

# 饥人谷前端入门训练营

老师： 若愚

# 课程规划

## Part1 HTML基础

---

-  任务1 开发环境搭建
-  任务2 HTML简介
-  任务3 快速掌握HTML常见标签
-  任务4 掌握HTML5标签

## Part3 CSS应用

---

-  任务9 CSS3常见特效
-  任务10 Flex布局的使用
-  任务11 CSS中的定位
-  任务12 字体图标的使用

## Part2 CSS核心

---

-  任务5 CSS的介绍和引入
-  任务6 CSS常见选择器的使用
-  任务7 掌握盒模型的概念
-  任务8 CSS常见样式

## Part4 开发dribbble静态首页

---

-  任务13 页面分析和结构搭建
-  任务14 实现导航条和banner
-  任务15 实现列表区
-  任务16 实现页面footer

# Part 2. CSS核心

CSS引入

# CSS基本语法

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <style>
    p {
      color: red;
    }

    @media screen and (max-width: 700px) {
      p { color: blue; }
    }
  </style>
</head>
<body>
  <p>饥人谷</p>
</body>
</html>
```

# 样式两种语法

```
/*这是注释*/
```

```
选择器 {  
    属性: 值;  
    属性: 值;  
}
```

```
@关键字 {  
    其他  
}
```

注意:

空格和缩进建议保留

半角符号，不要写成全角

# 引入方式一

## 通过<link>标签引用

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" >
  <link rel="stylesheet" href="index.css">
</head>
<body>
  <h1>Hello CSS!</h1>
</body>
</html>
```

# 引入方式二

## 通过CSS语法@import引入

- ✓ 放在style标签或者CSS文件中
- ✓ @import 必须放在CSS文件开头
- ✓ 语法后面的分号必不可少

```
<style>
  @import url("index.css");
  @import url('index.css');
  @import url(index.css);
  @import 'custom.css';
  @import "common.css";
</style>
```

# 引入方式三

## 内部样式

- ✓ 将CSS放在页面<style>标签中
- ✓ style标签可以放在head里，也可放在body内任意地方

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <style>
    h1 {
      background: orange;
    }
  </style>
</head>
<body>
  <h1>Hello CSS!</h1>
</body>
</html>
```



# 引入方式四

## 内联样式

- ✓ 不推荐，但在某些情况下也有用

```
<p style="background: orange; font-size: 24px;">  
  CSS  
</p>
```

# Part 2. CSS核心

## CSS选择器

# class、id、属性

```
<body>  
  <button id="login" class="btn btn-big">登陆</button>  
  <button id="register" class="btn btn-small" disabled>注册</button>  
</body>
```

## 作用

- ✓ id 标记页面独一无二的元素(身份证号)
- ✓ class给元素分类(打标签)

# 选择器分类

## 元素选择器(标签选择器)

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

# 选择器分类

## ID选择器

```
<p id="box">饥人谷</p>

<style>
  #box {
    border: 1px solid red;
  }
</style>
```

# 选择器分类

## 类选择器

```
<p class="box active">饥人谷</p>
```

```
<style>
```

```
  .active {
```

```
    font-size: 20px;
```

```
  }
```

```
</style>
```

# 选择器分类

## 通用选择器

```
* {  
  box-sizing: border-box;  
}  
  
.box * {  
  font-size: 24px;  
}
```

# 选择器分类

## 属性选择器

- ✓ `[attr]` 选择包含 `attr` 属性的所有元素，不论 `attr` 的值为何。
- ✓ `[attr=val]` 仅选择 `attr` 属性被赋值为 `val` 的所有元素。

```
[disabled] {  
  border: 1px solid #ccc;  
}
```

```
[data-color="gray"] {  
  color: #666;  
}
```



# 组合选择器

A, B

- ✓ A, B 同时选中A 和 B

```
<p class="author">饥人谷</p>
<p class="detail">求知若饥 虚心似谷</p>

<style>
  .author, .detail {
    color: red;
  }
</style>
```

# 组合选择器

A B

- ✓ 选中A的后代 B 元素 (A和B间有个空格)

```
<ul class="detail">
  <li>
    <ul>
      <li>被选中✓</li>
    </ul>
  </li>
  <li>被选中✓</li>
</ul>
<style>
  .detail li {
    color: red;
  }
</style>
```

# 组合选择器

A > B

- ✓ 选中A的**直接子元素**B

```
<li>
  <ul>
    <li>不被选中</li>
  </ul>
</li>
<li>被选中✓</li>
</ul>
<style>
  .detail li {
    color: red;
  }
</style>
```

# Part 2. CSS核心

## 盒模型

# 代码案例

## 块级盒子和行级盒子的差异

```
<p>欢迎</p>
<p>来饥人谷学习</p>
<span>前端</span>
<span>Java</span>
<style>
  p {
    border: 1px solid black;
  }
  span {
    border: 1px solid red;
  }
</style>
```

欢迎

来饥人谷学习

前端 Java

# 块级盒子

## 特征

- ✓ 盒子会在水平方向上扩展并占据父容器在该方向上的所有可用空间
- ✓ 每个盒子都会换到新行
- ✓ width 和 height 属性有效
- ✓ 内边距 (padding), 外边距 (margin) 和 边框 (border) 会将其他元素从当前盒子周围推开
- ✓ 可通过 display 属性改变显示类型

## 常见的块级标签

- ✓ `<h1>...<h6>`、`<p>`、`<div>`、`<ul>`、`<ol>`、`<li>`、`<dl>`、`<dt>`、`<dd>`、`<header>`、`<main>`、`<footer>`、`<aside>`、`<nav>`、`<section>`、`<table>`、`<pre>`
- ✓ 更多: [https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements)

# 行级盒子

## 特征

- ✓ 盒子不会产生换行。
- ✓ width 和 height 属性将不起作用。
- ✓ 内边距、外边距以及边框会被应用但是不会把其他处于 inline 状态的盒子推开。
- ✓ 可通过 display 属性改变显示类型

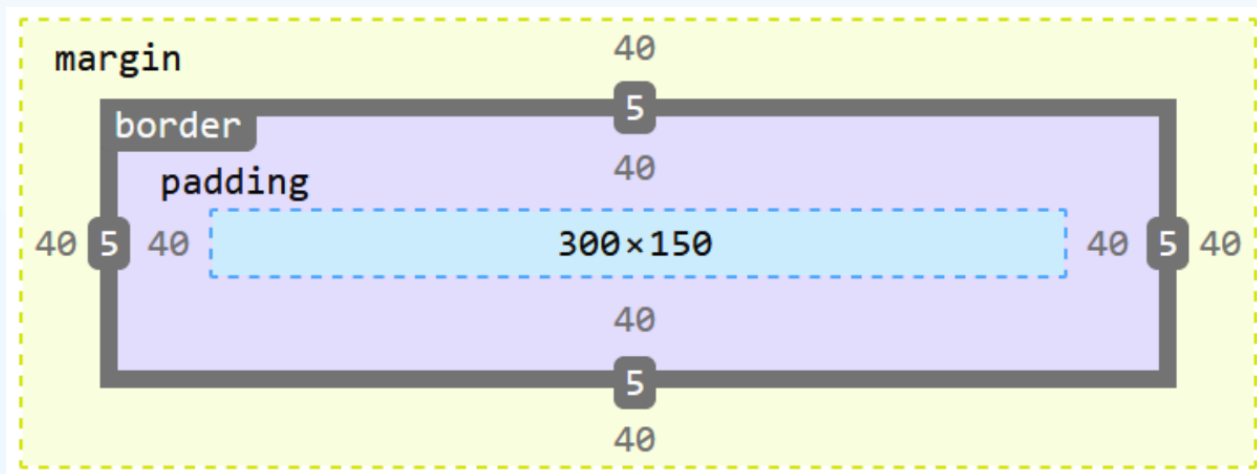
## 常见的行级标签

- ✓ <a>、<span>、<em>、<strong>、<img>、<input>、<label>、<textarea>、<select>、<button>、<code>...
- ✓ 更多行级元素参考 [https://developer.mozilla.org/en-US/docs/Web/HTML/Inline\\_elements#Elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements#Elements)

# 盒模型的各个部分

## 盒模型组成

- ✓Content: 这个区域是用来显示内容，大小可以通过设置 width 和 height.
- ✓Padding: 包围在内容区域外部的空白区域；大小通过 padding 相关属性设置。
- ✓Border: 边框盒包裹内容和内边距。大小通过 border 相关属性设置。
- ✓Margin: 这是最外面的区域，是盒子和其他元素之间的空白区域。大小通过 margin 相关属性设置。





# 标准盒模型

## 设置的width 和 height 指的是Content

- 宽度 =  $350 + 25 + 25 + 5 + 5$ ，高度 =  $150 + 25 + 25 + 5 + 5$
- margin不计入实际大小，但会影响盒子实际占用空间

```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```



# IE盒模型

设置的width和height包含内边距和边框

✓浏览器默认使用标准盒模型，如想使用IE盒模型可以设置box-sizing

```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  box-sizing: border-box; /* content-box */  
}
```



# 盒模型属性

## margin

✓ 外边距

```
.box {  
  margin: 10px;  
  margin: 10px 20px; /*10px 20px 10px 20px */  
  margin: 10px 20px 30px 40px; /*上右下左 */  
}  
  
.box {  
  margin: 0 auto; /* 0 auto 0 auto; 让块级盒子居中 */  
}  
  
.box {  
  margin-top: 10px;  
  margin-bottom: 20px;  
  margin-left: 30px;  
  margin-right: 40px;  
}
```

# 盒模型属性

## padding

✓ 内边距

```
.box {  
  padding: 10px;  
  padding: 10px 20px; /*10px 20px 10px 20px */  
  padding: 10px 20px 30px 40px; /*上右下左 */  
}  
  
.box {  
  padding-top: 10px;  
  padding-bottom: 20px;  
  padding-left: 30px;  
  padding-right: 40px;  
}
```

# 盒模型属性

## border

### ✓ 边框

```
.box {  
  border: 1px solid black; /* solid、dashed、dotted */  
  border-radius: 4px;  
  border-radius: 50%;  
}  
  
.box {  
  border-top: 10px solid #ccc;  
  border-bottom: 5px dashed yellow;  
  border-left: 10px solid transparent;  
  border-right: 1px dotted black;  
}
```

# Part 2. CSS核心

常见样式

# color举例

```
color: red;

color: #0f0;

color: #00ff00;

color: rgba( 34, 12, 64);

color: rgba( 34, 12, 64, 0.3);

/* 色相(0和360是红 120是绿 240是蓝) 饱和度(0%灰) 明度(0%是黑) */
color: hsl( 0, 100%, 50%);

color: hsla( 120, 100%, 64, 0.3);

color: currentColor;

/* 边框的颜色使用继承自父亲的color属性的颜色 */
border: 1px solid currentColor;
```

# font-size

## 文字大小

```
<div class='wrap'>  
  <div class='a'>饥人谷</div>  
</div>
```

```
<style>
```

```
.a { font-size: 28px; }
```

```
</style>
```



# font-family

- 设置字体

- ✓ `font-family: '微软雅黑', 'Microsoft YaHei', 'Source Han Sans', Helvetica, Arial, sans-serif;`
- ✓ 先找第一个字体，找不到再第二个，依次往后...
- ✓ 字体里包含空白字符最好要加引号（可以不加，但如果字体包含数字和其他特殊字符，一定要加引号）

```
font-family: Gill Sans Extrabold, sans-serif;  
font-family: 'Gill Sans Extrabold', sans-serif;  
font-family: 'Gill Sans Extrabold 2020', sans-serif;
```

/\* 以下都不对\*/

```
font-family: Gill Sans Extrabold 2020, sans-serif;  
font-family: test@foo, sans-serif;  
font-family: #POUND, sans-serif;
```

# font-style

- 用于设置文字是否以斜体显示

- ✓normal 正常

- ✓italic 斜体

- ✓oblique 模拟斜体

# font-weight

- 用于设置文字粗细

- ✓normal 正常

- ✓bold 粗体

- ✓lighter 比父元素字体细一级

- ✓bolder 比父元素字体粗一级

- ✓100-900 用于微调字体粗细

- 注意

- ✓设置字体的粗细还取决于字体库是否存在该粗细尺寸的字体的图形描述

- 匹配机制（了解）

- ✓400~500： 从400至500升序查找->从400开始降序往前查找->从500开始升序往后查找

- ✓小于400： 从设置值开始降序往前查找->从设置值开始升序往后查找

- ✓大于500： 从设置值开始升序往后查找->从设置值开始降序往前查找

# text-decoration

- 设置文字划线样式

- ✓none 取消下划线

- ✓underline 设置下划线

- ✓overline 上划线

- ✓line-through 中划线

```
<div class='a'>饥人谷a</div>  
<div class='b'>饥人谷b</div>  
<div class='c'>饥人谷c</div>  
<div class="d">饥人谷d</div>  
<a href='#'>饥人谷</a>
```

```
<style>  
a { text-decoration: none; }  
.a { text-decoration: underline; }  
.b { text-decoration: line-through; }  
.c { text-decoration: overline; }  
.d { text-decoration: green wavy  
underline; }  
</style>
```

饥人谷a  
~~饥人谷b~~  
饥人谷c  
饥人谷d  
饥人谷

# text-align

- 块级容器设置行内内容（例如文字）的对齐方式

- ✓left 左对齐

- ✓right 右对齐

- ✓center 居中

- ✓justify 文字向两侧对齐，对最后一行无效

- 注意

- ✓text-align 应用在块级容器上，对行内内容生效

# 省略文字

- 将文字设置为单行，超出展示...

```
p {  
  white-space: nowrap; /* 文字遇边界不折行，1行展示 */  
  overflow: hidden; /* 超出边界隐藏 */  
  text-overflow: ellipsis; /* 文本溢出时...展示 */  
}
```

- 将文字设置为多行，超出展示...

```
p {  
  overflow: hidden;  
  display: -webkit-box;  
  -webkit-line-clamp: 3;  
  -webkit-box-orient: vertical;  
}
```

# 其他

## •text-indent

✓设置首行文本的缩进

```
p {  
  text-indent: 2em; /* 2em是多少? */  
}
```

## •letter-spacing

✓设置文字/字母的间距

```
p {  
  letter-spacing: 3px;  
}
```

# list-style

- 修改列表样式

```
/*去掉默认的符号。这个常用*/
```

```
ul, ol {  
    list-style: none;  
}
```

```
/*列表方形符号*/
```

```
ul {  
    list-style: square;  
}
```

```
/*符号在盒模型内*/
```

```
ul {  
    list-style: inside; /*outside*/  
}
```

```
/*可以用图形符号*/
```

```
ul {  
    list-style: url(https://i.loli.net/2019/11/08/4vriKTQR6sXnFBW.png)  
}
```



# background

- 属性合并简写

```
.box{  
  background: #ccc url(images/bg.png) center no-repeat;  
}
```

- 多重背景

```
.box{  
  background: #ccc url(images/bg.png) center no-repeat,  
             #ccc url(images/bg2.png) 20px 20px no-repeat;  
}
```

# border

## •用法

✓border: 1px solid red;

- solid 实线
- dotted 点线
- dashed 虚线

```
.list {  
  border: 1px solid #ddd;  
}  
  
.list > .item {  
  border-bottom: 1px solid #ddd; /* 下边框 */  
}
```

# border-radius

## √设置圆角

```
.box{  
  border-radius: 4px;  
  border-radius: 4px 8px 16px 32px; /*左上 右上 右下 左下*/  
}
```

## √实现圆形

- 宽高相等，border-radius为宽高的一半以上

```
.box{  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
  border-radius: 50px;  
}
```

# border

## •border应用

### √实现三角形

```
.box {  
  width: 0;  
  height: 0;  
  border: 50px solid transparent;  
  border-radius: 50%;  
  border-top-color: red;  
}
```

### √实现半圆

```
.box {  
  width: 0;  
  height: 0;  
  border: 50px solid transparent;  
  border-radius: 50%;  
  border-top-color: red;  
  border-right-color: red;  
}
```

# table

## •border-collapse

✓用于设置表格边框是分开还是合并。

- collapse 合并
- separate 分开

```
table {  
  border-collapse: collapse;  
}
```