

CSS Skill Questions

1). Among these selectors which selector has the highest specificity ranking for selecting the anchor link element?

ul li a

a

.example a

div a

A) .example a

B) div a

C) a

D) ul li a

➤ Suppose there are four <a> element selectors like given above having different style attributes and values. For eg:- ul li a {background-color: 'red'}, a {background-color: 'green'}, etc. Now which style is supposed to be applied on the <a> HTML element is determined by the browser on the basis of a term called 'specificity'.

➤ Specificity is a calculated algorithmic value based upon the specificity score of each element inside selector. All the selectors listed above:- 'ul li a', '.example a', 'div a', etc. have a specificity score. Browser calculates the specificity score of each selector and the selector getting highest score has its properties getting applied on <a> HTML element.

➤ Specificity score of a selector is the sum of the selector weight of each of the element present inside the selector. There are basically 4 selector weight categories:- Inline styles, ID, CLASS & TYPE. If the element inside the selector group is an ID, it gets highest weight and if it is a TYPE, it gets lowest weight. However, if there is any inline style in the targeted HTML element, it will override every specificity discussed above. Consider the ID element getting a score of 100, CLASS getting 10 and TYPE getting 1. Inline styles get a score of 1000.

➤ Going with this logic, 'ul li a' will get a score of $1 + 1 + 1 = 3$. All the elements inside this selector ('ul', 'li' & 'a') are TYPE and hence each get a score of 1. '.example a' will get a score of $10 + 1 = 11$ as '.example' is a CLASS and 'a' is a TYPE. Similarly, 'div a' will have score of 2 and 'a' will have score of 1. This way, the selector in option B has the highest specificity ranking.

➤ '!important' will override even inline styles. It is an exception.

2). Using an attribute selector, how would you select an <a> element with a "title" attribute?

A) a[title]{...}

- B) a > title {...}
- C) a.title {...}
- D) a=title {...}

3). What is the CSS selector for an <a> tag containing the 'title' attribute?

- A) a[title]
- B) a > title
- C) a=title
- D) a.title

4). CSS grid and flexbox are now becoming a more popular way to create page layouts. However, floats are still commonly used, especially when working with an older code base, or if you need to support older browser version. What are two valid techniques used to clear floats?

- A) Use the "clearfix hack" on the floated element and add a float to the parent element.
- B) Use the overflow property on the floated element or the "clearfix hack" on either the floated or parent element.
- C) Use the "clearfix hack" on the floated element or the overflow property on the parent element.

D) Use the "clearfix hack" on the parent element or use the overflow property with a value other than "visible."

- In CSS, we can push an HTML element to right, left, or any other position with the help of 'float' property. In the file [clear.html](#), we can see that class 'div1' has a style property 'float: left' which pushes the element 'div1' on the web page to left hand side.
- But, the floating of 'div1' will create a vacant space on a better part of the region. Hence, the next element 'div2' will come up and fill the vacant space placing itself right beside the 'div1' block. This is the only reason we see both 'div1' and 'div2' on the same line.
- This breaks the styling as we never wanted both the blocks to appear on the same line. For the mitigation of this issue, we use another property called 'clear'. As we can see in the file, 'div3' class has same styling properties and values as 'div1' class. Just like 'div2' being next to 'div1', 'div4' is next to 'div3' as per code but it is still residing on the next line of 'div3' in the web.
- This magic happened because of 'clear: left' applied on the 'div4' block. This styling property instructs the HTML element to place itself right below the element having 'float:

left' property. Hence, 'div4' is placed right below the 'div3' block rather than beside it on the same line like 'div2' placed with 'div1'.

- In the file [clearfix.html](#), under the 'Without Clearfix', the image is floated to left but the image is taller than the element containing it so it overflows outside which can be very well seen on the web.
- In the same file, the 'With Clearfix' text appears right below the image and not beside it because of 'clear: right' style applied on the element. The image below the 'With Clearfix' is contained within the element containing it and is not overflowed because of a property 'overflow: auto' wrapped inside the class 'clearfix' which is in turn applied on the parent element of the image. This CSS fixing can be called as clearfix. Clearfix is not a CSS property which can be applied on some element.
- The style property 'overflow: visible' will highlight the overflow of the image. Hence, other values of 'overflow' such as 'scroll', 'hidden' and 'auto' can be used.

5). What element(s) do the following selectors match to?

5.1) `.nav { ...; }`

5.2) `nav { ...; }`

5.3) `#nav { ...; }`

A)

A.1). An element with an ID of "nav"

A.2). A nav element

A.3). An element with a class of "nav"

B) They all target the same nav element.

C)

C.1). An element with an class of "nav"

C.2). A nav element

C.3). An element with a id of "nav"

D)

D.1). An element with an class of "nav"

D.2). A nav element

D.3). An div with a id of "nav"

6). When adding transparency styles, what is the difference between using the opacity property versus the background property with an rgba() value?

A) Opacity specifies the level of transparency of the child elements. Background with an rgba() value applies transparency to the background color only.

B) Opacity applies transparency to the background color only. Background with an rgba() value specifies the level of transparency of an element, as a whole, including its content.

C) Opacity specifies the level of transparency of an element, including its content. Background with an rgba() value applies transparency to the background color only.

D) Opacity applies transparency to the parent and child elements. Background with an rgba() value specifies the level of transparency of the parent element only.

- Opacity is a CSS styling property which specifies the level of transparency, whereas rgba() is an inbuilt function of CSS which applies the color as well as transparency on an HTML element. Because 'opacity' is a property and rgba() is a function, opacity specifies the level and rgba applies the transparency and it is not the other way around as shown in options B and D.
- The value of 'opacity' property can range from 0.1 to 1.0. Lower the value, more is the transparency.
- In the [opacity.html](#) file, we can clearly see the difference between 'opacity' property and 'rgba()' function induced transparency. In 'opacity' induced transparency, the text or the content inside the element also gets affected by the transparency whereas in rgba() induced transparency, the text or the content inside the element remains unaffected.

7). Which statement about block and inline elements is true?

A) By default, block elements are the same height and width as the content container between their tags; inline elements span the entire width of its container.

B) By default, block elements span the entire width of its container; inline elements are the same height and width as the content contained between their tags.

C) A <nav> element is an example of an inline element. <header> is an example of a block element.

D) A is an example of a block element. <div> is an example of an inline element.

- Block elements in HTML always start with a new line. They take up entire available width of the container containing them. Two block elements always have a default margin space between them which browsers add.
- <p> and <div> are commonly used block level elements. In the file *block&inline.html*, we can clearly see the distinction between both p and div elements. Both these are block level elements which start from new line and take up entire width of the screen or container.
- In this file, there is a inside the div element which is an example of inline element. As we can see, span element has height as that of div element. This inline element does not start from new line with default margin.

8). CSS grid introduced a new length unit, fr, to create flexible grid tracks. Referring to the code sample below, what will the widths of the three columns be?

```
.grid {
    display: grid;
    width: 500px;
    grid-template-columns: 50px 1fr 2fr;
}
```

- A) The first column will have a width of 50px. The second column will be 50px wide and the third column will be 100px wide.
 - B) The first column will have a width of 50px. The second column will be 150px wide and the third column will be 300px wide.
 - C) The first column will have a width of 50px. The second column will be 300px wide and the third column will be 150px wide.
 - D) The first column will have a width of 50px. The second column will be 500px wide and the third column will be 1000px wide.
- The file *grid.html* shows the demonstration of a grid layout. There are two most important aspects of a grid layout without which a grid structure cannot manifest.
 - First one is that the parent element of the grid structure which is the element having class 'grid-container' in this case must be having a property 'display: grid'. Secondly, it must be having a property called 'grid-template-columns'.
 - We can see in the file that the value for property 'grid-template-columns' in 'grid-container' class is 'auto auto auto' which indicates three columns because 'auto' is written three times and each column's width is automatically adjusted automatically because of 'auto' value.

- In the given CSS code sample, the total width of grid is 500px. This grid will have three columns because three items are written in value against 'grid-template-columns' property.
- The first column's width is 50px which is fixed. '1fr' means one fractional unit or one part of the available space. So by calculation, we can see that total grid width is 500px and first column's width is 50px implying there are total of $500 - 50 = 450$ px remaining for two columns.
- Second column has 1fr (1 part) and third column has 2fr (2 parts) totalling to 3 parts divided between remaining 450px. So $450\text{px} / 3 = 150\text{px}$ per part. This means 1 fr = 150px. Hence, the width of all the columns would be 50px, 150px ($1 * 150 = 150$), 300px ($2 * 150 = 300$).

9). If the width of the container is 500 pixels, what would the width of the three columns be in this layout?

`.grid { display: grid; grid-template-columns: 50px 1fr 2fr; }`

A) 50px, 150px, 300px

B) 50px, 200px, 300px

C) 50px, 100px, 200px

D) 50px, 50px, 100px

10). What is the line-height property primarily used for?

A) to control the height of the space between two lines of content

B) to control the height of the space between heading elements

C) to control the height of the character size

D) to control the width of the space between characters

- By adding property 'line-height' to the parent element of the text node, we can control the spacing between two lines of the text content.
- The value of this style property can be 'normal', or any number such as '1.6', '2.0', etc. Or any number in pixel units such as '5px', '7px', etc. Or even percentage like '200%'.

11). Three of these choices are true about class selectors. Which is NOT true?

A) Multiple classes can be used within the same element.

- B) The same class can be used multiple times per page.
- C) Class selectors begin with a leading period
- D) Classes can be used multiple times per page but not within the same element.

12). What is not true about class selectors?

- A) Only one class value can be assigned to an element.
- B) An element can have multiple class value.
- C) Class selectors are marked with a leading period.
- D) More than one element can have the same class value.

13). There are many properties that can be used to align elements and create page layouts such as float, position, flexbox and grid. Of these four properties, which one should be used to align a global navigation bar which stays fixed at the top of the page?

- A) position
- B) flexbox
- C) grid
- D) float

- In CSS, any element can be positioned to any endpoint using 'position' property. Based upon the value of this property, the element can be either fixed, sticky, absolute, relative, etc. The file [position.html](#) shows the impact of different values of the 'position' property on the element.
- The possible values of 'position' property are 'static', 'relative', 'fixed', 'absolute' & 'sticky'. The file shows the impression each of these values create on an element.
- If we apply 'position: static' on any element, the properties like 'left', 'right', 'top' & 'bottom' do not apply on the element. This is the default value and the standing of an element even if we do not declare this property on the element at all. 'position: static' property goes with the flow of the document and places itself accordingly. In the file, #div-1 element has a class 'box' which in turn has a property 'left: 600px'. Because the 'position' here is 'static', the element has not moved in left direction despite inheriting from 'box' class.
- The element with the property 'position: relative' takes the properties like 'left', 'right', 'top' & 'bottom' and moves accordingly. For this reason, the #div-2 element in the web has moved left by 600px.

- The element with the property 'position: fixed' can be adjusted by the left, right, etc. properties. This property makes the element fixed on the page and even if we scroll the page down, the element does not go off of the screen. Hence, if we want a global navigation bar which stays on top even after scroll down, we can use this property.
- The element with the property 'position: absolute' finds its closest ancestor with 'position: relative' property and positions itself according to that. In the file we can see that #div-4 is the immediate parent of #div-5 element. Hence, whatever value we apply for the property 'left', 'right', 'top' or 'bottom' upon the #div-5 element is adjusted as per the #div-4 element. For eg: - the 'top: 70px' on #div-5 means 70px beneath the top of #div-4 element and not the document body. If #div-5 does not find parent with 'position: relative', it takes document body as the reference for positioning.
- There is another property 'position: sticky' which can toggle between 'position: fixed' and 'position: relative' based upon the scroll position. We can see that in #div-6 element on the web. For this property to work, it is necessary to add another property on the same element which is 'top'. The 'top' property will position the element from top which will indicate that the element will have 'position: relative' till the value of 'top' in the scroll. As soon as scroll moves past that, the property will change to 'position: fixed'.

14). In the shorthand example below, which individual background properties are represented?

background: blue url(image.jpg) no-repeat scroll 0px 0px;

A)

background-color: blue;
background-image: url(image.jpg);
background-repeat: no-repeat;
background-attachment: scroll;
background-position: 0px 0px;

B)

background-color: blue;
background-img: url(image.jpg);
background-position: no-repeat;
background-scroll: scroll;
background-size: 0px 0px;

C)

background-color: blue;
background-src: url(image.jpg);
background-repeat: no-repeat;

background-wrap: scroll;

background-position: 0px 0px;

D)

background-color: blue;

background-src: url(image.jpg);

background-repeat: no-repeat;

background-scroll: scroll;

background-position: 0px 0px;

- The 'background' property is the shorthand of 'background-color' showing the color, 'background-image' showing image, 'background-position' showing position, 'background-size' showing size, 'background-repeat' showing repeat, 'background-origin' showing positional area, 'background-clip' painting area of the background image & 'background-attachment' showing whether the background image will be scrolled or fixed with the rest of the page.
- 'background-position' will show position taking the values like 'left center', 'left bottom', 'right top', 'right center', etc. 'background-size' will take values like 'auto', 'cover', '50%', '30px', etc.
- 'background-repeat' will indicate how will the background images will repeat. For eg: 'repeat-x' will repeat the image in x-direction and 'repeat-y' in y-direction. 'no-repeat' will not repeat the image.
- 'background-origin' will show from where will the background image start. For eg: the value of this property 'padding-box' will start the image from upper left corner of the padding edge whereas the value 'border-box' will start the image from the upper left corner of the border edge and the value 'content' will start the image from upper left corner of the inside content.
- The 'background-clip' will indicate how much the background image will extend beyond the container. For eg: if the value of this property is 'content-box', the image will not extend beyond the content border of the container. If the value is 'padding box', the image will extend beyond the content border but not beyond the padding border. Similarly, the value 'border-box' indicates that the image will go beyond the content and padding borders but not the actual container border.

15). In this example, according to cascading and specificity rules, what color will the link be?

```
.example { color: yellow; }
```

```
ul li a { color: blue; }
```

```
ul a { color: green; }
```

```
a { color: red; }
```

```
<ul>
```

```
<li><a href="#" class="example">link</a></li>
```

```
<li>list item</li>
```

```
<li>list item</li>
```

```
</ul>
```

A) green

B) yellow

C) blue

D) red

16). When elements overlap, they are ordered on the z-axis (i.e., which element covers another). The z-index property can be used to specify the z-order of overlapping elements. Which set of statements about the z-index property are true?

A). Larger z-index values appear on top of elements with a lower z-index value. Negative and positive numbers can be used. z-index can only be used on positioned elements.

B) Smaller z-index values appear on top of elements with a larger z-index value. Negative and positive numbers can be used. z-index must also be used with positioned elements.

C) Larger z-index values appear on top of elements with a lower z-index value. Only positive numbers can be used. z-index must also be used with positioned elements.

D) Smaller z-index values appear on top of elements with a larger z-index value. Negative and positive numbers can be used. z-index can be used with or without positioned elements.

- 'z-index' is the property based on which, elements can stack on one another. Larger the value, more the top position element acquires. 'z-index' entertains both positive and negative numbers because the default 'z-index' value of every element is -1.
- An important note here is that the property 'z-index' can work only if along with this property, 'position' property is applied. It is more preferable if the element has properties 'position: absolute' and 'z-index' both inside itself, but still the 'position' property declaration along with its value is necessary.

- If two HTML elements overlap without a z-index, the element positioned last as per the code will be shown on the top.
- If we have a an element and its child and we want the child to be shown on top of the parent element, it is simply not possible. If suppose there are two sibling components and we want to give different z-index values to each of them, then the parent element must have property 'position: relative'.
- The elevation from z-index happens due to comparision of sibling elements. The elevation level does not hold up outside the parent element. For eg: if there are two sibling components having z-index values 1 and 2 respectively, outside the parent element if we put another element having z-index = 3, the element will not overlap the sibling elements having z-index values 1 or 2.

17). When elements within a container overlap, the z-index property can be used to indicate how those items are stacked on top of each other. Which set of statements is true?

A)

- A.1). Larger z-index values appear on top elements with a lower z-index value.
- A.2). Negative and positive number can be used.
- A.3). z-index can be used only on positioned elements.

B)

- B.1). Smaller z-index values appear on top of elements with a larger z-index value.
- B.2). Negative and positive numbers can be used.
- B.3). z-index can be used with or without positioned elements.

C)

- C.1). Smaller z-index values appear on top of elements with a larger z-index value.
- C.2). Negative and positive number can be used.
- C.3). z-index must also be used with positioned elements.

D)

- D.1). Larger z-index values appear on top of elements with a lower z-index value.
- D.2). Only positive number can be used.
- D.3). z-index must also be used with positioned elements.

18). What is the difference between the following line-height settings?

line-height: 20px;

line-height: 2;

A) The value of 20px will set the line-height to 20px. The value of 2 will set the line-height to twice the size of the corresponding font-size value.

B) The value of 20px will set the line-height to 20px. The value of 2 is not valid.

C) The value of 20px will set the line-height to 20px. The value of 2 will default to a value of 2px.

D) The value of 20px will set the line-height to 20px. The value of 2 will set the line-height to 20% of the corresponding font-size value.

19). In the following example, what color will paragraph one and paragraph two be? (Alternative: In this example, what color will paragraphs one and two be?)

```
<section>
```

```
    <p>paragraph one</p>
```

```
</section>
```

```
<p>paragraph two</p>
```

```
section p { color: red; }
```

```
section + p { color: blue; }
```

A) Paragraph one will be blue, paragraph two will be red.

B) Both paragraphs will be blue.

C) Paragraphs one will be red, paragraph two will be blue.

D) Both paragraphs will be red.

- The link https://www.w3schools.com/cssref/css_selectors.asp gives full reference of all the types of CSS class selectors.
- 'section p { color: red; }' CSS selector picks up the <p> element which is a descendant of <section> element. Even if <p> is grand child or great grand child of <section> element, still this style would have applied on it because for this style of apply on 'paragraph 1', <p> must be a descendant of <section> element.

- 'section + p { color: blue }' CSS selector picks up the first <p> element placed immediately after the <section> element. So in case there is 'paragraph 3' text wrapped inside <p> element placed immediately after 'paragraph 2', this style would not apply on 'paragraph 3'.
- If in case the second <p> element wrapping 'paragraph 2' is in turn wrapped inside a <div> element, this style would not apply on it and 'paragraph 2' would never be blue in color.

20). What are three valid ways of adding CSS to an HTML page?

A)

A.1). External; CSS is written in a separate file.

A.2). Inline; CSS is added to the <head> of the HTML page.

A.3). Internal; CSS is included within the HTML tags.

B)

B.1). External; CSS is written in a separate file and is linked within the <header> element of the HTML file.

B.2). Inline; CSS is added to the HTML tag.

B.3). Internal; CSS is included within the <header> element of the HTML file.

C)

C.1). External; CSS is written in a separate file and is linked within the <head> element of the HTML file.

C.2). Internal; CSS is included within the <header> element of the HTML file.

C.3). Inline; CSS is added to the HTML tag.

D)

D.1). External; CSS is written in a separate file and is linked within the <head> element of the HTML file.

D.2). Inline; CSS is added to the HTML tag.

D.3). Internal; CSS is included within the <head> element of the HTML file.

21). Which of the following is true of the SVG image format? (Alternative: Which statement about the SVG image format is true?)

A) CSS can be applied to SVGs but JavaScript cannot be.

B) SVGs work best for creating 3D graphics.

C) SVGs can be created as a vector graphic or coded using SVG specific elements such as <svg>, <line>, and <ellipse>.

D) SVGs are a HAML-based markup language for creating vector graphics.

- SVG images are pure XML based. Using SVGs, we can create vector graphics using XML format.
- SVGs can also be coded using SVG specific elements like <svg>. For eg: for creating a circle, inside <svg> element, we write <circle> elements with all the attributes like 'cx', 'cy', 'r', 'stroke', etc.
- SVG stands for Scalable Vector Graphics. Vector Graphics is a form of computer graphics in which, visual images are artificially created using geometric shapes like circle, rectangle, square, etc.

22). In the example below, when will the color pink be applied to the anchor element?

```
a:active { color: pink; }
```

A) The color of the link will display as pink after its been clicked or if the mouse is hovering over the link.

B) The color of the link will display as pink on mouse hover.

C) The color of the link will display as pink while the link is being clicked but before the mouse click is released.

D) The color of the link will display as pink before it has been clicked.

- ':active' selector is used to select the link which is clicked. ':link' selector selects unvisited links, ':visited' selects visited links and ':hover' selects mouse-hovered links.
- However, these selectors can be used in any element and not only <a> element. ':active' must come after ':hover' if declared in order to be effective.

23). To change the color of an SVG using CSS, which property is used?

A) Use background-fill to set the color inside the object and stroke or border to set the color of the border.

B) The color cannot be changed with CSS.

C) Use fill or background to set the color inside the object and stroke to set the color of the border.

D) Use fill to set the color inside the object and stroke to set the color of the border.

- If we define a circle inside the <svg> element, the 'fill' attribute inside the <circle> element defines the color of the circle whereas 'stroke' and 'stroke-width' attributes inside <circle> element define the outline of the <circle> element. 'stroke' defines border color and 'stroke-width' border width of the element.

24). When using position: fixed, what will the element always be positioned relative to?

A) the closest element with position: relative

B) the viewport

C) the parent element

D) the wrapper element

- In the file position.html, the element having 'position: fixed' is given 'left' property to which it responds and moves in that direction. But the element having this property has moved with respect to the <body> of the HTML page or we can say with respect to the viewport.

25). By default, a background image will repeat _____.

A) only if the background-repeat property is set to repeat

B) indefinitely, vertically, and horizontally

C) indefinitely on the horizontal axis only

D) once, on the x and y axis

- By default, the background image repeats itself over both x and y axis. There are various properties which can alter the repetition of a background image. For eg: if 'no-repeat' value is added to the property 'background-repeat', the image will not be repeated.
- Similarly, values 'repeat-x' and 'repeat-y' will make the repeat of background image repeat only in X and Y directions respectively. The default value of this property is 'repeat' enabling the repeat of background image horizontally as well as vertically.

26). When using media queries, media types are used to target a device category. Which choice lists current valid media types?

A) print, screen, aural

B) print, screen, television

C) print, screen, speech

D) print, speech, device

- Media Queries are nothing but special CSS styles which work only when the dimensions of the viewport of the browser get to a specified value. i.e: we see lot of web applications which behave differently in computer screen than they do when they are opened in mobile phones. This change in styling can be CSS programmed by media queries.
- The file [media-types.html](#) show how change in the viewport dimension impact the background color. We can see this change on adjusting the size of control panel of the browser.
- We can see different background colors for different mobile screens. The 'screen' written in the media query defined inside the <style> element is the media-type. There are basically 3 media-types: 'print', 'screen' & 'speech'.
- The 'print' type of media query applies only when the page in the browser goes out for printing. The 'screen' type applies when the page is opened on a screen which can be a computer screen or mobile screen, etc. The 'speech' type applies when we listen to audio from the page. Mostly for web applications, we use 'screen' media type only.
- From the media query '@media screen and (orientation: portrait) and (min-width: 400px)', the 'orientation' and 'min-width' are properties of the query which can be used so that the styling specified inside the query only applies when the viewport adjusts to the values of the properties mentioned in the query.
- For eg: the query mentioned in previous point applies only when we apply mobile screen on the browser along with a width greater or equal to 400px. Only if both these properties' values are satisfied, one can see 'cornflowerblue' color on the screen.
- If the query is like '@media screen and (orientation: portrait), (min-width: 400px)' the CSS will require any one of the two conditions ('orientation: portrait' and 'min-width: 400px') above to be satisfied because the ',' here instead of 'and' will work as an 'OR' operator.
- One important note here is that the media queries will run in mobile screens only if we add a <meta> element inside the <head> element of the document as added in the file.

27). How would you make the first letter of every paragraph on the page red?

A) p::first-letter { color: red; }

B) p:first-letter { color: red; }

C) first-letter::p { color: red; }

D) first-letter:p { color: red; }

- There are various types of CSS selectors: Type, Id, Class, Universal, Combination & Psuedo.
- Type selectors are the selectors picking elements on the basis of their type. i.e: `p { ...; }`, `div { ...; }`, `h2 { ...; }`, etc. Class and Id selectors select elements based on their class and id. i.e: `.div1 { ...; }`, `#div2 { ...; }`. Universal selectors select everything on the document. i.e: `* { ...; }`.
- The file [selectors.html](#) shows different type of combination and pseudo selectors. The first selector '`div p { ...; }`' selects a `<p>` element which is the child of `<div>` element. Hence, `#p1` is colored blue and not `#p2` element. '`#div > span`' selects the `` element which is the direct child of `#div` element. Hence, the `#span1` element is not colored because it is the grand child of `#div` element.
- The selector '`#div3 + .p3`' selects the element right next to `#div3` element having class '`p3`'. Hence the styling never gets applied on element inside `#div3` having class '`p3`'. The selector '`.p3 ~ #p4`' selects `#p4` element which is the direct sibling of element having class '`p3`'.
- There are two categories of psuedo selectors: psuedo class selectors & psuedo element selectors. Psuedo class selectors are used to define the state of an element, i.e: clicked, hovered, etc. whereas psuedo element selectors are use to style a part of an element, i.e: first letter, first line, etc.
- Psuedo class selectors are denoted by ':' mark. In the file, the '`a:hover`' is such a selector because 'hover' defines the state of the element. Psuedo element selectors are denoted by '::' mark. In the file, '`#p4::first-letter`' is such a selector as it styles a part of an element. For this reason, option A is the correct answer.

28). In this example, what is the selector, property, and value?

`p { color: #000000; }`

- A) "p" is the selector, "#000000" is the property, "color" is the value
- B) "p" is the selector, "color" is the property, "#000000" is the value
- C) "color" is the selector, "#000000" is the property, "#p" is the value
- D) "color" is the selector, "p" is the property, "#000000" is the value

29). What is the rem unit based on?

- A) The rem unit is relative to the font-size of the p element.
- B) You have to set the value for the rem unit by writing a declaration such as `rem { font-size: 1 Spx; }`

C) The rem unit is relative to the font-size of the containing (parent) element.

D) The rem unit is relative to the font-size of the root element of the page.

- There are many units for measuring font-size of an element. Among these units, 'em' and 'rem' are units which really help build scalable applications.
- The value 'xyz em' means the font-size and the line height of a particular element is xyz times that of the parent element. For eg: let's say we have a <div> element which has defined the 'font-size' property to 16px value. Then we have a <p> element as the child of <div> element which has defined the 'font-size' property's value to '2em'. So the text inside the <p> element will have 'font-size' equal to $2 * 16 = 32px$. Same goes with the 'line-height' property but for that, the <div> element must hold the 'font-size' property and its value. If this property is not defined, the default 'font-size' value is taken.
- The unit 'rem' does the same as 'em' but with just one difference. 'em' takes value from parent element whereas 'rem' takes reference from the root element. So for this if we want the custom scale of 'font-size', we have to define the 'font-size' property inside the <html> element directly. Else, default font-size will be taken and the values defined on 'rem' will scale accordingly.

30). Which choice would give a block element rounded corners?

A) corner-radius: 10px;

B) border-corner: 10px;

C) corner-curve: 10px;

D) border-radius: 10px;

31). In the following media query example, what conditions are being targeted?

@media (min-width: 1024px), screen and (orientation: landscape) { ... }

A) The rule will apply to a device that has either a width of 1024px or wider, or is a screen device in landscape mode.

B) The rule will apply to a device that has a width of 1024px or narrower and is a screen device in landscape mode.

C) The rule will apply to a device that has a width of 1024px or wider and is a screen device in landscape mode.

D) The rule will apply to a device that has a width of 1024px or narrower, or is a screen device in landscape mode.

- Just like $2 + 3 * 5 = 17$ according to the BODMAS rule, in the equation 'A' OR B AND 'C', firstly 'B' AND 'C' is calculated, then 'A' OR "result of 'B' AND 'C'" is calculated.

- Hence in the given media query, first the condition 'screen and (orientation: landscape)' is checked and then the condition '(min-width: 1024px)' is checked and ORed with the result of first checked condition.

32). CSS transform properties are used to change the shape and position of the selected objects. The transform-origin property specifies the location of the element's transformation origin. By default, what is the location of the origin?

A) the top left corner of the element

B) the center of the element

C) the top right corner of the element

D) the bottom left of the element

- The 'transform' property changes the shape and position of an element. With the help of this property, we can scale and rotate the element. The file [transform-2.html](#) clearly shows how the transformed element looks like in comparison to original element.
- Here we can see that #transform-1, #transform-2 & #transform-5 elements are scaled big to 1.5 times whereas rest of the elements are rotated 45degrees.
- We can rotate or scale an element but the point of origin remains a concern. For eg: the elements #transform-1 and #transform-2 both are scaled but the point of origin are different for both of them. Hence, both the elements look different.
- This point of origin is usually addressed using a property called 'transform-origin'. For element #transform-1, the 'transform-origin' value is '0% 0%' indicating x = 0px and y = 0px and also indicating that the point of origin will be top-left corner.
- The value 'x% y%' of 'transform-origin' property indicates x% right from left and y% bottom from top of the top-left corner of the original element.
- The file [transform.html](#) shows beautiful animation of how the different values of 'transform-origin' impact the position of the axis upon which the element is rotating.
- By default, the value of this property is '50% 50%' which is exactly the center of the original element. We can see that in #transform-6 element inside transform-2.html file.

33). Which of the following is not a valid color value? (Alternative: Which choice is not a valid color value?)

A) color: #000

B) color: rgb(0,0,0)

C) color: #000000

D) color: 000000

34). What is the vertical gap between the two elements below?

```
<div style="margin-bottom: 2rem;">Div 1</div>
```

```
<div style="margin-top: 2rem;">Div 2</div>
```

A) 2rem

B) 32px

C) 64px

D) 4rem

- If there are two sibling elements with one having top and another having bottom margins, the total distance or margin between them is equal to the greatest positive margin. This is called Margin Collapsing.
- In this case the total gap between #div-1 and #div-2 is 2rem as both the elements have equal margins. This example is very well demonstrated in the file [*margin-collapsing.html*](#).
- In this file we see that #div-3 is having height of 2rem it completely covers the gap between #div-1 and #div-2. This proves that the gap between #div-1 and #div-2 is merely 2px as the margin has collapsed.
- For margin collapsing the happen, there are certain conditions:- the elements like shown in the example must be adjacent; and there must be no border, padding, etc. inside either of the elements.
- Collapsing margin concept is relevant only in vertical margins.

35). When using the Flexbox method, what property and value is used to display flex items in a column?

A) flex-flow: column; or flex-direction: column

B) flex-flow: column;

C) flex-column: auto;

D) flex-direction: column;

- Flexbox is an entire module and not just a CSS property. Flexbox includes plenty of properties and values which basically deal with how are the items inside the element arranged.
- In any flexbox model, there are particularly two axis: 'main axis' and 'cross axis'. Both these axes are perpendicular to each other. The 'main axis' starts from a point in the model known as 'main-start' and ends at 'main-end'. Similarly, cross axis starts from 'cross-start' and ends at 'cross-end'. The distance between 'main-end' and 'main-start' is called 'main-size' and the same between 'cross-start' and 'cross-end' is called 'cross-size'.

- Most of the properties included in flexbox model can be applied only to parent element. The properties and the values in parent element determine the orientation, space between, wrapping, and its arrangement in main and/or cross axes.
- The file [flexbox.html](#) shows how different properties and their values impact the arrangement of child elements inside the container.
- The property 'flex-direction' will determine if the items inside the element will be arranged in row format or a column format. We can see the difference between the values 'row', 'row-reverse', 'column' & 'column-reverse' in the containers 1, 3, 11 & 12 respectively.
- By default, CSS will try to fit every item inside the container in one row. However, this setting can be altered by 'flex-wrap' property. The default value of this property is 'nowrap' meaning that everything will be stacked inside one line without considering defined widths of the elements. We can see the difference between the values 'wrap', 'nowrap' & 'wrap-reverse' in the containers 1, 5 & 9 respectively.
- The property 'flex-flow' is a shorthand for the properties 'flex-direction' & 'flex-wrap' combined. If we define 'flex-flow: column', then there will be no syntax error as the default value of 'flex-wrap' is 'nowrap'. This is why in the given problem, both 'flex-direction: column' and 'flex-flow: column' will display items in column direction. Hence option A is correct.
- Whenever we define 'flex-direction: row', the horizontal axis or row axis becomes our 'main axis' and vertical axis or column axis becomes our 'cross axis'. Opposite happens when we define 'flex-direction: column'.
- The property 'justify-content' determines how are the items arranged along the main axis of the model. The values 'flex-start' mean that the items will be stacked right next to each other from the 'main-start'; 'flex-end' mean that the items will be arranged the same way but starting from 'main-end'. The difference between the values 'flex-start', 'flex-end', 'center', 'space-around', 'space-between' & 'space-evenly' can clearly seen from containers 1, 2, 13, 14, 15 & 16 respectively.
- The property 'align-items' determine the arrangement of items inside the element along the 'cross axis' of the model. We can see the difference between the values 'flex-start', 'flex-end', 'center' & 'stretch' from the containers 17, 18, 19 & 20 respectively.
- The property 'align-content' particularly manages the lines between the items along the 'cross axis' of the model. Similarly, 'justify-content' manages the space between each element along the 'main axis'. We can see the difference between 'align-items: flex-start' and 'align-content: flex-start' from the containers 17 & 21 respectively. While 'align-items' preserves the space between items along 'cross axis', the 'align-content' particularly manipulates space between the items along the 'cross axis'. We can see the difference between the values 'flex-start', 'flex-end', 'center', 'stretch', 'space-around' & 'space-between' from containers 21, 22, 23, 24, 25 & 26 respectively.
- The properties 'row-gap' and 'column-gap' determine the space between rows and columns respectively which can be seen from containers 27 & 28. The property 'gap' is the shorthand for 'row-gap column-gap'. It means if we define 'gap: 10px 20px', we have defined 'row-gap: 10px' as well as 'column-gap: 20px'.

- While the above discussed properties are supposed to be applied on the container or the parent element, the properties like 'flex-grow' & 'align-self' are supposed to be applied on individual items.
- As shown in container 29, items 1, 2 & 4 take one unit of space, the item 3 takes 2 units of space. If we observe the HTML code, we can see that each of the child elements 'item 1', 'item 2' & 'item 4' have a class called 'single-grow' which defines 'flex-grow: 1' and the #item-3 element defines 'flex-grow: 2'. So we can say that 'flex-grow' defines how much proportional space an item must take or at least try to take in comparison to other items inside the model.
- In container 30, we see that the 'item 4' box is touching the bottom border and appears differently aligned than the rest of the elements. For this miracle, the property 'align-self: flex-end' defined on #item-4 is responsible. This property gives freedom to individual elements to place themselves however they want along the 'cross axis' of the model.

36). Which type of declaration will take precedence?

A) any declarations in user-agent stylesheets

B) important declarations in user stylesheets

C) normal declarations in author stylesheets

D) important declarations in author stylesheets

- 'cascade' is an algorithm which determines which property: value will apply on a particular element based upon the precedence of the declaration of the property.
- There are many layers under which the same style property with different values can be declared. It is upon the 'cascade' algorithm to decide which value to take and apply on the particular element. First let us define the layers here:-
- User-agent stylesheet:- Even if we do not apply any CSS styling on an HTML document, the browser by itself gives default styling to the element. This browser stylesheet is also known as 'user-agent' stylesheet.
- Author stylesheet:- The stylesheet as a web developer we apply on the HTML document manually either by inline styling, internal styling or by external styling, is called 'author' stylesheet.
- User stylesheet:- After applying the manual styling on a document, when we run it on browser, we can modify the stylesheet or even create new stylesheet on the browser itself. When we right click on any element and click on 'Inspect Element' option, we see the HTML as well as CSS code which we can modify, delete or even add new elements and style to it at our own will using the browser. This stylesheet is called 'user' stylesheet.
- Cascade stylesheet:- Apart from the above layers, we can define our own layer which is also known as cascade layer. The file [*cascading-layer.html*](#) shows a layer named 'type' is created which holds a stylesheet. All the styles outside of this 'type' layer are packed into a single layer known as 'anonymous layer'. The styles in the 'anonymous' layer will override the

styles defined inside the layer 'type' regardless of the specificity. Here we can see that the selector '.box p' has higher specificity compared to 'p'. But still the styling of only 'p' applies to the element.

- These stylesheets are also known as 'origins'. The cascading algorithm decides which value of a property among the same property declared in different origins must apply on an element depending upon 4 relevant factors:-
- Relevance:- The 'cascade' algorithm will filter out all the properties which are declared for different elements and keep only those which are relevant to the particular element.
- Importance:- The value of the property defined inside the origin of the greater precedence wins and only that value of the property gets applied on the particular element. The order of precedence is as follows:-

- 1). user-agent
- 2). user
- 3). author
- 4). CSS @keyframe animations
- 5). author (with !important style declaration)
- 6). user (with !important style declaration)
- 7). user-agent (with !important style declaration)
- 8). CSS transitions

Here 1st origin has least precedence and 8th origin has the highest precedence. As an example, we can say that if we define the property 'margin-left: 8px' in 'author' origin, 'margin-left: 12px !important' in 'user' origin and 'margin-left: 16px !important' in 'user-agent' origin, the value '16px' will be applied on the element as 'user-agent (with !important)' origin has precedence of 7 which is highest. Hence in the given question, option B is correct.

- Specificity:- If we have two properties which are relevant to the element and declared in the same origin, 'cascade' algorithm checks the 'specificity' of the selector holding the property and the property enclosed within the selector with highest 'specificity' wins and the value of that property gets applied on the element.
- Order of appearance:- If we have two classes relevant to the element, defined in same origin and having same specificity, the 'cascade' algorithm checks the order of the classes declared and the class which is last declared wins over other classes and has its property: value getting applied on the element.

37). Question 37 in the web.

38). There are two sibling combinators that can be used to select elements contained within the same parent element; the general sibling combinator (~) and the adjacent sibling combinator (+). Referring to example below, which elements will the styles be applied to?

```
h2 ~ p { color: blue; }
```

h2 + p { background: beige; }

<section>

<p>paragraph 1</p>

<h2>Heading</h2>

<p>paragraph 2</p>

<p>paragraph 3</p>

</section>

A) Paragraphs 2 and 3 will be blue. The h2 and paragraph 2 will have a beige background.

B) Paragraphs 2, and 3 will be blue, and paragraph 2 will have a beige background.

C) Paragraph 2 will be blue. Paragraphs 2 and 3 will have a beige background.

- The file [selectors.html](#) demonstrates the terminology of different selector elements. In the given problem, h2 ~ p { ...; } means select all the <p> elements coming right after <h2> element which also happen to be direct sibling of <h2> element.
- Whereas h2 + p { ...; } means select only first <p> element coming right after <h2> element which is also a direct sibling of <h2> element.
- Hence, paragraph 2 and 3 both will have blue color but only paragraph 2 will have beige background color.

39). Which element(s) will be blue?

h2 ~ p { color: blue; }

<section>

<p>P1</p>

<h2>H2</h2>

<p>P3</p>

<p>P4</p>

</section>

A) P3

B) P1, P3 and P4

C) P3 and P4

D) P1

40). Question 40 in the web.

41). There are many advantages to using icon fonts. What is one of those advantages?

- A) Icon fonts increase accessibility.
- B) Icon fonts can be used to replace custom fonts.
- C) Icon fonts can be styled with typography related properties such as font-size and color.
- D) Icon fonts are also web safe fonts.

42). What is the difference between 'display: none' and 'visibility: hidden'?

- A) Both will hide the element on the page, but 'display: none' has greater browser support. 'visibility: hidden' is a new property and does not have the best browser support
- B) 'display: none' hides the elements but maintains the space it previously occupied. 'visibility: hidden' will hide the element from view and remove it from the normal flow of the document
- C) 'display: none' hides the element from view and removes it from the normal flow of the document. 'visibility: hidden' will hide the element but maintains the space it previously occupied.
- D) There is no difference; both will hide the element on the page.

43). What selector and property would you use to scale an element to be 50% smaller on hover?

- A) element:hover {scale: 0.5;}
- B) element:hover {transform: scale(0.5);}
- C) element:hover {scale: 50%;}
- D) element:hover {transform: scale(50%);}

44). Which statement regarding icon fonts is true?

- A) Icon fonts can be inserted only using JavaScript.
- B) Icon fonts are inserted as inline images.
- C) Icon fonts require browser extensions.
- D) Icon fonts can be styled with typography-related properties such as font-size and color.

45). The values for the font-weight property can be keywords or numbers. For each numbered value below, what is the associated keyword?

font-weight: 400;

font-weight: 700;

- A) bold; normal

- B) normal; bold
- C) light; normal
- D) normal; bolder

46). Using the :nth-child pseudo class, what would be the most efficient way to style every third item in a list, no matter how many items are present, starting with item 2?

- A) `li:nth-child(3 + 2n) { margin: 0 5 px; }`
- B) `li:nth-child(3n + 2) { margin: 0 5 px; }`
- C) `li:nth-child(2), li:nth-child(5), li:nth-child(8) { margin: 0 5 px; }`
- D) `li:nth-child(2n + 3) { margin: 0 5 px; }`

- The CSS selector '`p:nth-child(2)`' selects every `<p>` element which is 2nd child of its parent.
- For $n = 0, 1, 2, 3, \text{etc.}$, the corresponding selector will be '`li:nth-child(2)`', '`li:nth-child(5)`', '`li:nth-child(8)`', '`li:nth-child(11)`'.

47). Which selector would select only internal links within the current page?

- A) `a[href="#"] { ... }`
- B) `a[href~="#"]`
- C) `a[href^="#"]`
- D) `a[href="#"]`

- By default, `a[href^="#"]` means CSS will select all the `<a>` elements whose 'href' attribute starts with "#".
- The `<a>` element having 'href' equal to '#' will redirect to the current page.

48). What is the difference between the margin and padding properties?

- A) Margin adds space around and inside of an element; padding adds space only inside of an element.
- B) Margin adds space around an element; padding adds space inside of an element.
- C) Margin adds a line around an element, padding adds space inside of an element.
- D) Margin adds space inside of an element, padding adds space around an element.

49). What is not a valid way of declaring a padding value of 10 pixels on the top and bottom, and 0 pixels on the left and right?

A) padding: 10px 10px 0px 0px;

B) padding: 10px 0px;

C) padding: 10px 0;

D) padding: 10px 0px 10px 0px;

- 'padding: 2px 4px 6px 8px' is a combination of properties:- 'padding-top: 2px', 'padding-right: 4px', 'padding-bottom: 8px' & 'padding-left: 10px'.
- 'padding: 2px 4px 6px' is a combination of properties:- 'padding-top: 2px', 'padding-right: 4px', 'padding-left: 4px' & 'padding-bottom: 6px'.
- 'padding: 2px 4px' is a combination of properties:- 'padding-top: 2px', 'padding-bottom: 2px', 'padding-right: 4px', 'padding-left: 4px'.

50). Is there an error in this code? If so, find the best description of the problem.

```
@font-face {  
    font-family: 'Avenir', sans-serif;  
    src: url('avenir.woff2') format('woff2'), url('avenir.woff') format('woff');  
}
```

A) The font file formats are not supported in modern browsers.

B) The src attribute requires a comma between the URL and format values.

C) There are no errors in the example.

D) The sans-serif inclusion is problematic.

- In the 'font-face' rule declaration, the 'font-family' property must include only one value which is supposed to be fetched from the given resource in the 'src' attribute below. To provide the browser with the hint of what is the format of the font, we can define the value inside format() function like it is defined in the problem. This helps browser to suitably pick the right font.

51). Which style places an element at a fixed location within its container?

A) position: absolute;

B) display: flex;

C) display: block;

D) float: left;

- 'display: block' property will display an element as if it is a block. For eg: <div> and <p> elements are naturally block elements which always start on new line. But, element is not one of them. If we give this property to element, every new element will start from new line acting as if it is a block. So, it has nothing to do with placing a block.
- 'display: flex' arranges child elements of the element on which this property gets applied. 'float: left' will display the element in the left corner of the parent element. This position can change with the change in position or size of the parent element.
- 'position: absolute' will place an element at a fixed point and the position of this point will be decided upon the 'float' property on the element itself and the nearest ancestor having 'position: relative' property. Hence option A is correct.

52). The calc() CSS function is often used for calculating relative values. In the example below, what is the specified margin-left value?

```
.example { margin-left: calc(5% + 5px); }
```

A) The left margin value is equal to 5% of its parents element's width plus 5px

B) The left margin value is equal to 5% of the viewport width plus 5px

C) The left margin value is equal to 5% of the closest positioned element's width plus 5px

D) The left margin value is equal to 5% of the selected element's width (.example) plus 5px

- The calc() is a special function that calculates the value of a property dynamically. We can use this for certain use cases like increasing the font size once the end user zooms the page.
- The file [calc.html](#) demonstrates how both the form and banner are placed with varying widths. We can see that the class 'banner' has a width of (100% - 80px) which is relative to the width of body of the document. We can see this by changing the width of the screen.
- Here the 'width: calc(100% - 80px)' means that width must be equal to 100% width of the parent of the element which is document body in this case subtracted by 80px length. Similar is the case with 'input' and '#formbox' selectors. Hence in the given problem, option A is correct.

53). Which code would you use to absolutely position an element of the logo class?

A) .logo { position: absolute; left: 100px; top: 150px; }

B) .logo { position: absolute; margin-left: 100px; margin-top: 150px; }

C) .logo { position: absolute; padding-left: 100px; padding-top: 150px; }

D) .logo { position: absolute; left-padding: 100px; top-padding: 150px; }

54). In this example, what color will Paragraph 1 be?

```
p:first-of-type { color: red; }
```

```
p { color: blue; }
```

```
.container { color: yellow; }
```

```
p:first-child { color: green; }
```

```
<div class="container">
```

```
    <h1>Heading</h1>
```

```
    <p>Paragraph1</p>
```

```
    <p>Paragraph2</p>
```

```
</div>
```

A) blue

B) green

C) red

D) yellow

- The selector 'p:first-of-type { ...; }' will select first <p> child element of its parent and 'p:first-child { ...; }' will select the element which is the first child of its parent and is a <p> element.
- In the demonstration as given in the file [selectors-2.html](#), we can see that paragraph 1 is red in color because it is not the first element of its parent but the first <p> element of its parent. Paragraph 2 will be blue because of the 'p { ...; }' selector getting applied on it.
- We can see that paragraph 3 is green in color because it is the first child of its parent. Paragraph 4 will be blue in color as normal but paragraph 5 will be red in color because it is not the first child but the first <p> element of its parent <body> element.

55). What is the ::placeholder pseudo-element used for?

- A) It is used to format the appearance of placeholder text within a form control.
- B) It specifies the default input text for a form control.
- C) It writes text content into a hyperlink tooltip.
- D) It writes text content into any page element.

56). Which statement is true of the single colon (:) or double colon (::) notations for pseudo-elements-for example, ::before and :before?

- A) All browsers support single and double colons for new and older pseudo-elements. So you can use either but it is convention to use single colons for consistency.
- B) In CSS3, the double colon notation (::) was introduced to create a consistency between pseudo-elements from pseudo-classes. For newer browsers, use the double colon notation. For IE8 and below, using single colon notation (:).
- C) Only the new CSS3 pseudo-elements require the double colon notation while the CSS2 pseudo-elements do not.
- D) In CSS3, the double colon notation (::) was introduced to differentiate pseudo-elements from pseudo-classes. However, modern browsers support both formats. Older browsers such as IE8 and below do not.

57). Which choice is not valid value for the font-style property?

- A) normal
- B) italic
- C) none
- D) oblique

58). When would you use the @font-face method?

- A) to set the font size of the text
- B) to load custom fonts into stylesheet

C) to change the name of the font declared in the font-family

D) to set the color of the text

- The 'font-face' CSS property is used to define custom fonts inside the document. Here is how it is declared:-
- ```
@font-face {
 font-family: myFont; /* required */
 src: url('sensation_light.woff'); /* required */
 font-stretch: normal; /* optional */
 font-style: italic; /* optional */
 font-weight: normal; /* optional */
}
```

59). You have a large image that needs to fit into a 400 x 200 pixel area. What should you resize the image to if your users are using Retina displays?

A) 2000 x 1400 pixels

B) 200 x 100 pixels

C) 800 x 400 pixels

D) 400 x 200 pixels

- CSS pixels are device dependent. It means they adjust themselves according to the screen resolution of the device.
- Some devices like Apple have Retina display with high resolution which means that the images inside it look sharper than in the other devices.
- The sharpness of the image depends upon the pixel density ratio that these displays offer. Pixel density ratio indicates the normal distance between two pixels in the screen. Screens with higher pixel density ratio have lower distance between two pixels which tend to make image look sharper.
- Retina displays have 2x of the pixel density ratio. It means that the distance between two pixels will be half in this display compared to other. Hence, the image in normal device whose width is 400px and height is 200px, would have width and height equivalent to 200px and 100px respectively in retina display.
- For this reason, we have to double the width and height of the image if we want to display it in retina display and hence option C with 800px x 400px image dimensions is correct.

60). In Chrome's Developer Tools view, where are the default styles listed?

A) under the User Agent Stylesheet section on the right

B) in the third panel under the Layout tab

C) under the HTML view on the left

D) in the middle panel

- When we press Ctrl + Shift + I in Chrome window, we see a control panel with 'Elements' tab activated. Under this tab, we can see both the HTML code and the CSS styles.
- Under the 'Styles' tab where the CSS styles are defined, if we scroll down deep, we can see one section having a heading 'user agent stylesheet'. Under this heading, the styles defined are the default styles provided by the browser which is Google Chrome in this case.

61). While HTML controls document structure, CSS controls \_\_\_\_\_.

A) semantic meaning

B) content meaning

C) document structure

D) content appearance

62). What is the recommended name you should give the folder that holds your project's images?

A) images

B) #images

C) Images

D) my images

63). What is an advantage of using inline CSS?

A) It is easier to manage.

B) It is easier to add multiple styles through it.

C) It can be used to quickly test local CSS overrides.



D) It reduces conflict with other CSS definition methods.

64). Which W3C status code represents a CSS specification that is fully implemented by modern browsers?

A) Proposed Recommendation

B) Working Draft

C) Recommendation

D) Candidate Recommendation

- W3C is the short form for 'World Wide Web Consortium'. This consortium is a community headed by Tim Berners Lee that maintains and develops the world wide web.
- Lets say for instance that W3C is developing a new version of CSS which has more features than the current ongoing version. This development will go in several stages which include:- Working Draft (WD), Candidate Recommendation (CR), Proposed Recommendation (PR) & W3C Recommendation (REC).
- After enough discussions, a Working Draft (WD) of the specification that is supposed to be implemented in the new feature of CSS is put up in public for community review. This is the first stage of the development.
- The next stage Candidate Recommendation (CR) is sure of meeting its goals and in this stage, aid from the development community is elicited for knowing how implementable the standard is.
- The next stage Proposed Recommendation (PR) the document of the specification is sent to W3C Advisory for final call on whether to implement the specification.
- The final stage is 'Recommendation' (REC) under which the W3C promotes and encourages widely to use the specification. At this stage, browsers fully implement the said specification.

65). Are any of the following declarations invalid?

color: red; /\* declaration A \*/

font-size: 1em; /\* declaration B \*/

padding: 10px 0; /\* declaration C \*/

A) Declaration A is invalid.

B) Declaration B is invalid.

C) Declaration C is invalid.

D) All declarations are valid.

66). Which CSS will cause your links to have a solid blue background that changes to semitransparent on hover?

A) `a:link { background: #0000ff; } a:hover { background: rgba(0, 0, 255, 0.5); }`

B) `a { color: blue; } a:hover { background: white; }`

C) `a:link { background: blue; } a:hover { color: rgba(0, 0, 255, 0.5); }`

D) `a:hover { background: rgba(blue, 50%); } a:link { background: rgba(blue); }`

67). Which CSS rule takes precedence over the others listed?

A) `div.sidebar {}`

B) `* {}`

C) `div#sidebar2 p {}`

D) `.sidebar p {}`

- The precedence of selectors will be calculated through specificity as all of these selectors belong to the same origin.
- The specificity score of 'div.sidebar' in 'id-class-type' format is '0-1-1'. Similarly, the score of selector in option B is '0-0-0', in option C is '1-0-2' and in option D is '0-1-1'.
- Hence, the selector in option B wins over other selectors giving it highest precedence.

68). The body of your page includes some HTML sections. How will it look with the following CSS applied?

```
body { background: #ffffff; /* white */ }
```

```
section {
```

```
 background: #0000ff; /* blue */
```

```
 height: 200px;
```

```
}
```

- A) blue sections on a white background.
- B) Yellow sections on a blue background.
- C) Green sections on a white background.
- D) blue sections on a red background.

69). Which CSS keyword can you use to override standard source order and specificity rules?

- A) !elevate!
- B) \*prime
- C) override
- D) !important

70). You can use the \_\_\_\_\_ pseudo-class to set a different color on a link if it was clicked on.

- A) a:visited
- B) a:hover
- C) a:link
- D) a:focus

71). Which color will look the brightest on your screen, assuming the background is white?

- A) background-color: #aaa;
- B) background-color: #999999;
- C) background-color: rgba(170, 170, 170, 0.5);
- D) background-color: rgba(170, 170, 170, 0.2);

- The brightness of a model on which we apply the 'background-color' property, can be defined by 'filter: brightness(xyz)' property – value. But on other hand, the brightness is also associated with the whiteness of the background color of the model. More the background color is white, more it is bright. The whiteness can be adjusted with the help of transparency of a model. More the model is transparent, whiter it is and thus brighter it is. For this reason, option D defining the least opacity of the colors in all the other options will pose brightest model among models posed by all the other options.

72). Which CSS selector can you use to select all elements on your page associated with the two classes header and clear?

A) ".header clear" {}

B) header#clear {}

C) .header.clear {}

D) .header clear {}

- The file [selectors.html](#) shows that selector '.c1.c2' gets applied on #div4 element. The selector '.c1.c2' indicates select an element having both 'c1' and 'c2' classes. We can also see that 'c1' and 'c2' individual selectors are also defined individually with their own set of properties and values. But these selectors are overridden by '.c1.c2' selector.

73). A universal selector is specified using \_\_\_\_\_.

A) "h1" string

B) "a" character

C) "p" character

D) "\*" character

74). In the following CSS code, 'h1' is the \_\_\_\_\_, while 'color' is the \_\_\_\_\_.

```
h1 { color: red; }
```

A) property; declaration

B) declaration; rule

C) "p" character

D) selector; property

75). What is an alternate way to define the following CSS rule?

```
font-weight: bold;
```

A) font-weight: 400;

B) font-weight: medium;

C) font-weight: 700;

D) font-weight: Black;

76). You want your styling to be based on a font stack consisting of three fonts. Where should the generic font for your font family be specified?

A) It should be the first one on the list.

B) Generic fonts are discouraged from this list.

C) It should be the last one on the list.

D) It should be the second one on the list.

- We can define the font on any text defined inside an element with the help of 'font-family' property defined inside the element selector. For eg: 'font-family: "Times New Roman", Times, serif'. This property can hold several font styles at once in a fallback system. This system means that the CSS will try the first font defined inside the property from left. If it fails to fetch the first font, CSS will go for the second font from left. If that font fails, CSS will go to next until it reaches the last font. This font is called as the generic font as it will be applied if all the declared fonts fail. Generic font is always declared last on the list of fonts.

77). When using a font stack to declare the font family, in what order should the values appear?

A) The first value is the first choice, followed by alternative options, ordered by preference. The last option should be a generic font.

B) The first value is the first choice. The order of the alternative options does not matter. It depends on what is available on the user's computer.

C) The first value is the first choice, and must be followed by at least one alternative option before adding the generic font.

D) The first value is the first choice, followed by a maximum of three alternatives.

78). What is one disadvantage of using a web font service?

A) It requires you to host font files on your own server.

B) It uses more of your site's bandwidth.

C) It offers a narrow selection of custom fonts.

D) It is not always a free service.

79). How do you add Google fonts to your project?

A) by using an HTML link element referring to a Google-provided CSS

B) by embedding the font file directly into the project's master JavaScript

C) by using a Google-specific CSS syntax that directly links to the desired font file

D) by using a standard font-face CSS definition sourcing a font file on Google's servers

- Unlike other web fonts, google fonts are completely free of cost to use and also are hassle free. The method of using google fonts is also very simple.
- Following are the steps of using google fonts:-
  - 1). Visit <https://fonts.google.com/> website and browse through a list of available fonts.
  - 2). Select the desired font. A page describing all the details of that font would open up.
  - 3). Click on the plus (+) icon under the 'Styles' section.
  - 4). On the top right corner of the page there must be a symbol which would show the title "View selected families" on hover. Click on it.
  - 5). Copy the <link> HTML code and paste into the <head> element inside your file. Also declare CSS as specified in the page wherever you intend to use the google font.
- The file [web-fonts.html](#) is a living example of this.

80). Using the following HTML and CSS example, what will equivalent pixel value be for .em and .rem elements?

```
html { font-size: 10px; }
```

```
body { font-size: 2rem; }
```

```
.rem { font-size: 1.5rem; }
```

```
.em { font-size: 2em; }
```

```
<body>
```

```
<p class="rem"></p>
```

```
<p class="em"></p>
```

</body>

A) The .rem will be equivalent to 25px; the .em value will be 20px.

B) The .rem will be equivalent to 15px; the .em value will be 20px.

C) The .rem will be equivalent to 15px; the .em value will be 40px.

D) The .rem will be equivalent to 20px; the .em value will be 40px.

- The first selector 'html { ...; }' will give a font size of 10px. The second selector 'body { ...; }' will give a font size of  $2\text{rem} = 2 * 10 = 20\text{px}$ .
- The third selector '.rem { ...; }' will give a font size of  $1.5\text{rem} = 1.5 * 10 = 15\text{px}$ . Whereas the fourth selector '.em { ...; }' will give a font size of  $2\text{em} = 2 * (\text{font size of parent} = \text{<body>}) = 2 * 20 = 40\text{px}$ .

81). What property is used to adjust the space between text characters?

A) font-style

B) text-transform

C) font-variant

D) letter-spacing

- The property 'letter-spacing' defines space between two words in text. An example of this can be seen from [web-fonts.html](#) file where we have declared manual spacing in the text.

82). What is the correct syntax for changing the cursor from an arrow to a pointing hand when it interacts with a named element?

A) .element {cursor: pointer;}

B) .element {cursor: hand;}

C) .element {cursor: move-hand;}

D) .element {cursor: pointer-hand;}

83). What is the effect of this style?

background-position: 10% 50%;

A) The background image is placed 10% from the left and 50% from the top of its container

- B) The background image is placed 10% from the bottom and 50% from the left of its container
- C) The background image is placed 10% from the right and 50% from the bottom of its container
- D) The background image is placed 10% from the top and 50% from the left of its container

84). How will the grid items display?

`grid-template-columns: 2fr 1fr;`

- A) The first column is twice the height of the second column and will be as wide as the content
- B) The first column is half the size of the container and the second column will absorb the remaining space
- C) The first column is twice as wide as the second column and will fit proportionally within the grid container
- D) The first column is twice the width and height of the second column, and will fit proportionally within the grid container

85). Which style rule would make the image 50% smaller during a hover?

``

- A) `img#photo:hover {scale: 0.5;}`
- B) `img#photo:hover {transform: scale(0.5);}`
- C) `img#photo {hover-scale: 0.5;}`
- D) `img#photo:hover {size: smaller;}`

86). Which CSS properties can you use to create a rounded corner on just the top-left and top-right corners of an element?

- A. `border-radius: 10px 10px 0 0;`
  - B. `border-top-left-radius: 10px;` and `border-top-right-radius: 10px;`
  - C. `border-radius: 10px 0;`
  - D. `border-top-radius: 10px;`
- A) A and C
  - B) C and D



C) B and C

D) A and B

- The 'border-radius' property is the shorthand for 'border-top-left-radius', 'border-top-right-radius', 'border-bottom-right-radius' & 'border-bottom-left-radius'. This means that 'border-radius: 10px 10px 0 0' will define 'border-top-left-radius: 10px', 'border-top-right-radius: 10px', 'border-bottom-right-radius: 0' & 'border-bottom-left-radius: 0'.
- 'border-radius: 5px 15px 30px 20px'  
top-left corner: 5px, top-right corner: 15px, bottom-right corner: 30px, bottom-left corner: 20px.
- 'border-radius: 15px 50px 30px'  
top-left corner: 15px, top-right & bottom-left corners: 50px, bottom-right corner: 30px.
- 'border-radius: 15px 50px'  
top-left & bottom-right corners: 15px, top-right & bottom-left corners: 50px.
- 'border-radius: 15px'  
top-left, top-right, bottom-left & bottom-right corners: 15px.
- Even if we define less values inside the 'border-radius' property, all the four corners tend to get rounded. Hence, options C & D are incorrect. Hence, options A & B apply perfectly.

87). Review the HTML example below. Then choose the list of selectors that select the <p>, from lowest to highest specificity.

<section>

<p class="example">...</p>

</section>

- A) 1. section \\* { ...; } || 2. [class\*='example'] { ...; } || 3. p.example { ...; } || 4. section p { ...; }
- B) 1. p { ...; } || 2. p.example { ...; } || 3. section p { ...; } || 4. [class\*='example'] { ...; }
- C) 1. p.example { ...; } || 2. section p { ...; } || 3. [class\*='example'] { ...; } || 4. section \\* { ...; }
- D) 1. p { ...; } || 2. section p { ...; } || 3. [class\*='example'] { ...; } || 4. p.example { ...; }

88). Which property is used to create a drop shadow effect on an HTML element?

A) element-shadow

B) outer-shadow

C) dropbox-shadow

D) box-shadow

- The property 'box-shadow: 5px 10px 8px 12px #888888 inset' sets the shadow effect on the HTML element.
- In the above defined property, '5px' is the vertical offset indicating height of the shadow above the top of element. Negative value will give offset on the opposite side. i.e: height below the bottom of the element. '10px' is the horizontal offset indicating width of shadow from the right of the right side of the element. Negative value will give offset in opposite side. i.e: width of the element from the left of the left side. Both these attributes are required to fill the property. Rest all the attributes are optional
- '8px' is the blurriness radius. Higher the number, blurrier the shadow is. '12px' defines the spread of the shadow. '#888888' is the shadow color and 'inset' attribute will set the shadow inside the element instead of setting it outside the element.

89). What is the correct selector for targeting all text inputs that are not disabled?

A) `input[type="text"]:not([disabled]) {...}`

B) `input[type="text"]:not("disabled") {...}`

C) `input[type*="text"]:not([disabled="disabled"]) {...}`

D) `input[type="text"]:not([type="disabled"]) {...}`

- ':not(p)' CSS selector will select all the elements that are not <p> elements. The selector of a text input which is disabled is 'input[disabled]'.
- In the option A defined selector, 'input[type="text"]' will first target all the text inputs. Then '[disabled]' will select all the elements that have 'disabled = true' property inside the text input. The ':not' operating selector will remove all the elements having 'disabled: true' property from the target and target all the remaining text inputs.

90). How can you create a semi-transparent background color?

A) `background-color: hsl(0, 0, 0, 0.5);`

B) `background-color: rgbx(0, 0, 0, 0.5);`

C) `background-color: rgba(0, 0, 0, 0.5);`

D) `background-color: rgba(0, 0, 0, 1);`

91). Using this HTML markup, how would you select only the headings contained within the <header> element?

```
<header>

 <h1>Heading 1</h1>

 <h2>Heading 2</h2>

</header>

<h2>Heading 2</h2>
```

A) header h1, header h2 {...}

B) header h1 + header h2 {...}

C) header h1, h2 {...}

D) h1, h2 {...}

92). Suppose you want to have a list of items (.item) displayed in a row and in reverse order using flexbox. What is the error in the CSS below?

```
.container { display: flex; }

.item {

 border: 1px solid red;

 flex-direction: row-reverse;

}
```

A) The value for flex-direction should be reverse-row.

B) The .container element should have a property of flex: display.

C) The flex-direction property should be declared in the container.

D) The display value should be flex-inline to display the items in a row.

93). Which choice is not a valid transition?

A) transition: margin 1000ms ease-in-out;

B) transition: color 1.3s ease-in;

C) transition: position 400ms linear;

D) transition: opacity 1s ease-in;

- The property 'transition' is a shorthand of mainly four properties: 'transition-property', 'transition-duration', 'transition-timing-function', 'transition-delay'. The 'transition' property makes the styling property of any element dynamic and dependent of user action on the web page.
- For eg: a container when hovered upon changes its color and when cursor is moved back, it again goes back to having its original color. The dynamic styling of this type can be induced by 'transition' property.
- The file [transition.html](#) shows the effects of transition property on elements. Because the property 'transition' is dependent on user action, it is necessary to define a property on an element which applies only when user interacts with the element and that property is same as the 'transition-property' applied in the 'transition' super property of the same element. i.e: selectors 'input[type=checkbox]:checked', '.container:hover' and '.container-2:hover' define such properties. The selector 'input[type=checkbox]:checked' has a property 'margin' which is also defined as the 'transition-property' in the 'transition' super property defined inside 'input[type=checkbox]' selector. Because of this, the margin of element only changes when the user checks the input checkbox.
- 'transition-property' is the property which needs to be altered. 'duration' is the duration of transition. 'ease-in', 'ease-out', 'ease-in-out' are 'transition-timing-functions'. This indicates how much speed will transition the transition have over time. For eg: 'ease-in' means that the transition will happen slowly during the start of the 'transition-duration' but will gradually pick up speed and will become relatively faster. The property 'ease-out' is the opposite; meaning it will give a fast start and relatively slow end. 'ease-in-out' will give both, a slow start and a slow end. 'transition-delay' will delay the transition by some seconds or milliseconds depending upon the value of the property.
- The file shows examples of all the transition properties mentioned in the options above. There is a 'transition-property' for 'margin', 'color' & 'opacity' but not for 'position'. Hence the transition defined in option C is invalid transition.

94). In this example, what color will the paragraphs be and why?

```
article p { color: blue; }
```

```
article > p { color: green; }
```

```
<article>
```

```
<p>Paragraph 1</p>
```

<aside> <p>Paragraph 2</p> </aside>

</article>

- A) Paragraph 1 will be blue. Paragraph 2 will be green.
- B) Both paragraphs will be green.
- C) Paragraph 1 will be green. Paragraph 2 will be blue.
- D) Both paragraphs will be blue.

95). Review the declaration of border style shown below. What is the corresponding longhand syntax?

border: 1px solid red;

- A) border-size: 1px; border-style: solid; border-color: red;
- B) border-size: 1px; border-type: solid; border-color: red;
- C) border-width: 1px; border-style: solid; border-color: red;
- D) border-width: 1px; border-line: solid; border-color: red;

96). Pseudo-classes are used to \_\_\_\_\_.

- A) style the state of the selected element
- B) insert presentational content
- C) style a specific part of the selected element
- D) style the elements using class selectors

97). In this example, what styles will be applied to which elements?

section { color: gray; }

<section>

<p>paragraph</p>

<a href="#">link</a>

</section>

- A) The paragraph and link will be gray.
- B) The background color of the section element will be gray.
- C) The paragraph will be gray. The link will be the browser default, black.
- D) Only the paragraph will be gray.

98). Which answer is an example of a type selector (also sometimes referred to as an element selector)?

- A) .header {...}
- B) header {...}
- C) #header {...}
- D) header > h1 {...}

99). What is the correct order for listing different link states in a website so those states display correctly on the page?

- A) a, a:hover
- B) :link, :visited, :hover, :active, :focus
- C) :active, :focus, :hover, :link, :visited
- D) :link, :visited, :focus, :hover, :active

- There is a priority in styles which is supposed to be followed when <a> element is concerned. The file [a-tag-states.html](#) is an example of this.
- As we can see in that file five psuedo class selectors are applied on one <a> element. Whenever the file is opened in the web first the link is black colored which indicates 'a:link' psuedo class selector is applied. 'a:link' selector gets applied on unvisited links.
- When the link is hovered upon, the 'a:hover' selector gets applied. But if this style is written before the 'a:link' selector, 'a:link' will override the 'a:hover' even if the link is actually hovered upon, just because the link is still unvisited. Once the link is visited, the overriding charade will end because 'a:link' will never be applied on the link element.
- After the link is visited, 'a:visited' selector gets applied. Now if 'a:hover' is declared before 'a:visited' the style of 'a:hover' will never get applied on the element even if the element is actually hovered upon because 'a:hover' will be overridden by 'a:visited' as the link is

actually visited. Hence, 'a:hover' is always declared after 'a:link' and 'a:visited' to avoid style overriding issues.

- The selector 'a:active' gets applied only at the time the link is clicked. This style must be declared only after 'a:hover' because if 'a:hover' is placed after 'a:active', the 'a:hover' will override the other selector even when the link is actually clicked because at the time of click, the cursor is still hovering upon the element. Hence, 'a:active' must always be declared after 'a:hover'.
- As far as the selector 'a:focus' goes, its order and priority of declaration does not matter at all. It can be declared anywhere at any position without any sort of styles getting effected. For this reason, options B and D are both correct.

100). Which selector is used to select the paragraph element that is a direct descendent of section?

- A) section \* p
- B) section + p
- C) section ~ p
- D) section > p

101). For this code, what is the font color of the hypertext link?

```
ul { --color: red; }
```

```
p { color: var(--color); }
```

```
a { color: var(--color, orange); }
```

```
<p>Paragraph</p>
```

```

```

```
 list item a link
```

```
 list item
```

```

```

A) red

B) orange

C) blue

D) black

- In CSS, we can declare variables and use it wherever we find convenient. This is particularly helpful in large & complex web applications. These variables can be set or manipulated with either manually or by using JavaScript.

- Just like variables in any programming language, variables in CSS also have two kind of scopes: local and global. The file [variables.html](#) shows how can variables be defined and used. The scope of each variable totally depends upon the position of its definition.
- Few examples of variable definition and access are as:-
 

```
--color: red // variable definition
color: var(--color) // fetch value of variable '--color' and use it as a value to the property 'color'
color: var(--color, orange) // fetch value of '--color' if it is accessible. If CSS does not get the value of '--color', the value to the property 'color' will be 'orange' by default.
```
- The variables declared in ':root' selector are global variables. These variables are accessible from anywhere. Because '--globalColor' is defined in ':root' selector, it can be accessed in all the other selectors and that color is only applied.
- The variable '--color' is defined in 'ul' selector. Hence it is available in 'ul' as well as all the selectors whose selecting elements are descendants of <ul>. Here, <a> is a descendant of <ul> and due to this, the '--color' defined in 'ul' selector is accessible in 'a' selector also. For this reason, the hyperlink text is colored red.
- Because <p> element is not a descendant of <ul> 'p' selector cannot fetch the value of '--color' so CSS uses the fallback color which is var(--globalColor). Because '--globalColor' is a global variable, it is accessible in 'p' selector and hence, the text inside <p> will be colored 'yellow'.

102). Which statement is not true?

- A) Specificity determines which CSS rule is applied by the browsers.
- B) When two selectors apply to the same element, the one with lower specificity wins.
- C) The last rule defined overrides all previous rules and even conflicting rules.
- D) Rules with more specific selectors have greater specificity.

103). What is the output of the margin value when used within this context, assuming that its containing element is larger than 800px?

```
.example {
 width: 800px;
 margin: 0 auto;
}
```

- A) The example element will have 0 margin space around the whole element. The auto value will center align the element horizontally and vertically within its container.
- B) The example element will have 0 margin space on the left and right. It will be sized automatically on the top and bottom, which will center align the element within its container.



C) The example element will have 0 margin space on the top and bottom. The margin will be sized automatically on the left and right, which may center align the element within its container.

D) The margin value is invalid because its missing a unit measurement after the 0.

- The 'margin' property is the shorthand for 'margin-top', 'margin-right', 'margin-bottom' & 'margin-left' properties. If there are two values in 'margin' property, like in 'margin: 0 auto', the 'margin-top' and 'margin-bottom' properties will have values 0px whereas the other two properties will have value 'auto'.
- Hence, the example element will have zero margin space on top and bottom and it will be sized automatically on left and right side.

104). There are currently four viewport-percentage lengths that can be used to define the value relative to the viewport size: vw, vh, vmin, and vmax. If the current viewport size has a width of 800px and a height of 600px, what will these values be equivalent to in pixels?

10vw = ?px

10vh = ?px

10vmin = ?px

10vmax = ?px

A) 10vw = 80px 10vh = 60px 10vmin = 60px 10vmax = 80px

B) 10vw = 60px 10vh = 80px 10vmin = 80px 10vmax = 60px

C) 10vw = 8px 10vh = 6px 10vmin = 6px 10vmax = 8px

D) 10vw = 6px 10vh = 8px 10vmin = 8px 10vmax = 6px

- The measurement units 'vw', 'vh', 'vmin' and 'vmax' measure with respect to the viewport dimensions of the browser.
- 1vw = 1% of the viewport width.
- 1vh = 1% of the viewport height.
- 1vmin = 1% of the minimum dimension of the viewport (i.e: 1% of 600px in 800px x 600px viewport size).
- 1vmax = 1% of the maximum dimension of the viewport (i.e: 1% of 800px in 800px x 600px viewport size).

105). Referring to the HTML markup and CSS example below, which element(s) will be targeted?

```
p:first-of-type:first-letter { color: red; }
```

```
<body>
```

```
 <p>Paragraph 1.</p>
```

```
 <p>Paragraph 2.</p>
```

```
</article>
```

```

 <h1>Heading</h1>

 <p>Paragraph 3.</p>

 <p>paragraph 4.</p>

 </article>

 <section>

 <p>Paragraph 5.</p>

 <p>Paragraph 6.</p>

 </section>

</body>

```

- A) The first letter in all paragraphs will be red.
  - B) Only the first letter in paragraphs 1 and 5 will be red.
  - C) The first letter in paragraphs 1, 3, and 5 will be red.
  - D) Only the first letter in paragraph 1 will be red.
- 'p:first-of-type' selects all the <p> elements which are first <p> child of their respective parents whereas 'p:first-of-type:first-letter' will select all the first letters of the text defined inside the already selected <p> elements.

106). Which five style features are associated with the box model?

- A) margin, padding, border, width, height
- B) width, height, z-index, overflow, font size
- C) margin, padding, font size, line height, border
- D) font size, line height, letter spacing, width, height

107). Which choice will not set all links that include domain.com to pink?

- A) `a[href$='domain.com'] { color: pink; }`
  - B) `a[href='*domain.com'] { color: pink; }`
  - C) `a[href*='domain.com'] { color: rgb(255, 155, 155); }`
  - D) `a[href*='domain.com'] { color: pink; }`
- Option A will target all the <a> elements whose 'href' value will end with substring 'domain.com'. Options C & D will target all the <a> elements whose 'href' attribute will contain the substring 'domain.com'. Option B will target the <a> elements having 'href' value exactly equal to '\*domain.com' which is not same as 'domain.com' and hence, it will not color the desired element to pink.

108). Which property and value pair could be used to apply a linear gradient effect?

A) `css background: linear-gradient(#648880, #293f50);`

B) `css background-image: linear(#648880, #293f50);`

C) `css background: gradient(linear, #648880, #293f50);`

D) `css background-color: linear-gradient(#648880, #293f50);`

- If we want to add more than one background color in the element and also want a smooth transition between all of them, the 'linear-gradient' can be used.
- The file [linear-gradient.html](#) shows this very clearly. We can see that there are three elements: #div-1, #div-2 & #div-3, all having two colors and there is smooth transition between both the colors in all the elements when seen in the web.
- Inside the 'linear-gradient()' function, in the first argument we can define the direction of the color i.e: in the #div-2 element, 'to right' direction is defined and hence the colors in #div-2 go from left to right. Similar thing happens in #div-3 element.

109). You want to add a background circle behind an icon. Which style declaration is correct?

A)

```
.glyphicon-bgcircle {
 circle-radius: 50%;
 margins: 50px;
 background-color: #fdadc6;
 color: rgba(255, 255, 255, 1);
 font-size: 24px;
}
```

B)

```
glyphicon-bgcircle {
 border-circle: 50%;
 padding: 50px;
 background-color: #fdadc6;
 color: rgba(255, 255, 255, 1);
 font-size: 24px;
}
```

C)

```
.glyphicon-bgcircle {
 border-radius: 50%;
```

```
padding: 50px;
background-color: #fdadc6;
color: rgba(255, 255, 255, 1);
font-size: 24px;
}
```

D)

```
.glyphicon-bgcircle {
 radius-rounded: 50%;
 margins: auto;
 background-color: #fdadc6;
 color: rgba(255, 255, 255, 1);
 font-size: 24px;
}
```

110). Which items are valid values for the font-size property?

- A. font-size: xsmall
- B. font-size: 50%
- C. font-size: 1em
- D. font-size: 20px

A) C, D

B) B, C, D

C) A, C

D) A, B, C, D

- The font-size of an element can be valued in lot of units. Apart from units 'em' and 'px', '%' is also used; i.e: 'font-size: 5%' means 5% of the size of parent element's font size.
- Other values are 'x-small', 'xx-small', 'small', 'large', 'x-large', 'xx-large' indicating the absolute size of fonts. The values 'smaller' & 'larger' are relative to the font size of parent element. The default value of font-size is 'medium'.

111). Question 111 on the web.

112). Given this code, which CSS declaration of .overlay will span the entire width and height of its container?

<style>

```
.container {
 position: relative;
 height: 200px;
 width: 200px;
 border: 1px solid black;
}
</style>
<div class="container"><div class="overlay"></div></div>
```

A)

```
.overlay {
 position: static;
 top: 200px;
 bottom: 200px;
 right: 200px;
 left: 200px;
 background-color: rgba(0, 0, 0, 0.5);
}
```

B)

```
.overlay {
 position: absolute;
 top: 200px;
 bottom: 200px;
 right: 200px;
 left: 200px;
 background-color: rgba(0, 0, 0, 0.5);
}
```

C)

```
.overlay {
 position: static;
 top: 0;
 bottom: 0;
 right: 0;
 left: 0;
```

```
background-color: rgba(0, 0, 0, 0.5);
}
```

D)

```
.overlay {
 position: absolute;
 top: 0;
 bottom: 0;
 right: 0;
 left: 0;
 background-color: rgba(0, 0, 0, 0.5);
}
```

113). Which missing line of code would place the text on top of the image?

```
<div class="container">

 <p>The z-index property is cool!</p>
</div>

img {
 position: absolute;
 left: 0px;
 top: 0px;
 // Missing line
}
```

A) z-index: 1;

B) z-index: -1;

C) z-index: 0;

D) z-index: true;

- The property 'z-index' only works on positioned elements. So, if there are two elements stacked inside a container as shown in the question, the elevation of each element in respect to others depends upon the 'z-index' given on each element.
- When we declare 'z-index' property in <img> element, the <img> element tries to elevate itself in comparison to its sibling elements which is <p> in this case.
- We we define 'z-index: 0' or 'z-index: 1' inside the <img> element, the 'z-index' of <p> is first checked which we have not defined at all. So, the value of 'z-index' for <p> element is

taken to be '-1' by default. Now because the 'z-index' of <img> is greater than that of <p>, the image will show on top of the text.

- But if we define the 'z-index: -1' for <img> element, it will indicate the CSS to keep all the sibling elements of <img> above it and keep <img> at the bottom. Hence, <p> element shifts above the <img> element and text is shown on top of the image.
- This example is well illustrated in the file [z-index-1.html](#). In this file, if we uncomment the styling of 'p', and then change the 'z-index' value of <img> to '2', a proper 'z-index' value comparison of both <img> and <p> elements will happen and because the 'z-index' of <img> is greater than that of <p> element, image will again shift on top of the text. But if we change the 'z-index' of <p> to '3', text will again stack on the top of image. For reference, you can visit [z-index-2.html](#) file.

114). To make the font size of an element one size smaller than the font size of the element's container, which style property would you apply?

A) font-size: reduce;

B) font-size: 8px;

C) font-size: -1em;

D) font-size: smaller;

- 'font-size: smaller' makes the font size of the element little smaller compared to that of the parent element.

115). Given this markup, which selector would result in the text being highlighted in yellow?

```
#BLM
```

A) .highlight { background-color: yellow; }

B) #highlight { background-color: yellow; }

C) .highlight { color: yellow; }

D) #highlight { color: yellow; }

116). To prevent a background image from tiling in any direction, which style property would you apply?

A) background-repeat: no-repeat;

B) background-repeat: fixed;

C) background-repeat: none;

D) background-tile: none;

117). To rotate an object 30 degrees counterclockwise, which style property would you apply?

A) transform: rotate(-30deg);

B) transform: rotate(30deg);

C) rotate: 30deg;

D) spin: 30deg;

118). Which style rule would you apply to set the background image to display the contents of the wood.png file?

A) background-image: file('wood.png');

B) background-image: url('wood.png');

C) background-image: wood.png;

D) image: wood.png

119). What style rule would set the font color of only paragraph two to blue?

```
<section><p>paragraph one</p></section><p>paragraph two</p>
```

A) section > p { color: blue; }

B) p { color: blue; }

C) section + p { color: blue; }

D) p + section { color: blue; }

120). You want to move an element up 100px. Which CSS property would you use?

A) transform: translateX(-100px)

B) transform: translateY(-100px)

C) transform: translateY(100px)

D) transform: translateX(100px)

121). Which style will horizontally center the inner <div> within the outer <div>?

```
<div id="outer">
```

```
 <div id="inner">Center Me!</div>
```

```
</div>
```

A) #inner { width: 50%; } #outer { width: 100%; }

B) #inner { left: 0; right: 0; position: center; }

C) #inner { text-align: center; }



D) #inner { width: 50%; margin: 0 auto; }

- The property 'margin: auto' will automatically adjust the element's margin in such a way that the element floats to the right-most part of its parent element's width.
- In the file centering.html, if we give 'margin-left: auto' to #outer element, the #outer element will float to the right side of the document body.
- Similarly in #inner element if we give width of 50% and 'margin: auto', the text inside the element will float to the right of the #inner element. The property 'text-align: center' will automatically place the text inside the #inner element to the center.

122). Which corner will the puppy be in when these CSS rules are applied?

```
.pen {
 height: 100px;
 width: 100px;
 border: 2px dashed brown;
 position: relative;
}

#puppy {
 position: absolute;
 right: 80px;
 bottom: 0px;
}

<div class="pen">

</div>
```

- A) top-right corner
- B) bottom-right corner
- C) top-left corner
- D) bottom-left corner

123). Which choice uses the correct syntax for adding a hover pseudo class to element?

- A) a:hover {...}
- B) a :hover {...}
- C) a.hover {...}
- D) a hover {...}

124). Which missing code will give "Cellar Door" a shadow?

```
<style>
 #cellar-door {
 box-shadow: 3px 5px 10px #000;
 }
 .text-shadow {
 text-shadow: 3px 5px 10px #000;
 }
</style>
```

```
<h1 _____> Cellar Door</h1>
```

A) class="text-shadow"

B) id="cellar-door"

C) id="text-shadow"

D) class="cellar-door"

- The property 'box-shadow' applies to <div> elements and 'text-shadow' applies to text element.

125). Which choice is a valid example of a comment in CSS?

A) -- This line has been cancelled.

B) /\* This line has been cancelled. \*/

C) // This line has been cancelled.

D) # This line has been cancelled.

126). Which element(s) will be displayed in blue text?

```
h2 ~ p { color: blue; }
```

```
<section>
 <p>P1</p>
 <h2>H2</h2>
 <p>P3</p>
 <p>P4</p>
</section>
```

- A) P3
- B) P1, P3 and P4
- C) P3 and P4
- D) P1

127). When these pseudo class selectors are applied to a link, what states will the styles be applied to?

a:visited { ...; }

a:active { ...; }

a:hover { ...; }

a:focus { ...; }

A) :visited styles are applied after the link has been opened. :active styles are applied on mouse click and hold. :hover styles are applied on mouse over or mouse click and hold. :focus styles are applied when a Tab key on a keyboard is used to navigate through links.

B) :visited styles are applied after the link has been opened. :active styles are applied on mouse click. :hover styles are applied on mouse over. :focus styles are applied on mouse click and hold, or when a Tab key on a keyboard is used to navigate through links.

C) :visited styles are applied after the link has been opened. :active styles are applied on mouse click only and before mouse release. :hover styles are applied on mouse over. :focus styles are applied when a Tab key on a keyboard is used to navigate through links.

D) :visited styles are applied on mouse hold. :active styles are applied when the link has been opened. :hover styles are applied on mouse over. :focus styles are applied when a Tab key on a keyboard is used to navigate through links.