

CSS/CSS3

Three ways to insert css:

(1) External style sheet

(2) Internal style sheet

(3) Inline style

	Priority (cascade)	Performance(generally)	Amount of work (generally)	Development & maintenance
External	low	Reduce the rendering time from server	Little(can be reused in unlimited pages)	Swap between css file and html file. But easy to maintenance.
Internal	high	Middle	more (can be reused in one page)	middle
Inline	highest	No look up time	Most (can not be reused at all)	Easy to develop and test

Part 1: basic CSS

1. What is the use of column layout in CSS?

CSS column layout helps you to divide your text in to columns.

2. ID vs Classes

- ID's are unique
 - Each element can have only one ID
 - Each page can have only one element with that ID
- Classes are NOT unique
 - You can use the same class on multiple elements.
 - You can use multiple classes on the same element.

3. When would you use CSS float?

Float is used when you want to make an element of your page (usually an image) be pushed to the right or left and make other elements wrap around it.

4. When would you use CSS clear?

When you want an element on the left or right of the floating element not to wrap around it, you can use clear.

5. What are Sass, LESS, and Stylus? Why do people use them? How does something like Compass relate to Sass?

They are CSS preprocessors. They are an abstraction layer on top of CSS. They are a special syntax/language that compile down into CSS. They make managing CSS easier, with things like variables to handle vendor prefixes (among other things). They make doing best practices easier, like concatenating and compressing CSS.

6. CSS positioning

- Fixed positioning: An element with fixed position is positioned relative to the browser window.
- Relative positioning: A relative positioned element is positioned relative to its normal position
- Absolute positioning: An absolute position element is positioned relative to the first parent element that has a position other than static.
- Z-index: The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

7. CSS specificity

- Equal specificity: the latest rule is the one that counts.
- Unequal specificity: the more specific rule is the one that counts.

8. !important

- ```
2. p {
3. color: red !important;
```

```
4. }
5. #thing {
6. color: green;
7. }
8. <p id="thing">Will be RED.</p>
```

The paragraph is will be red, even though the ID selector has higher specificity. The **!important** rule overrides that particular property.

#### 10. How to make an element center

Margin: auto;

#### 11. What are sprites and why would use them? How do you go about creating them?

##### What are possible alternatives to sprites?

To enhance the performance, we always reduce the resource request.

An image sprite is a collection of images put into a single image.

A web page with many images can take a long time to load and generates multiple server requests.

Using image sprites will reduce the number of server requests and save bandwidth.

Generally speaking, the slowest thing a website can do is request a resource. The fewer requests a site needs to make, the faster it is.

Manually creating sprites is certainly a possibility but it isn't very efficient. There are helper tools like SpriteCow and SpriteMe, Spriting with Compass, or Grunticon. It's always interesting to hear a real workflow approach.

#### 12. Explain the difference between visibility:hidden and display:none

- visibility:hidden simply hides the element, while it will still take up space and affect the layout of the document.
- display:none also hides the element, but will not take up space and the page will appear as if the element is not present.

#### 13. What is the purpose of the z-index and how is it used?

- The z-index property specifies the stack order of an element.
- An element with greater stack order is always in front of an element with a lower stack order.
- z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).

#### 14. What are child selectors and Descendant selectors

A child selector is used when you want to match an element that is the child of another specific element. The parent and child selectors are separated by >(greater-than sign).

A descendant selector is used when you want to match an element that is the descendant of another specific element. The two selectors are separated by space.

## 15. What are pseudo classes?

Pseudo classes allow you to identify HTML elements on characteristics (as opposed to their name or attributes). The classes are specified using a colon to separate the element name and pseudo class. (A good example is the `:link` and `:visited` pseudo classes for the HTML `A` element. Another good example is `first-child`, which finds an element's first child element.)

## 16. Make sure you know other selectors: sibling selectors, attribute selectors and the combination of selectors.

# 1. Examples of selectors and classes:

CSS selectors are used to select HTML elements.

The selectors decide which elements in the DOM tree should follow the CSS rules.

- 1) A **selector** is the element that is linked to a particular style. E.g. `p` in

```
p { padding: 10px; }
```

- 2) A **class selector** is a selector that uses a defined class (multiple per page).

E.g. `p.section` in

```
p.section { padding: 10px; }
```

- 3) An **ID selector** is a selector that uses an individually assigned identifier (one per page). E.g. `p#section` in

```
CSS: #section { padding: 10px; }
```

```
(X)HTML: <p id="section">Text</>
```

- 4) A **contextual selector** is a selector that defines a precise cascading order for the rule.

E.g. `p span` in

```
p span { font-style: italic; }
```

defines that all span-elements within a p-element should be styled in *italics*.

- 5) An **attribute selector** matches elements which have a specific attribute or its value.

E.g. `p span` in

```
p[title] { font-weight: bold; }
```

matches all p-elements which have a `title` attribute.

- 6) **Pseudo-classes** are special classes that are used to define the behavior of HTML elements. They are used to add special effects to some selectors, which are applied automatically in certain states. E.g. `:visited` in

```
a:visited { text-decoration: underline; }
```

- 7) **Pseudo-elements** provide designers a way to assign style to content that does not exist in the source document. Pseudo-element is a specific, unique part of an element that can be used to generate content “on the fly”, automatic numbering and lists. E.g. `:first-line` or `:after` in

```
p:first-line { font-variant: small-caps; }
```

```
a:link:after { content: " (" attr(href) ")"; }
```

## Part 2: CSS3

### 1. What is the difference between CSS and CSS3 ?

Answers : 1 CSS3 is upgraded version of CSS with new features like **Selectors, Box Model, Backgrounds and Borders, Text Effects, 2D/3D Transformations, Animations, Multiple Column Layout, User Interface** etc

### 2. CSS Box-model

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content. The box model allows us to place a border around elements and space elements in relation to other elements.



**Content** - The content of the box, where text and images appear

**Padding** - Clears an area around the content. The padding is transparent

**Border** - A border that goes around the padding and content

**Margin** - Clears an area outside the border. The margin is transparent

### 3. **What new futures added in CSS3 for Borders?**

Answer : 3 following border futures added

- border-radius
- box-shadow
- border-image

### 4. **CSS3 background**

In CSS3 it is possible to specify the size of the background image, which allows you to re-use background images in different ways.

In CSS3 we can also use multiple background, we can insert like this:

```
background-image: url(decoration.png), url(ribbon.png), url(old_paper.jpg);
```

### 5. **What is the word wrap / word wrapping in CSS3 ?**

To allow long words to be able to break and wrap onto the next line in css3 we used word-wrap property like below class

```
.wordwrappcds{ word-wrap:break-word;}
```

### 6. **What is the CSS3 animation ?**

CSS3 animations allows animation of most HTML elements without using JavaScript or Flash!

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS3 animation, you must first specify some keyframes for the animation.


Keyframes hold what styles the element will have at certain times.

When the animation is created in the @keyframe, bind it to a selector, otherwise the animation will have no effect.

Bind the animation to a selector by specifying at least these two CSS3 animation properties:

- Specify the name of the animation
- Specify the duration of the animation

## 7. What are the properties Associated with CSS3 Multi-Column Feature?

By using the Multi-Column feature of CSS3 the web  designers can get their output in multiple columns. The properties associated with multi-columns would be column-count, column-gap and column-rule.

column-count : used to specify the number of columns,

column-gap : used to specify the width of the gap between columns and

column-rule : used to specify the width, style and color of the rule between columns.

## 8. How to use an custom font?

In @font-face, using url to define the font resource and define a font-family. We can use that font by adding font-family into rules.

## 9. Should be familiar with other CSS3 new features

## 2. Link

```
a:link {color:#FF0000;} /* unvisited link */
```

```
a:visited {color: rgb(255,0,0);} /* visited link */
```

```
a:hover {color: tan;} /* mouse over link */
```

```
a:active {color:#0000FF;} /* selected link */
```

**Order is important.**

<hr>: thematic break. Attributes are not supported in HTML5, use CSS instead.

**text-decoration:** none |underline |overline |line-through |initial |inherit;

**List:**

```
ul.circle {list-style-type:circle}
```

```
ul.square {list-style-type:square}
```

```
ol.upper-roman {list-style-type:upper-roman}
```

```
ol.lower-alpha {list-style-type:lower-alpha}
```

```
ul {list-style-image:url('sqpurple.gif');}
```

```
ul li{
```

```
background-image:url(sqpurple.gif);
```

```
background-repeat:no-repeat;
```

```
background-position:0px 5px;
```

```
padding-left:14px;
```

```
}
```

list-style-position: inside |outside (specifies if the list-item markers should appear inside or outside the content flow.)



## Part 1. LESS and SASS

### 1. What is LESS?

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themeable and extendable.

```
1. @base: #f938ab; /* variable */
2.
3. /* function */
4. .box-shadow(@style, @c) when (iscolor(@c)) {
5. -webkit-box-shadow: @style @c;
6. box-shadow: @style @c;
7. }
8. .box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
9. .box-shadow(@style, rgba(0, 0, 0, @alpha));
10. }
11. .box {
12. color: saturate(@base, 5%);
13. border-color: lighten(@base, 30%);
14. div { .box-shadow(0 0 5px, 30%) } /* mix in */
15. }
```

### 2. How to use LESS?

- install less through npm
- define the less file
- compile with lessc to generate css file

```
1. npm install -g less
2. lessc styles.less styles.css
```

Also, you can use it directly in web page, which is acceptable but not prefer.

```
1. <link rel="stylesheet/less" type="text/css" href="styles.less" />
2. <script src="less.js" type="text/javascript"></script>
```

### 3. What are the advantages/disadvantages of using CSS preprocessors? (SASS, Compass, Stylus, LESS)

- **Benefits:** Ability to use nested syntax, define variables and mix-in, use of mathematical and operational functions, and the ability to join multiple files into a single one.
- **Disadvantages:** Difficulties tracking file size, maintenance and updating issues, difficulties debugging.

## Part 3: Responsive Design

### 10. What is responsive design all about?

Responsive design is about taking one code base and making it work for all of different devices with different sizes and different capabilities. Part of that is *media queries* and different visuals. Part of that is *different resources* (e.g. different JavaScript to handle touch vs click or different images to accommodate the screen).

### 11. How to create a responsive design page?

Define page as above question, or using Bootstrap grid system like below. The total column number is 12. With grid system, you can define the different layout by different device width. Also, grid system support nested, offset or push/pull.

```
3. <div class="row">
4. <div class="col-lg-2 col-md-3 col-sm-6 col-xs-12"></div>
5. </div>
```

```
6. /* Extra small devices (phones, less than 768px) */
7. /* No media query since this is the default in Bootstrap */
8. /* Small devices (tablets, 768px and up) */
9. @media (min-width: @screen-sm-min) { ... }
10.
11. /* Medium devices (desktops, 992px and up) */
12. @media (min-width: @screen-md-min) { ... }
13.
14. /* Large devices (large desktops, 1200px and up) */
15. @media (min-width: @screen-lg-min) { ... }
```

### 12. How to do responsive design without bootstrap?

For different resolution and different media type, we need different the layouts or styles for some elements to make web user friendly.

- media queries
  - In the same CSS file, using @media to get the width of the screen to make the style.
  - In the <link> using media to get the width of the screen to select which file to load

- Choose what you like, framework usually use the first one.

#### **4. What are those components you have ever used?**

Navbar(without JavaScript dependence), Pagination, Modal, Tab, Tooltip, Popover, Alert, Collapse, Carousel.

Ui.bootstrap contains Carousel, Collapse, Datepicker, Dropdown, Modal, Pager, Pagination, Popover, Progressbar, Tabs, Timepicker, Tooltip.