**Name: Lauren Van Vlierbergen**

**ICE- 3- Concept Review**

1. Which of the following is a self-closing tag?
   1. img
   2. br
   3. h1
   4. src
   5. A and B
   6. None of the above
2. What are the three “parts” of an HTML document
   1. protocol,domain, document
   2. doctype, body, head,
   3. request, response, return
   4. browser, client, server
   5. None of the above

Head vs. Header? Which one is the required element?

1. Which of the following is an example of an absolute reference?
   * <http://www.cnn.com>
   * <http://www.cnn.com/menu.pdf>
   * <http://www.cnn.com/menu.html>
   * <menu.html>
   * [menu.pdf](menu.html)
   * A, B, & C
   * D and E
   * None of the above
2. Which line of code indicates to the browser that the page should be rendered using HTML5?
   * <!DOCTYPE html5>
     1. No-html 5 is simple and all you need to declare it is <!DOCTYPE html5>
   * <!DOCTYPE html>
     1. Yes - Doing so will cause even browsers that don't presently support HTML5 to enter into standards mode, which means that they'll interpret the long-established parts of HTML in an HTML5-compliant way while ignoring the new features of HTML5 they don't support. Much simpler than previous versions.
   * <html doctype = “html5”>
     1. No-There is no “html5” and plus this option is missing the important <!DOCTYPE>
   * All of the above
3. What are the three “parts” of a valid URL
4. protocol, domain, document
5. doctype, body, head,
6. request, response, return
7. browser, client, server
8. Which of the following is an example of relative reference?
   * <http://www.cnn.com>
   * <http://www.cnn.com/menu.pdf>
   * <http://www.cnn.com/menu.html>
   * <menu.html>
   * [menu.pdf](menu.html)
   * A, B, & C
   * D and E
   * None of the above
9. Which of the following is an example of the “cascading” in CSS.
   * The style attribute in a tag overrides the default styling of that element
   * The style attribute in a tag overrides styling from an external style sheet of that element
   * You can write rules that override Bootstrap styling
   * A and B
   * A, B, and C
   * None of the above
10. The form tags input = “text” and input = “email” look identical to the user. Which of the following is the best reason to use email instead of text:
    * To ensure code validation
    * To provide input validation on some browsers
    * So the browser knows to style the input field differently
    * All of the above
    * None of the above
11. In the following code snippet, what value is given for the bottom margin:

margin: 5px 10px 8px 6px;

1. 5px
2. 10px
3. 8px
   1. There are properties for setting the margin for each side of an element (top, right, bottom, and left)
4. 6px
5. None of the above
6. In the following code snippet, what value is given for the bottom margin:

margin: 5px 10px;

1. 5px
   1. 2 values, the top and bottom margins are set to the first value and the right and left margins are set to the second
2. 10px
3. 8px
4. 6px
5. None of the above
6. In the following code snippet, what value is given for the bottom margin:

margin: 10px;

1. 5px
2. 10px
   1. 1 value, all 4 sides have the same margin.
3. 8px
4. 6px
5. None of the above

**????????????**

**SHORT ANSWER:**

Consider the following code:

          <div class = "col-md-6 col-lg-2">

1. In an xs viewport \_\_\_\_\_ columns will be used to display the div.
2. In a sm viewport \_\_\_\_\_ columns will be used to display the div.
3. In a md viewport \_\_\_\_\_ columns will be used to display the div.
4. In a lg viewport **2** columns will be used to display the div.

* All columns need to add up to 12. A col-lg-2 will make 2 equal columns (6 and 6 = 12). This code snippet represents 1 row.

1. What is a reason for viewing your site on a local server?

* **Keep files secure on local server**
  + If something needs to be changed in your code, you can fix it before the public tries to access your site
* **Testing purposes**
  + Allows you to preview and test various aspects of your website before hosting it

1. What are the four tenets of Accessibility?
   * Perceivable
   * Operable
   * Understandable
   * Robust
2. What two attributes should you always include with an img tag?
   * alt
   * src

**Explanation:**

**SRC.**The only attribute you need to get an image to display on a web page is the SRC attribute. This attribute identifies the name and location of the image file to be displayed.

**ALT**To write valid XHTML and HTML4, the ALT attribute is also required. This attribute is used to provide nonvisual browsers with text that describes the image. Browsers display the alternative text in different ways. Some display it as a pop-up when you put your mouse over the image, others display it in properties when you right-click on the image, and some don't display it at all

Give an example of a:

1. pseudo-element
   * ::after
   * ::before

**Explanation:**

A CSS **pseudo-element** is a keyword added to a selector that lets you style a specific part of the selected element(s). For example, [::first-line](https://developer.mozilla.org/en-US/docs/Web/CSS/::first-line) can be used to change the font of the first line of a paragraph. This is a CSS Selector

1. pseudo-class
   * :focus
   * :hover

**Explanation:**

A CSS **pseudo-class** is a keyword added to a selector that specifies a special state of the selected element(s). For example, [:hover](https://developer.mozilla.org/en-US/docs/Web/CSS/:hover) can be used to change a button's color when the user hovers over it.

1. static measurement
   * position: static;
   * <div> all elements are static by default

**Explanation:**

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

**fluid measurement** - a **layout** that uses proportions to measure blocks of content, images, or any other item. This allows the web page to stretch and contract relative to a screen size.

**ARIA label –**

* Aria-describedby = “ ”;
* Role = “ ”

**Explanation:**

Web accessability Initiative. WAI-ARIA is an incredibly powerful technology that allows developers to easily describe the purpose, state and other functionality of visually rich user interfaces - in a way that can be understood by Assistive Technology. WAI-ARIA has finally been integrated into the current working draft of the HTML 5 specification.

1. What attribute can you set if you want to give the user keyboard access to it, e.g. it is reachable via tabbing?

tabindex = “1”

tabindex = “2”

?

1. What is the difference visually on a page between using **display: none** vs **visibility: hidden**?

The elements are both visually hidden but with visibility:hidden means space is still allocated for the tag on the page

display:none means that the tag in question will not appear on the page at all (although you can still interact with it through the dom). There will be no space allocated for it between the other tags.

visibility:hidden means that unlike display:none, the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn't seen on the page.

1. What does the term “mobile first” mean?

It’s a design term used in UX that focuses on designing an online experience for mobile before designing it for the desktop. Starting on a smaller screen and then scaling up is a better approach then vise versa. Today, many consumers surf the web with their mobile phones so it’s important for companies to have a user-friendly mobile experience with their companies website.

1. What is a default style sheet and what is its purpose?

Default style sheet: a style sheet used in HTML 4 and CSS 2 that expresses generally accepted default style information for each element.

1. Name two browser prefixes
   * -webkit-
   * -moz-

Explanation:

The CSS browser prefixes that you can use (each of which is specific to a different browser) are:

* Android: -webkit-
* Chrome: -webkit-
* Firefox: -moz-
* Internet Explorer: -ms-
* iOS: -webkit-
* Opera: -o-
* Safari: -webkit-

1. Which CSS3 property determines the stacking order of elements if they are placed in the same spot using position: absolute?

* Z-index

Explanation:

An absolutely positioned element no longer exists in the normal document layout flow. Instead, it sits on its own layer separate from everything else. This is very useful — it means that we can create isolated UI features that don't interfere with the position of other elements on the page. Second, notice that the position of the element has changed — this is because [top](https://developer.mozilla.org/en-US/docs/Web/CSS/top), [bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/bottom), [left](https://developer.mozilla.org/en-US/docs/Web/CSS/left), and [right](https://developer.mozilla.org/en-US/docs/Web/CSS/right) behave in a different way with absolute positioning. Instead of specifying the direction the element should move in, they specify the distance the element should be from each containing element's sides. So in this case, we are saying that the absolutely positioned element should sit 30px from the top of the "containing element", and 30px from the left.

Yes, you can, by using the [z-index](https://developer.mozilla.org/en-US/docs/Web/CSS/z-index) property. "z-index" is a reference to the z-axis. You may recall from previous points in the source where we discussed web pages using horizontal (x-axis) and vertical (y-axis) coordinates to work out positioning for things like background images and drop shadow offsets. (0,0) is at the top left of the page (or element), and the x- and y-axes run across to the right and down the page (for left to right languages, anyway.)

Web pages also have a z-axis — an imaginary line that runs from the surface of your screen, towards your face (or whatever else you like to have in front of the screen). [z-index](https://developer.mozilla.org/en-US/docs/Web/CSS/z-index) values affect where positioned elements sit on that axis — positive values move them higher up the stack, and negative values move them lower down the stack. By default, positioned elements all have a z-index of auto, which is effectively 0.

1. Name a **position** value that doesn’t utilize any offsets.

* static

**Explanation:**

**Elements** by default **have** the position value of **static**, meaning they don't **have**, nor **will** they accept, any specific box **offset** properties as in (left right top and bottom values) so they **will** just be positioned as they normally should

o **offset** an element it's position has to be position:relative

the co-ordinates, top, right, bottom and left serve different purposes depending on if the element is relatively or absolutely positioned.

**When is an element offset as opposed to moved?**

when you actually offset using position: relative; the element is not removed from the flow, and indeed the space that the element would have taken up if it had remained static (the default) is still reserved for it, therefore you have just offset it from it's original position. Any element following it will appear where it would have done even if you hadn't offset it's predecessor - like this [example](http://jsfiddle.net/clairesuzy/D7p3k/)

**Moving, not offsetting**

If however you actually want to move an element, then it needs to be removed from the flow, so there is no space reserved for it, and then that's when you use position:absolute; or fixed.. that is the difference

**Summary**

* static is the default, and you just use margins to move it around, it will ignore co-ordinates and z-index
* relative is reserved space, co-ordinates will offset it from it's original space
* absolute will remove the element from the flow and the co-ordinates will be calculated according to it's [containing block](http://www.w3.org/TR/CSS2/visudet.html#containing-block-details), which is the nearest relatively positioned ancestor (or the body element if no relatively positioned ancestors exist)
* fixed does not have a containing block, i.e. you can't specify which element it should be positioned in relation to, it will just fix itself in relation to the viewport

and finally an element will not accept a z-index if it's position is the default of static, so position: relative; without any co-ordinates applied is similar to static, but it is useful for z-indexing and being a containing block for absolutely positioned elements

<https://stackoverflow.com/questions/5602124/css-offset-properties-and-static-position>

### Values

**static**

The element is positioned according to the normal flow of the document. The [top](https://developer.mozilla.org/en-US/docs/Web/CSS/top), [right](https://developer.mozilla.org/en-US/docs/Web/CSS/right), [bottom](https://developer.mozilla.org/en-US/docs/Web/CSS/bottom), [left](https://developer.mozilla.org/en-US/docs/Web/CSS/left), and [z-index](https://developer.mozilla.org/en-US/docs/Web/CSS/z-index) properties have no effect. This is the default value.

**relative**

The element is positioned according to the normal flow of the document, and then offset relative to itself based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were static. This value creates a new [stacking context](https://developer.mozilla.org/en/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context) when the value of z-index is not auto. The effect of relativeon table-\*-group, table-row, table-column, table-cell, and table-captionelements is undefined.

**absolute**

The element is removed from the normal document flow; no space is created for the element in the page layout. Instead, it is positioned relative to its closest positioned ancestor if any; otherwise, it is placed relative to the initial [containing block](https://developer.mozilla.org/en-US/docs/Web/CSS/All_About_The_Containing_Block). Its final position is determined by the values of top, right, bottom, and left. This value creates a new [stacking context](https://developer.mozilla.org/en/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context) when the value of z-index is not auto. Absolutely positioned boxes can have margins, and they do not collapse with any other margins.

**fixed**

The element is removed from the normal document flow; no space is created for the element in the page layout. Instead, it is positioned relative to the screen's viewport and doesn't move when scrolled. Its final position is determined by the values of top, right, bottom, and left. This value always creates a new [stacking context](https://developer.mozilla.org/en/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context). When an ancestor has the transform, perspective, or filter property set to something other than none, that ancestor is used as the container instead of the viewport (see [CSS Transforms Spec](https://www.w3.org/TR/css-transforms-1/#propdef-transform)). In printed documents, the element is placed in the same position on every page.

**sticky**

The element is positioned according to the normal flow of the document, and then offset relative to its flow root and [*containing block*](https://developer.mozilla.org/en-US/docs/Web/CSS/All_About_The_Containing_Block), including table-related elements, based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements. This value always creates a new [stacking context](https://developer.mozilla.org/en/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context). Note that sticky, by specification, will not work inside element with overflow: hidden or auto. (ref: [Github issue on W3C CSSWG](https://github.com/w3c/csswg-drafts/issues/865))

1. Write a code snippet to make a list that lists its items with square.

ul {

list-style-type: square;

}

?

1. Consider the code snippet below that uses grid positioning.

**.two {**

**grid-column: 2 / 4;**

**grid-row: 1 / 3;**

**}**

How many columns does it span? 2 columns

How many rows does it span? 2 rows

Explanation:

<https://developer.mozilla.org/en-US/docs/Web/CSS/grid-area>

We have quite a lot of code here to position each item. It should come as no surprise to know there is a [shorthand](https://developer.mozilla.org/en-US/docs/Glossary/shorthand_properties). The [grid-column-start](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-column-start) and [grid-column-end](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-column-end) properties can be combined into [grid-column](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-column), [grid-row-start](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-row-start) and [grid-row-end](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-row-end) into [grid-row](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-row).

\*\* review this on the website because I’m a little confused about grids

1. Write a snippet of code that causes all divs with a class of **blue** to use white font on a blue background when the div is “clicked” with the mouse.

.blue a:active {

font-color: white;

background-color: blue;

}

????

1. Write a snippet of code that sets the background-image of the header to the file maizeAndBlue.jpg. You should assume that the css file is in the css subfolder and that the image is in the img folder.

header {

background-image: url(‘file:///Users/laurenvanvlierbergen/Desktop/ css/img/maizeAndBlue.jpg’)

}

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

See the html and css files on the next page. What is the color, background-color, and font-family for each of the paragraphs below? Assume all fonts ***except Courier*** are supported by your browser. (Please contact me if you are not able to view the image.)

Color Background-color Font

1. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. B \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. C \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. E \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. F \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. G \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. H \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>CSS Preferences</title>
 <link rel="stylesheet" type="text/css" href="ReviewQuestions.css">
</head>
<body>
 <p id = "p1" class = "mon">
  A
 </p>
 <p id = "p2" class = "mon tue">
  B
 </p>
 <p id = "p3" class = "tue mon">
  C
 </p>
 <p id = "p4">
  D
 </p>
 <p id = "p5" class = "thur">
  E
 </p>
 <p id = "p6" class = "fri">
  F
 </p>
 <p class = "thur mon">
  G
 </p>
 <p id = "p7" class = "mon" 
    style="background-color:green">
  H
 </p>
</body>
</html>**

#p1, #p2, #p3{
 width:55px;
 line-height: 50px;
 text-align: center;
}

#p4, #p5, #p6, #p7{
 width:55px;
 line-height: 50px;
 text-align: center;
 background: lightgrey;
}

.mon{
 background-color: pink;
 color: green;
 font-family: Times, Helvetica;
}

.tue{
 font-family: Courier, Cursive; 
 color: white;
 background-color: black;
}

.thur{
 background-color: blue;
 font-family: Arial;
}
