MTA: HTML5 Application Development Fundamentals – Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Exam 98-375: HTML5 Application Development Fundamentals

Manage the application life cycle (20–25%)

Understand the platform fundamentals

- packaging and the runtime environment: app package, app container, credentials/permission sets, host process, leverage existing HTML5 skills and content for slate/tablet applications

Manage the state of an application

- manage session state, app state, and persist state information; understand states of an application; understand the differences between local and session storage

Debug and test an HTML5-based, touch-enabled application

- touch gestures; understand which gestures you test on a device

Build the user interface (UI) by using HTML5 (25–30%)

Choose and configure HTML5 tags to display text content

Choose and configure HTML5 tags to display graphics

- when, why, and how to use Canvas; when, why, and how to use scalable vector graphics (SVG)

Choose and configure HTML5 tags to play media

- video and audio tags

Choose and configure HTML5 tags to organize content and forms

- tables, lists, sections; semantic HTML

Choose and configure HTML5 tags for input and validation

Format the user interface by using Cascading Style Sheets (CSS) (20–25%)

Understand the core CSS concepts

- separate presentation from content (create content with HTML and style content with CSS); manage content flow (inline versus block flow); manage positioning of individual elements(float versus absolute positioning); manage content overflow (scrolling, visible, and hidden); basic CSS styling

Arrange UI content by using CSS

- use flexible box and grid layouts to establish content alignment, direction, and orientation; proportional scaling and use of "free scale" for elements within a flexible box or grid; order and arrange content; concepts for using flex box for simple layouts and grid for complex layouts; grid content properties for rows and columns; use application templates

Manage the flow of text content by using CSS

- regions and using regions to flow text content between multiple 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; use "positioned floats" to create text flow around a floating object

Manage the graphical interface by using CSS

- graphics effects (rounded corners, shadows, transparency, background gradients, typography, and Web Open Font Format); two-dimensional (2-D) and three-dimensional (3-D) transformations (translate, scale, rotate, skew, and 3-D perspective transitions and animations); SVG filter effects; Canvas

Code by using JavaScript (30–35%)

Manage and maintain JavaScript

- create and use functions; jQuery and other third-party libraries

Update the UI by using JavaScript

- locate/access elements; listen and respond to events; show and hide elements; update the content of elements; add elements

Code animations by using JavaScript

- use animation; manipulate the canvas; work with images, shapes, and other graphics

Access data access by using JavaScript

- send and receive data; transmit complex objects and parsing; load and save files; App Cache; datatypes; forms; cookies; localStorage

Respond to the touch interface

- gestures, how to capture and respond to gestures

Code additional HTML5 APIs

- GeoLocation, Web Workers, WebSocket; File API

Access device and operating system resources

- in- memory resources, such as contact lists and calendar; hardware capabilities, such as GPS, accelerometer, and camera