to: video and audio tags

2.4. Choose and configure HTML5 tags to organize content and forms.

This objective may include but is not limited to: tables, lists, sections; semantic HTML

2.5. Choose and configure HTML5 tags for input and validation.

3. Format the User Interface by Using CSS

3.1. Understand the core CSS concepts.

This objective may include but is not limited to: separating presentation from content – create content with HTML and style content with CSS; managing content flow - inline vs. block flow; managing positioning of individual elements – float vs. absolute positioning; managing content overflow – scrolling, visible, and hidden; basic CSS styling

3.2. Arrange user interface (UI) content by using CSS.

This objective may include but is not limited to: using flexible box and grid layouts to establish content alignment, direction, and orientation; proportional scaling and use of “free scape” for elements within a flexible box or grid; ordering and arranging content; concepts for using flex box for simple layouts and grid for complex layouts; grid content properties for rows and columns; using application templates

3.3. Manage the flow of text content by using CSS.

This objective may include but is not limited to: regions and using regions to flow text content between multiple <div> sections – content source, content container, dynamic flow, flow-into, flow-from, msRegionUpdate, msRegionOverflow, msGetRegionContent(); columns and hyphenation and using these CSS settings to optimize the readability of text; using “positioned floats” to create text flow around a floating object

3.4. Manage the graphical interface by using CSS.

This objective may include but is not limited to: graphics effects - rounded corners, shadows, transparency, background gradients, typography, and Web Open Font Format; 2D and 3D transformations – translate, scale, rotate, skew, and 3D perspective transitions and animations; SVG filter effects; Canvas)

4. Code by Using JavaScript

4.1. Manage and maintain JavaScript.

This objective may include but is not limited to: creating and using functions; using Windows Library for JavaScript, jQuery, and other third-party libraries

4.2. Update the UI by using JavaScript.

This objective may include but is not limited to: locating/accessing elements; listening and responding to events; showing and hiding elements; updating the content of elements; adding elements

4.3. Code animations by using JavaScript.

This objective may include but is not limited to: using the animation library

4.4. Access data access by using JavaScript.

This objective may include but is not limited to: sending and receiving data; transmitting complex objects and parsing; accessing databases and indexed DB; loading and saving files; App Cache

4.5. Respond to the touch interface.

This objective may include but is not limited to: gestures, how to capture and respond to gestures

4.6. Code additional HTML5 APIs.

This objective may include but is not limited to: GeoLocation, Web Workers, Web Sockets

4.7. Access device and operating system resources.

This objective may include but is not limited to: Windows Runtime (WinRT); in memory resources such as contact lists and calendar; hardware capabilities such as GPS, accelerometer and camera