

1. 语义化标签 & 为什么需要?

标签名是有实际意义的, 指明了该标签所包含的内容。

方便开发人员识别

Examples of **non-semantic** elements: `<div>` and `` - Tells nothing about its content.

Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

2. JS 引入方法?

2 种:

- 在 HTML 中 `<script type="text/javascript"></script>` 插入
- 写在 .js 文件中, 在 HTML 中通过 `<script src=".js"></script>` 引入

浏览器解释 html 时是按先后顺序的, 所以前面 的 script 就先被执行。在页面中 `<head>` 部分放置 `<script>` 元素, 浏览器解析 head 部分就会执行这个代码, 然后才解析页面的其余部分。放在 `<body>` 部分 JavaScript 代码在网页读取到该语句的时候就会执行。

3. CSS 引入方法?

4 种:

- 内联式 CSS 样式: 把 css 代码直接写在现有的 HTML 标签中, 如下面代码:

```
<p style="color:red">这里文字是红色。</p>
```

- 嵌入式 CSS 样式: 把 CSS 样式代码写在 `<style type="text/css"></style>` 标签之间, 并且一般情况下嵌入式 CSS 样式写在 `<head></head>` 之间。。如下面代码:

```
<style type="text/css">
  span{
    color:red;
  }
</style>
```

- 外部式/外联式 CSS 样式: 把 CSS 代码写一个单独的外部文件中, 这个 CSS 样式文件以 “.css” 为扩展名, 在 `<head>` 内使用 `<link>` 标签将 CSS 样式文件链接到 HTML 文件内, 如下面代码:

```
<link href="**base.css**" rel="stylesheet" type="text/css" />
```

- 导入式 CSS 样式: 使用 CSS 规则引入外部 CSS 文件

```
<style>
  @import url(style.css);
</style>
```

link 和 @import 的区别?

- link 是 XHTML 标签, 除了加载 CSS 外, 还可以定义 RSS 等其他事务; @import 属于 CSS 范畴, 只能加载 CSS。
- link 引用 CSS 时, 在页面载入时同时加载; @import 需要页面网页完全载入以后加载。所以会出现一开始没有 CSS 样式, 闪烁一下出现样式后的页面 (网速慢的情况下)
- link 是 XHTML 标签, 无兼容问题; @import 是在 CSS2.1 提出的, 低版本的浏览器不支持。
- link 支持使用 Javascript 控制 DOM 去改变样式; 而 @import 不支持。

4. CSS 选择器优先级?

内联样式 > id 选择器 > 类选择器 > 标签选择器 > 通配符选择器

5. CSS 属性继承?

可继承: font text list

不可继承: display 盒模型的 width/height/padding/border position

6. CSS 中百分比的使用?

- 宽和高在使用百分比值时, 其参照都是元素的包含块
- 对于 `margin` 和 `padding`, 其任意方向的百分比值, 参照都是包含块的宽度。
- `border radius`
- `background-position` 的百分比值, 取的参照是一个减法计算值, 由放置背景图的区域尺寸, 减去背景图的尺寸得到, 可以为负值
- `font-size` 直接参照父类
- `line-height`: 参照元素自身的 `font-size`
- `vertical-align`: 参照自身的 `line-height`
- 定位用的 `bottom`、`left`、`right`、`top`: 参照是元素的包含块。`left` 和 `right` 是参照包含块的宽度, `bottom` 和 `top` 是参照包含块的高度。
- `transform:translate`: 参照是变换的边界框的尺寸 (等于这个元素自己的 `border-box` 尺寸)。例如, 一个宽度为 `150px`, 高度为 `100px` 的元素, 定义 `transform:translate(50%, 50%)` 的效果是 `transform:translate(75px, 50px);`。

7. 伪类?

A CSS **pseudo-class** is a keyword added to a selector that specifies a special state of the selected element(s). For example, `:hover` can be used to change a button's color when the user's pointer hovers over it.

8. 伪元素?

A CSS **pseudo-element** is a keyword added to a selector that lets you style a specific part of the selected element(s). For example, `::first-line` can be used to change the font of the first line of a paragraph.

9. 竖排元素变横?

`display:inline;`

10. 让一个元素左/右对齐?

`flex:`

`justify-content: flex-start` 左对齐 `flex-end` 右对齐

`align-content: flex-start` 左对齐 `flex-end` 右对齐

`grid:`

`justify-items: start` 左对齐 `end` 右对齐

`justify-content: start` 左对齐 `end` 右对齐

`justify-self: start` 左对齐 `end` 右对齐

11. 让一个元素水平/垂直居中?

`flex:`

`justify-content: center`

`align-content: center`

`grid:`

`justify-items: center` 水平居中

align-items: center 垂直居中
justify-content: center 水平居中
align-content: center 垂直居中
justify-self: center 水平居中
align-self: center 垂直居中

12. CSS 的尺寸单位?

px(像素)、em、% 百分比

13. Sticky-footer

display:fixed;

14. media-query?

Media queries use conditional logic for applying CSS styling.

Media queries let you adapt your site or app depending on the presence or value of various device characteristics and parameters.

15. line-height V.S. height?

line-height: 行高

height: 内容的高度

16. display 属性?

- display:block:
 - 每个块级元素都从新的一行开始，并且其后的元素也另起一行。
 - 元素的高度、宽度、行高以及顶和底边距都可设置。
 - 元素宽度在不设置的情况下，是它本身父容器的 100%(和父元素的宽度一致)，除非设定一个宽度。
- display:inline:
 - 和其他元素都在一行上;
 - 元素的高度、宽度及顶部和底部边距不可设置;
 - 元素的宽度就是它包含的文字或图片的宽度，不可改变。
- display:inline-block:
 - 和其他元素都在一行上;
 - 元素的高度、宽度、行高以及顶和底边距都可设置。
- display:none: 设置此元素不会被显示。

17. position 属性?

- 绝对定位 (position: absolute): left、right、top、bottom 属性相对于其最接近的一个具有定位属性的父包含块，如果不存在这样的包含块，则相对于 body 元素，即相对于浏览器窗口。
- 相对定位 (position: relative): 相对于以前的位置移动，移动的方向和幅度由 left、right、top、bottom 属性确定。
- 固定定位 (position: fixed): 相对移动的坐标是视图 (屏幕内的网页窗口) 本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化。

18. viewport 标签?

viewport 是用户网页的可视区域。

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.