Module 3 Day 1

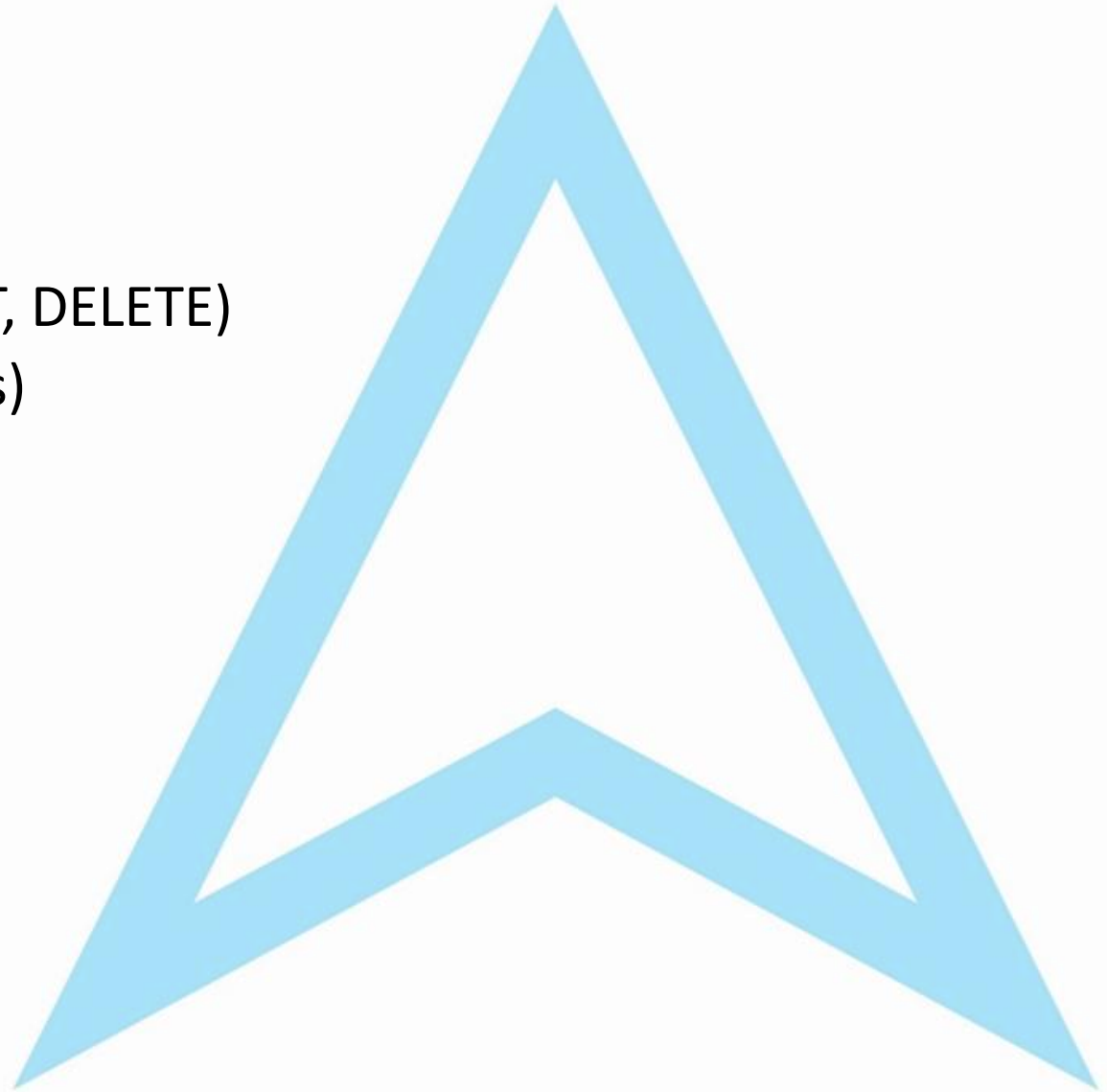
css

What makes an application?

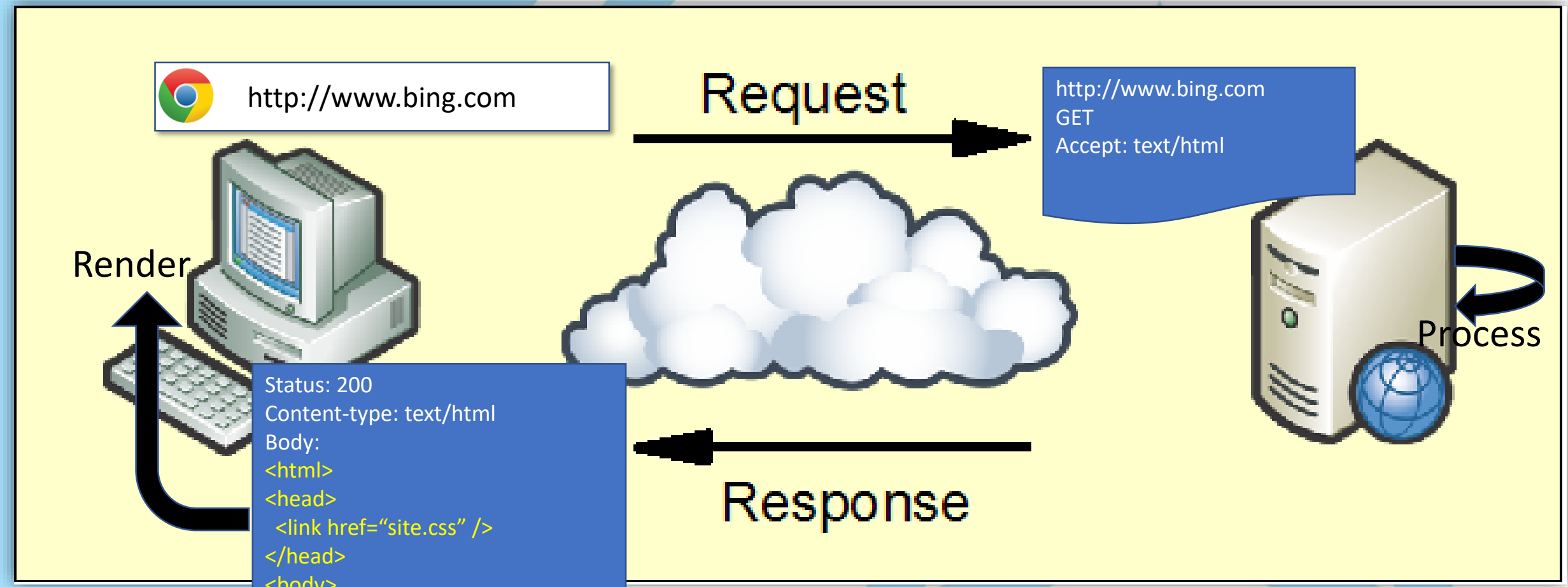
- Program Data
  - ✓ Variables & .NET Data Types
  - ✓ Arrays
  - ✓ More Collections (list, dictionary, stack, queue)
  - ✓ Classes and objects (OOP)
- Program Logic
  - ✓ Statements and expressions
  - ✓ Conditional logic (if)
  - ✓ Repeating logic (for, foreach, do, while)
  - ✓ Methods (functions / procedures)
  - ✓ Classes and objects (OOP)
  - ❑ Frameworks (MVC)
- Input / Output
  - User
    - ✓ Console read / write
    - ❖ HTML / CSS
    - ❑ Front-end frameworks (HTML / CSS / JavaScript)
  - Storage
    - ✓ File I/O
    - ✓ Relational database
    - ❑ APIs

HTTP Request-Response

- Request
  - URL – Uniform Resource Locator
  - HTTP Method / Verb (GET, PUT, POST, DELETE)
  - Headers (auth, content-type, cookies)
  - Body (sometimes)
- Response
  - Status (2xx, 3xx, 4xx, 5xx)
  - Headers (content-type, set-cookie)
  - Body (sometimes)

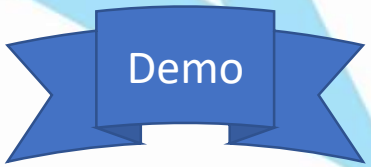


HTTP Request-Response



HTTP Request-Response

- Stateless
  - server “remembers” nothing about the client between requests
- Cyclic
  - Response from server contains references, links and redirects
  - Client responsible for making multiple requests to completely fulfill the user’s query and display a complete page
- Browser Developer Tools (F12)
  - Help you see all the requests that are taking place
  - Help understand performance issues



Semantic HTML

- Semantics == meaning
- Semantic elements – imply some meaning to parts of a page
  - main, article, section, aside, header, footer, nav
- Non-semantic elements
  - div, span, b, I
- Semantic HTML is preferred
- HTML applies *semantics*
- CSS applies *style*
- HTML5 Semantic Elements

Styles / Cascading Stylesheets (CSS)

- Settings that change the look of an element
- May be applied directly to an element using the *style* attribute
- May be added as an element in the HTML document *<style></style>*
- May be included in a stylesheet
  - A separate file linked to from the HTML doc
  - Usually *.css* file extension

CSS Rules - Properties

```
Selector(s) {
      property : value;
      property : value; …
}
```

- Property : value pairs
  - color : Red;
  - background-color : Yellow;
- https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Keyword_index

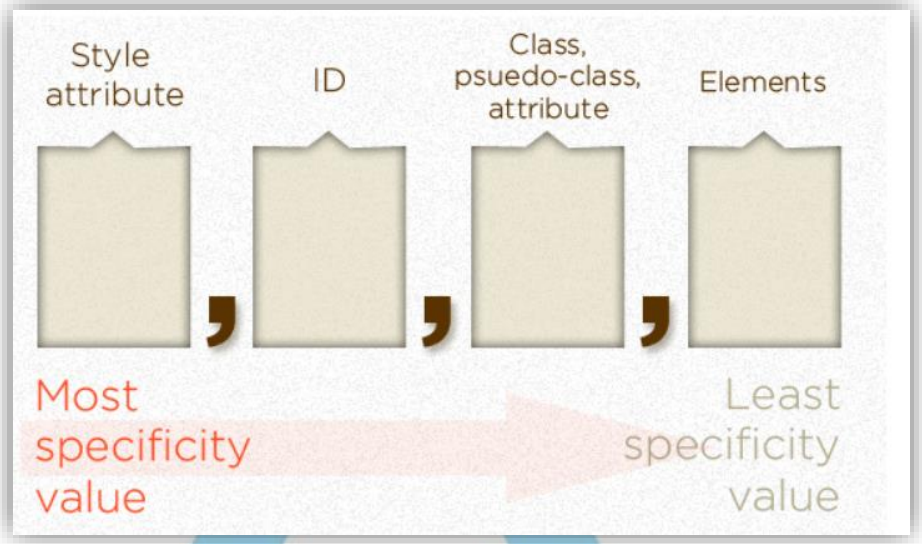
CSS Rules - Selectors

```
Selector(s) {
      property : value;
      property : value; …
}
```

- Type (element) selectors: p, body
- Class selectors: .className
- ID selectors: #mainDiv
- Multiple selectors: A, B
- Descendant: A B
- Child: A > B
- https://developer.mozilla.org/en-US/docs/Glossary/CSS_Selector

Specificity

- The most “specific” definitions apply
  - Inline styles trump stylesheets
  - Ids trump classes
  - Etc.
- https://www.w3schools.com/css/css_specificity.asp
- https://css-tricks.com/specifics-on-css-specificity/
- https://specificity.keegan.st/



Demo

Box Model

- Block Elements
  - div, h1-6, p, header, footer, section
  - Margin, padding, height effective
- Inline elements
  - span, a, img
  - Vertical settings have no effect
- Inline-block elements
  - select, button
- Change using the Display attribute

Demo

Layout

- Default
  - Left to right, top to bottom, in order they appear in the HTML document
- Relative
  - Relative to where it would otherwise be positioned in the normal flow
  - *top, right, bottom, and left*
- Absolute
  - Place the element relative to the parent ancestor
  - Elements are removed from the flow of the page.
  - Setting both *top* and *bottom*, or both *left* and *right*, you can "stretch" an element's dimensions.
- Fixed
  - Relative to the browser window
  - Does not scroll with the page.

Visual Studio Code (VS-Code)

- Open the *folder* “With Code”
- Install Live Server extension
- Running Live Server
- Updating code
- Stopping Live Server

CSS Variables (bonus ☺)

```css
:root {
    --gutter: 20px;
    --blue: #00adee;
    --green: #8cc36f;
    --bg-color: rgba(192, 192, 192, 0.1);
}

body {
    background-color: var(--bg-color);
}
```