Kevin Chen

3/14/16

CSS Tutorial Notes

* Syntax: [selector]{[property]: [value]; [property]: [value] […]}
  + Example: h1{color:blue; font-size:12px;}
* Selectors:
  + Of a particular tag: tag name. Example: p
  + Of a particular id (one element): #[id name] Example: #para1
  + Of a particular class (multiple elements): .[class name] Example: .center
  + Can be combined: Example: p.center selects only paragraph elements with class center.
  + Grouping selectors: Selectors can be combined separated by commas
  + Comments: /\* followed by \*/
* External Style sheets: add <link rel="stylesheet" type="text/css" href="mystyle.css"> in the <head> element of the HTML page.
* Cascading order: what style will be used if there are multiple styles defined for a HTML element? This is the priority of style sheets (#1 has highest priority):
  + Inline style sheets
  + External and internal style sheets (if both styles are in this category, then last read style sheet will be used)
  + Browser default
* Backgrounds: Use these properties to modify the background
  + Color: background-color property
  + Image: background-image property
  + Repeat x and y: By default, image repeats horizontally (x) and vertically (y). Set background-repeat property to these values: repeat-x for repeating only horizontally, repeat-y for repeating only vertically, no-repeat for no repeat.
  + background-position property: set position of background (e.g. right top)
  + background-attachment: can be set to fixed so that it does not scroll with rest of page.
  + Shorthand: all values can be separated by spaces without property specified. Order of properties: color, image, repeat, attachment, position. It doesn’t matter if one of the properties is missing as long as the others are in order
* Borders: change style using border-style property. Value: border-width, border-style, border-color (separated by spaces)
  + border-width: can be set to specific size or by using thin/medium/thick. Can contain 1-4 values.
  + border-color. 1-4 values. If this is not set, it inherits color of the element
  + 4 values: top, right, bottom, left. 3 values: top, left and right, bottom. 2 values: top and bottom, left and right. 1 value: all 4
  + Individual border sides can be selected by border-[top/left/bottom/right]-[property type (e.g. style or color]. Example: border-top-style
  + Shorthand: border-width, border-style (required), border-color
* Margins (white space around elements)
  + Shorthand: margin-top, margin-right, margin-bottom, margin-left
  + Specify pixel value for each property. 1-4 values for margin property
  + Can be set to “auto” to center horizontally.
* Inheritance: Nested elements are parents of the elements surrounding them. Use the value “inherit” to inherit a certain property from the parent
* Padding (space between element content and element border)
  + Shorthand: padding-top, padding-right, padding-bottom, padding-left
  + Specify pixel value for each property. 1-4 values for padding property
* Width/Height: height/width properties.
  + max-width will specify the maximum width an element can be. Actual width will be smaller if the window is not long enough. max-width property overrides width.
* Text: color property, text-align property (center, left, right, justify – each line takes equal width), text-decoration property (none – often used to remove underlines from links, overline, line-through, etc.), text-transformation property (uppercase, lowercase, capitalize), text indentation (value is in pixels), letter spacing (value is space between characters), line height (space between lines), text direction, word spacing (spacing between words)
* Fonts: 3 classes: Serif (small lines at ends of some characters – e.g. Times New Roman), Sans-serif (no lines – e.g. Arial), monospace (all characters take same space)
  + Font-family property – contain multiple fonts separated by commas as a “fallback” system. Quotes around font names with spaces. Specify generic family at the end.
  + Font-style property (normal, italic, oblique)
  + Font-size (set with pixels or em. em is recommended – by default 1em = 16 px). Use combination of percent (as default font-size in the <body> element) and em is best: this will work across all browsers.
  + Font-weight (e.g. normal, bold) and font-variant (e.g. normal, small-caps)
* Links: Pointers: a:link for normal, unvisited links, a:visited for visited links, a:hover for links when user hovers mouse over it, a:active for links the moment it is clicked. a:hover must come after a:link and a:visited, and a:active must come after a:hover
  + text-decoration property to remove underlines from links
  + background-color property
  + Create link buttons by applying background-color, center text alignment, etc.
* Lists: list-style-type property for the type of list item marker (the bullet/number)
  + list-style-image to set image as list item marker
  + list-style-position to set whether list-item markers appear inside or outside of content flow
  + Shorthand: list-style-type, list-style-position, list-style-image
  + Add background-color to ol or li to change color of entire list. Add to ol li or ul li to affect only individual items.
* Tables: border property, border-collapse (set to collapse in table pointer to have single instead of double border)
  + Select table, th, td for all borders. Select table for just outside border.
  + width/height: set in px or %. 100% for width will span across entire page.
  + text-align (for horizontal text alignment) and vertical-align (for vertical).
  + padding property
  + Horizontal dividers: use the border-bottom property for th and td.
  + Hover: use the :hover selector for <tr>
  + nth-child(even) selects even (2nd, 4th, etc.) child
  + Responsive table: adds a horizontal scroll bar if table is too long. Add a container with the property: style="overflow-x:auto;"
* Box model: all HTML elements are like boxes. Outside to inside: margin, border, padding, content
* Outlines: a line drawn around elements to make them stand out.
  + Differ from borders: outlines are not part of the element’s dimensions
  + outline-style property, outline-color property, outline-width (in pixels or thick/thin/medium)
  + Shorthand: outline-width, outline-style, outline-color
* Display property: setting it to none will make it hidden: page will be displayed as if it were not there. Set it to block or inline to make it appear.
  + visibility:hidden will also make it disappear, but it takes up space still
* Position property: Position property is first set before the element is positioned using the left, right, bottom, top properties
  + Position: static: elements are not affected by top/bottom/left/right properties. Positioned according to normal flow of page.
  + Position: relative: elements positioned relative to normal position
  + Position: fixed: elements are relative to viewport (will not move when scrolled)
  + Position: absolute: elements positioned to nearest positioned ancestor.
  + z-index property specifies order in which elements appear when stacked. Larger stack order will appear in front of lower order.
* Float property: specifies where an element should float. Can float right or left.
  + Can be used to wrap text around images.
  + Clear property controls behavior of floating elements. Can be used to prevent elements flowing around floating elements by specifying on which sides of an element floating elements are not allowed to float.
  + If a floated element is taller than element that contains it, it will overflow. Set overflow property to auto to fix this.
* inline-block property for display property makes elements similar to inline elements, but they can have width and height.
* Alignment:
  + Use margin: auto to horizontally center an element.
  + Use position: absolute or float to align left or right.
* Combinators: something that explains the relationship between the selectors
  + Descendent selector: (Space) Example: div p: selects all <p> elements inside <div> elements
  + Child selector: > Selects all elements that are immediate children of an element.