User Centered Design

Wednesday, January 25, 2023 5:01 PM

You != Users

Users !- designers/devs

CLS

Friday, January 27, 2023 5:04 PM

Def: Cumulative Layout Shift

- Amount that elements move around the page

Element Flow

Monday, January 30, 2023 5:34 PM

HTML elements have built-in styles called user-agent styles

- Inconsistent between browsers

Flow:

- HTML elements rendered top to bottom, and flows around elements
- Block flow: assumes elements take up 100% of one dimension and flow around each other
- Inline flow: elements flow inside another container, often left to right

Responsive vs Adaptive

Friday, February 3, 2023 5:00 PM

Responsive: page can respond to changes in UI contexts (window width)

Adaptive: page adapts to Display contexts, UI contexts, device context

Markup, HTML, XHTML

Friday, January 20, 2023 5:03 PM

Want to organize content and data

XML: allows you to define an arbitrary markup language

HTML: Hypertext (documents that can link to each other) Markup Language

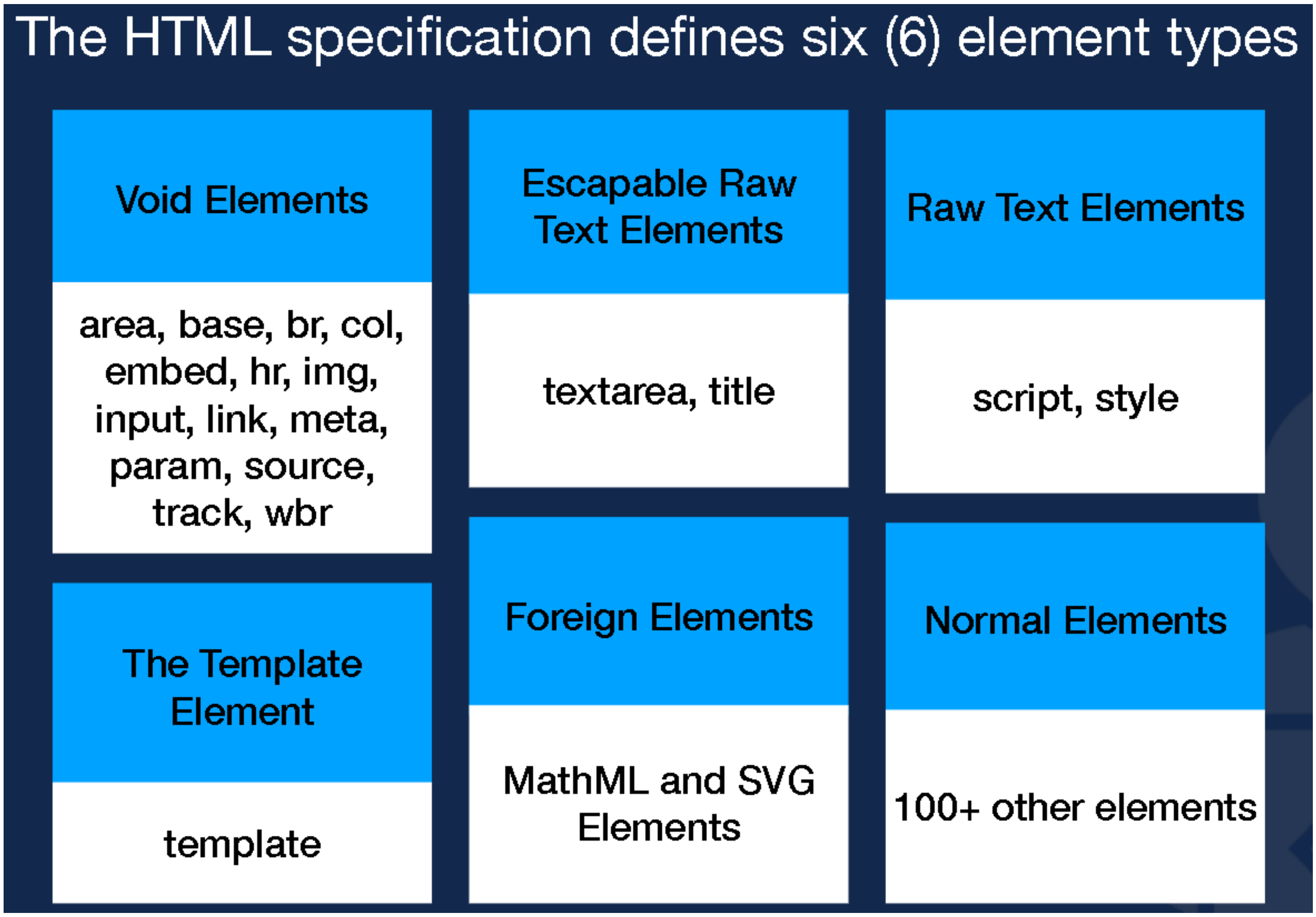
Element: type of object
Tag: instance of element
Empty element: does not contain data, may be <tag/>

- Often ignores white space

XHTML: Strict HTML, which follows an exact schema
- Hard to follow, but guaranteed to be exact

HTML Elements

Monday, January 23, 2023 5:35 PM



The HTML specification defines six (6) element types

Void Elements	Escapable Raw Text Elements	Raw Text Elements
area, base, br, col, embed, hr, img, input, link, meta, param, source, track, wbr	textarea, title	script, style
The Template Element	**Foreign Elements**	**Normal Elements**
template	MathML and SVG Elements	100+ other elements

```
Void: do not enclose any content
Escapable Raw: escape tags but not character entities
(&…)
Raw: contains other technologies (js, css) and are not
parsed
Template: defines a custom element
Foreign: 3rd party technology
Normal: Everything else, contains content
```

Images, Picture

Friday, January 27, 2023 5:02 PM

- Dimensions: Height/Width used to allocate space for an image, reduces CLS
- Alt: Text describing the image
- Src: Link to the image source
    - Formats: bitmap, vector
        - gif: animated, compressed - does well in low complexity
        - jpeg: compressed - does well in high complexity
        - png: lossless
        - webp: Versatile, smaller than png or jpeg
        - avif: sometimes smaller than webp
        - svg: scalable
- loading="lazy": tells browser to laod the image when needed
- fetchpriority="high": tells the browser to prioritize fetching this image

Notes:

- Images increasing the share of data sent
- Images drive conversion, needed to keep users interested
- Send less data, less often, from nearby, when you should

Picture:

- List multiple sources and displays the first match as an image

Forms & Form Submission

Friday, January 27, 2023 5:19 PM

- For user experience, **<u>not for validation or security</u>**
- Client side is inherently unsafe
- Hidden field: not show to the user, useful for including data to request without showing it

Notes:

- Who: Mostly users, sometimes programs
- What: Serialize the data (key=value&…)
- Where: send data to API endpoint
- When: on submit button
- How: method and action attributes

```
action = "collect.php" <- where
method = "post" <- how
enctype = "mime" <- format
target = "name/id" <- result
```

1) On submission
   - Press submit button
   - On return key
   - Using JS submit();
2) Check validity using HTML
3) Check validity using JS
4) Read all fields not disabled
5) In body: field_name=field_value&field_name=field_value& …
   - Encode in body for POST
   - Encode in URL for GET
   - Mime type: "application/x-www-form-urlencoded"

Tables

Friday, January 27, 2023 5:31 PM

Represents a collection of data

Soup Parsing

Monday, January 23, 2023 5:16 PM

Browser tries to parse the document as best as possible

Pros:

- Easy for the developer

Cons:

- Browser infers tags (may be inconsistent behavior)

Dividis, Semantic Markup Makes Landmarks

Monday, January 23, 2023 5:17 PM

<div>: generic block tag
<span>: generic inline tag

Poor readability

Semantic markup makes landmarks

- When you correctly use semantic markup like <header>, <footer>, <nav>, <main>, headings (<h1>, <h2>), <section>, <article>, and so on you provide understood landmarks for assistive devices and search engines to better understand the structure and meaning of your document

CSS

Monday, January 30, 2023 5:08 PM

Ways to define styles:
   Inline: add style attribute to single html element
   Document Wide: apply styles to an entire page by using a style element in the header
   Stylesheet: generalize rules and link styles in each document
   Import: import styles from an external source

Syntax:

```
   selector {
      property-name : value; - declaration  } rule
      Property-name : value;
   }
```

ID: unique identification for css selectors
Class: group of elements which can be selected in css

Media Queries: specify styles based on certain media constraints
- Avoids needing JS
- Part of responsive web design
- media="printer", can be used to specify styles for printing

User style sheet: Users can override css styles

FOUC: Flash of unstyled content - moment where content is unstyled and then suddenly becomes styled, can be a side effect of imported styles

Inheritance:
- Styles may trickle down the DOM tree from
- Some styles are not inherited, ex: border

Selectors

Wednesday, February 1, 2023 5:21 PM

Element - select all of the element specified
   Ex: element {}

ID - select a tag with specified ID
   Ex: #id {}
   Ex: element #id {} - select a tag of type element and has id

Class - select all tags that are member of the specified class
   Ex: .class {}
   Ex: element.class {} - select all tags of type element and member of class
   Note: don't use too many classes per element

Grouping: select all tags which match any selector
   Ex: h1, h2, h3 {}

Descendent: select descendent from ancestor
   Ex: ancestor descendent {}

Child: select direct child from parent
   Ex: parent > child {}

Adjacent: select sibling2 next to sibling1
   Ex: sibling1 + sibling2 {}

Pseudo Classes: Selects elements with a particular state, ie hover, enabled, etc

Rule Priority

Wednesday, February 1, 2023 5:33 PM

In general: the more specific the rule, the more priority it has

!important:

- Useful tool to prevent rule from being overridden
- Too much use can lead to poor maintainability

Values and Units

Friday, February 3, 2023 5:04 PM

Absolute: generally the same size on all devices
pt: 1/72 in
px: 1 pixel

Relative: depends on an different value
rem: root element
em: parent element
ex: height of x character
ch: width of 0 character
vw, vh: viewport width and height

- May be incorrect because of weird display shapes

Fonts

Friday, February 3, 2023 5:20 PM

Purpose: Differentiating text by purpose, but not grotesque

Challenges:
- Lack of differentiation between letters, leading to confusion

Serif Fonts: fonts with small lines and edges to each character
- Can communicate trust and authority
- Used in print

Sans Serif Fonts: literally without serif
- Looks more modern

Script Fonts: Handwritten fonts
- Grabs attention
- Hard to read

Monospaced: all characters take the same space
Proportional: each character takes the space it needs

JS, Defensive Coding

Friday, February 10, 2023 5:23 PM

Core language: ECMA Script (ES)

- Syntax
- Types
- Basic objects

Host environment

- Browser
- Nodejs (OS)

Quirks:

- Weak typing
- Asynchronous

Defensive Coding:

- Encapsulate code and assume the worst
    - Script modules
- Concerns
    - Variable and function name conflicts
    - Load order and network concerns
    - Poor error handling
    - Event rebinding
    - Browser quirks

Syntax, Variables, Parameter Passing, Equality

Wednesday, February 15, 2023 5:21 PM

Top to bottom execution
Case sensitive
Whitespace ignored
Semicolon optional (but should be added) - separates individual statements
Curly braces to group blocks

Variables:

- var: global variable, become properties of window object
  - Use with caution, but ok to use
- let: local variable
- const: constant variable

Parameters:

- Primitives: pass by value
- Objects: pass by reference

Equality:

- == : weak equality, tries to use type conversion
- === : strong equality, must be the same type and same value

DOM API

Wednesday, February 22, 2023 5:00 PM

Document Object Model - Language neutral (python xml dom) API for manipulating markup languages like HTML

Change HTML and CSS programmatically: Dynamic HTML (DHTML)

DOM is the performance floor for web app, all frameworks eventually rely on DOM

document.getElementByID("id"); - returns first element with id
document.getElementsByClassName("class"); - returns collection of elements with class membership
   Array.from(…) - may be needed to iterate over collections using array logic
document.getElementsByTagName("tag"); - returns collection of all elements with name

document.querySelector("css selector"); - returns the first matching element
   document.querySelector("*") - returns the h1 element
document.querySelectorAll("css selector"); - returns a collection of all matching elements

docuement.body/head - returns body and head
element.getAttribute(attr) / element.setAttribute(attr, val) - gets and sets an attribute of the element
   OR element.attr = x; - maps the attribute to the object directly

Event Listeners

Friday, February 24, 2023 5:34 PM

element.addEventListener(event, callback) - adds a listener to the element that triggers on event and runs callback

element.removeEventListener(event, callback) - removes the callback handler from the element's event listener

AJAX and SPA

Monday, February 27, 2023 5:21 PM

AJAX: Asynchronous Javascript and XML - Use a request and response architecture to implement web app

- Send requests to server
- Server responds with data
- Minimize data sent, and avoid page redirects to improve responsiveness
- REST APIs: stateless API

Custom Elements

Friday, March 3, 2023 5:44 PM

Components: self-contained pieces of self-contained HTML, CSS and JS

Idea: We can create custom components using JS

```
class CustomElement extends HTMLElement {
   constructor() {
      Super();
   }
}

customElements.define("custom-element", CustomElement);
```

ShadowDOM: DOM tree containing the current element's inner HTML elements

Many ways to make components: balance runtime vs delivery time

CRUD, SSR vs SSG vs CSR

Wednesday, March 8, 2023 5:01 PM

C - Create
R - Read
U - Update
D - Delete

SSR: Server side rendering, run code on the server to render pages ie. Php
SSG: Static Site Generation, static sites ie html
CSR: Client side rendering, ie react

PWAs, Service Worker

Friday, March 10, 2023 5:32 PM

Progressive Web Apps:
- Website that can become an app through progressive enhancement
- Website "installs" the app in the background as the site is repeatedly visited
- Uses service workers to intercept and cache web requests
- Power of native application with web availability
    - Little performance loss, WASM and other technologies can be as fast as native
    - However, DOM access slow

Web App: application written in web technologies but can run natively
- Write app in web technologies, wrap app using a stripped down web browser
- Manifest file: JSON file describing meta data including:
    - Icon
    - URL to load
    - Splash screen
    - Orientation,
    - Menu colors

App Shell: Cached shell loads instantly on repeated visits, dynamic content loads as needed

PRPL: Push, render, pre-cache, lazy load

Service worker: Proxies the internet connection, intercepts saves and caches requests
- Cache only: only pull from cache
- Cache then network: Try the cache first and then go to network