|  |  |
| --- | --- |
| **Main Features** | HTML stands for *Hypertext Markup Language*. This represents both a language and a file that displays web pages on a client browser. The language tokens are processed by the client browser and sent to a server, which converts the processed tokens into a page with display elements. The page is sent back to the client as a response, and then rendered on the browser.  HTML has formatted strings of phrases or assigned variables. Examples include:   * h1 – h6 * button * div * form * a * img * audio * table * and others   These tags above non-semantic tags. There are also semantic tags which will be discussed below (see section titled **Semantics**).  Attributes can include:   * class * id * and others |
| **Formats** | <opening-tag *attributes=”attribute-values” properties*>Text</closing-tag> |
| **Rules** | 1. All HTML files begin an <html> tag. For HTML5 file, they begin with <!DOCTYPE html> tag, at the top of the page. 2. <html> is an HTML entity that displays a web page in HTML 3. HTML entity has <head> and <body> within 4. <head> establishes title, metadata, and relative links 5. <body> establishes page elements (numerous types) 6. open tag first, then close tag after |
| **Priorities** | Parent elements can have child elements  Child elements inherit properties from parent |
| **Design Benefits** | Design principles of HTML provide an organized hierarchical structure to represent web pages with parent elements, child elements within, attributes defined, and properties defined. |

**HTML**

**Semantics**

|  |  |
| --- | --- |
| **Main Features** | Semantics are a human-readable version of the HTML. It gives meaning to a web page, gives more information about the tags, and gives more hooks for styling of my content.  **THINK NEWSPAPER!!**  **You have:**   * Header * Nav * Section * Article * Footer   Attributes can include:   * address * time   More ***meaning*** less presentation |
| **Formats** | <opening-tag>Text</closing-tag> |
| **Rules** | 1. all tags are within html <body> 2. open tag first, close tag after 3. non-semantic tags can be placed inside semantic tags |
| **Priorities** | None |
| **Design Benefits** | Design principles of Semantic HTML provide a user with additional features from their webpage, using frontend elements that are organize and appeal to readers of webpages. The naming conventions of semantic html are easy to read by developers and users alike. Also, the pages are accessible through various devices, which provide dynamically adjusted display of content. The content is in accordance with each semantic tag. |

**CSS**

|  |  |
| --- | --- |
| **Main Features** | CSS stands for Cascading Style Sheets. This enables styling for language formats such as HTML and XML. On web pages, it customizes the displays such as fonts, colors, spacing, state transitions, etc. On an html page, a header section can contain a user-defined link to a stylesheet located elsewhere (either locally or remotely). |
| **Formats** | (Optional token [none] . # [href=’example-reference’]) [tag name, class, or id] {  attribute: attribute-value;  } |
| **Rules** | 1. Documents can be formatted in any way that the styler wants 2. Selectors pick elements and define attributes for them 3. Specificity **overrides styling** for based on id->class->tag 4. Two or more selectors can be chained (e.g., h1.special) 5. Descendant combinators select elements that are nested within other HTML elements, so the selector is more specific and the element appears “stylized” in the context we expect 6. Chaining and specificity and can be combined to further **override elements *beyond the general rules*** 7. One styling can be performed for multiple selectors (one-to-many) |
| **Priorities** | Priorities go to id value, then class value, then tag name |
| **Design Benefits** | Design principles of CSS encourage modularity, separation of duties, and maintainability of solution artifacts. It facilitates customization of HTML elements and saves a developer time. It is also device compatible, including syntax that encourage dynamic display between different types of devices (for example, laptops, tablets, smartphones, etc). Finally, CSS is purposed for decoration, as it renders a unique web page and user experience as intended by the developer. |

**Colors**

|  |  |
| --- | --- |
| **Main Features** | Colors have a full range of effects on web pages. They are used on backgrounds, text, images, animations, transition, user interactions, and many more aspects of a website. Within HTML, colors can be assigned for individual tags. Within CSS, colors can be assigned or defined or templated for one tag (coloring one element) or multiple tags (coloring multiple elements).  Pertaining to CSS specifically, the following colors can be applied to a page:   * Foreground/Background color * Hexadecimal varieties * Red-Green-Blue (RGB) varieties * Hue, Saturation, and Lightness (HSL) combinations * Opacity Features and Alpha Blends |
| **Formats** | Hexadecimal: Three (3) or Six (6) characters,  beginning with #, and  followed by sets of values 0 to 15 encoded as 0 through F  [Examples] 3 characters: #8A0 or #8a0 6 characters: #2F2F2F or #2f2f2f  RBG: Set of three values, each value ranging from 0 to 255  [Examples] rgb(0, 255, 192)  HSL: Hue is 0 to 360, Saturation is 0% to 100%, Lightness is 0% to 100%  [Examples] hsl(180, 60%, 80%)  Adding an Alpha value: attribute *a* is from 0 to 1.0  [Examples] rgba(40, 12, 200, 0.25)  hsla(38, 11%, 22%, 0.9)  **\*Note: All values end with a semi-colon (;)** |
| **Rules** | 1. Hex values need to written as one of [0123456789ABCDEF] 2. Each of the RGB, HSL, and Alpha values must abide by their range |
| **Priorities** | None |
| **Design Benefits** | Design principles of color make it an essential part of web design. The conventions are easy-to-follow and easy for developers to assign. There is room to be very creative with front-end colors, as one benefits from a full range of colors can be assigned. As a result, visualization of elements are easy to change, modify, and control. |

**Typography**

|  |  |
| --- | --- |
| **Main Features** | Typography is the art of arranging text – meaning *styling local fonts* or *text layouts* or *adding external fonts*. Local fonts can be styled from singular font or grouped as a font-family, with additional formatting features (such as weight, size, etc.). Text layouts can establish spacing, height, and alignment. External fonts can be retrieved sources on the web via a <link> placed in the HTML header section. |
| **Formats** | For styling local fonts, use conventional CSS formats  Example *attribute: attribute-value;*  For text layouts, use conventional CSS formats  Example *attribute: attribute-value;*  For external fonts, link in HTML as shown below:  HTML: add sets of <link> inside the <head> section  For filepath, assign @fontface ruleset as shown below:  @fontface {  font-family: ‘name’;  src: url(‘relative-filepath’) format (‘format-name’),  url(‘relative-filepath’) format (‘format-name’),  url(‘relative-filepath’) format (‘format-name’);  } |
| **Rules** | 1. All fonts are checked for browser compatibility 2. Styling local fonts follow typical CSS conventions 3. Text layouts also follow typical CSS conventions 4. CSS fontfaces should follow appropriate file formats (woff, woff2, otf, ttf) |
| **Priorities** | For font-family, check first font, then second font, then next font – if none of the fonts work, use browser default. |
| **Design Benefits** | Design principles of typography create fun and engaging interfaces. |