JavaScript

Lightning overview -

not meant to be a tutorial

What is JavaScript

- High level programming language
  - Dynamic typing
  - Object orientation (prototype based)
- Multi-paradigm
  - Event-driven
  - Functional - composition of functions, functions as objects
  - Imperative - direct computation through procedures and functions
- Relatively easy to learn
  - similarities with Python, C/C++, Java (no direct relationship)

Why JavaScript?

- Most web browsers have a dedicated engine
  - Designed from the ground up for the web
- APIs:
  - Text, dates, regular expressions (pattern matching)
  - Standard data structures (dictionaries, ...)
  - Document Object Model - manipulate the browser
  - No native IO (no file access etc.) but provided through APIs
- Most power when used for DOM manipulation

Basic Examples

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript#inline_javascript_handlers

- Variables and basics: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#what_is_a_variable
-

Learning Resources

- https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics
  - https://mdn.github.io/beginner-html-site-scripted/

- https://developer.mozilla.org/en-US/docs/Learn/JavaScript
- https://learnjavascript.online/

Custom Elements

Adding custom elements

- XML allows arbitrary namespaces and tag definitions
  - Applications can be defined on top of XML tag definitions
- But HTML5 does not use the same approach
- Requirement for elements:
  - Meaning: what does a tag mean - <title>, <h1> etc OK, but is <my-button> actually a button?
  - Rendering: how should a tag be shown: provide display details for each tag
  - States: checkbox can be checked or blank - what about custom tags?
  - Customized built-in element or Autonomous custom element?

https://html.spec.whatwg.org/multipage/custom-elements.html

Web Components

- Custom elements
  - JS API to create custom element tags
- Shadow DOM
  - API to keep styling of component separate from rest of page
- HTML Templates
  - <template> and <slot> tags to write markup templates

Web Component Examples

- https://github.com/mdn/web-components-examples
- https://mdn.github.io/web-components-examples/editable-list/
- https://mdn.github.io/web-components-examples/edit-word/
- https://mdn.github.io/web-components-examples/word-count-web-component/
- https://css-tricks.com/an-introduction-to-web-components/

Summary

- Custom Elements: API to extend HTML5 element/tag capabilities
- Shadow DOM: restrict scope of styling or modification of content
- HTML Templates

Combined: Web Components

- Goal: reuse
- Problem: limited standardization