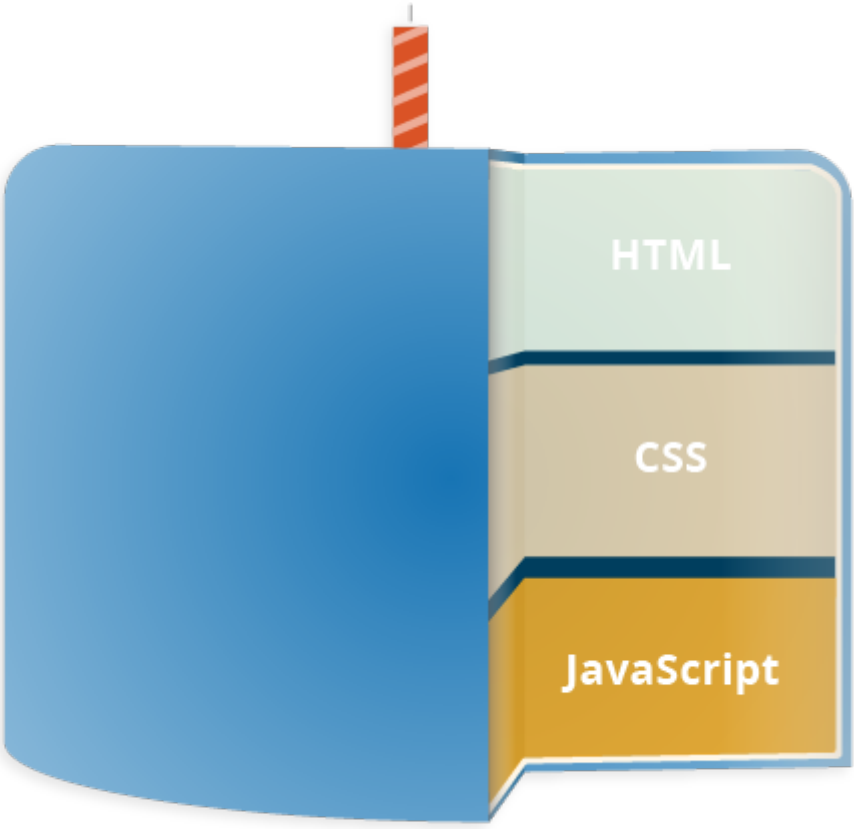
JavaScript

目标

- 基本了解 js 的常见语法
- 基本认识 js 的能力
- 可以独立完成一些简单功能开发

认识 JavaScript

JavaScript 是一种脚本，一门编程语言，它可以在网页上实现复杂的功能，网页展现给你的不再是简单的静态信息，而是实时的内容更新，交互式的地图，2D/3D 动画，滚动播放的视频等等。JavaScript 怎能缺席。它是标准 Web 技术蛋糕的第三层，其中 HTML 和 CSS 我们已经在学习中心的其他部分进行了详细的讲解。



- HTML是一种标记语言，用来结构化我们的网页内容并赋予内容含义，例如定义段落、标题和数据表，或在页面中嵌入图片和视频。
- CSS 是一种样式规则语言，可将样式应用于 HTML 内容， 例如设置背景颜色和字体，在多个列中布局内容。
- JavaScript 是一种脚本语言，可以用来创建动态更新的内容，控制多媒体，制作图像动画，还有很多。（好吧，虽然它不是万能的，但可以通过简短的代码来实现神奇的功能。）

这三层依次建立，秩序井然。以文本标签(text label)的简单示例。首先用 HTML 将文本标记起来，从而赋予它结构和目的：

```
<p>玩家1：小明</p>
```

然后我们可以为它加一点 CSS 让它更好看：

```
p {
  font-family: sans-serif, '黑体';
  letter-spacing: 1px;
  text-transform: uppercase;
  text-align: center;
  border: 2px solid rgba(0, 0, 200, 0.6);
  background: rgba(0, 0, 200, 0.3);
  color: rgba(0, 0, 200, 0.6);
  box-shadow: 1px 1px 2px rgba(0, 0, 200, 0.4);
  border-radius: 10px;
  padding: 3px 10px;
  display: inline-block;
  cursor: pointer;
}
```

最后，我们可以再加上一些 JavaScript 来实现动态行为：

```
const para = document.querySelector('p');

para.addEventListener('click', updateName);

function updateName() {
  let name = prompt('输入一个新的名字：');
  para.textContent = '玩家1：' + name;
}
```

JavaScript 能做的远不止这些。让我们来仔细探索。

它到底可以做什么？

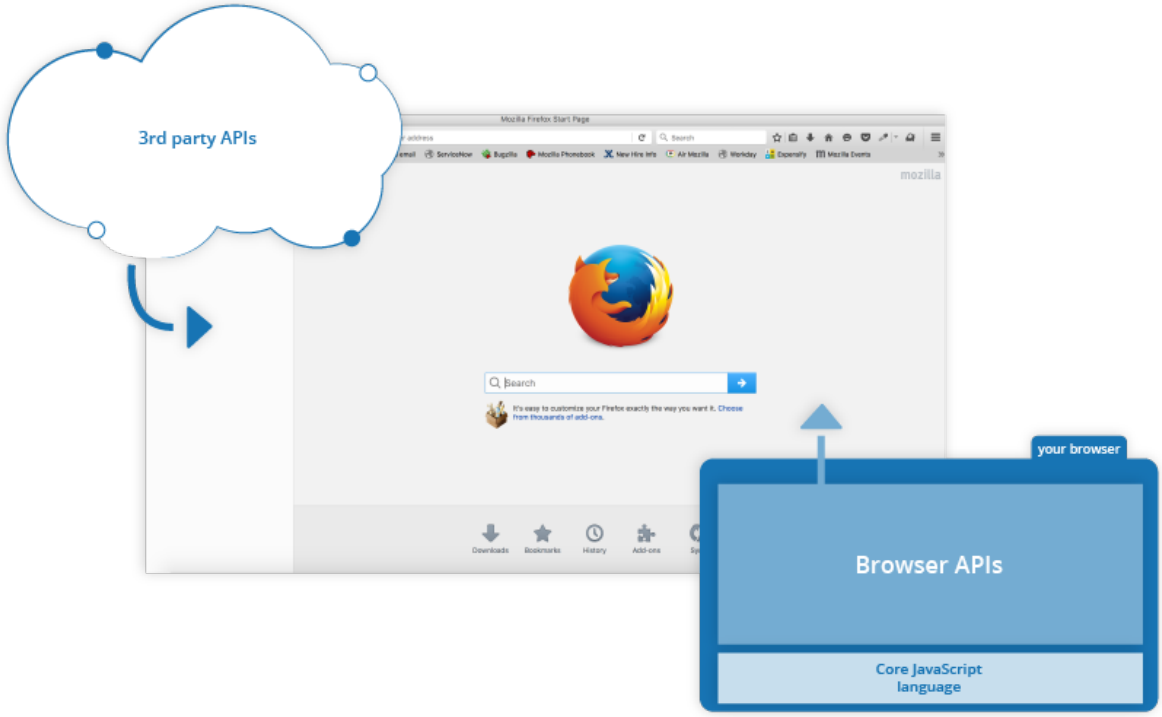
客户端（client-side）JavaScript 语言的核心包含一些普遍的编程特性，以让你可以做到如下的事情：

- 在变量中储存有用的值。比如上文的示例中，我们请求客户输入一个新名字，然后将其储存到 `name` 变量中。
- 操作一段文本（在编程中称为“字符串”（string））。上文的示例中，我们取字符串 "玩家1：" ，然后把它和 `name` 变量连结起来，创造出完整的文本标签，比如："玩家1：小明"。
- 运行代码以响应网页中发生的特定事件。上文的示例中，我们用一个 `click` 事件来检测按钮什么时候被点击，然后运行代码更新文本标签。
- 以及更多！

JavaScript 语言核心之上所构建的功能更令人兴奋。**应用程序接口（Application Programming Interfaces（API））**将为你的代码提供额外的超能力。

API 是已经建立好的一套代码组件，可以让开发者实现原本很难甚至无法实现的程序。就像现成的家具套件之于家居建设，用一些已经切好的木板组装一个书柜，显然比自己设计，寻找合适的木材，裁切至合适的尺寸和形状，找到正确尺寸的螺钉，再组装成书柜要简单得多。

API 通常分为两类。



浏览器 API 内建于 web 浏览器中，它们可以将数据从周边计算机环境中筛选出来，还可以做实用的复杂工作。例如：

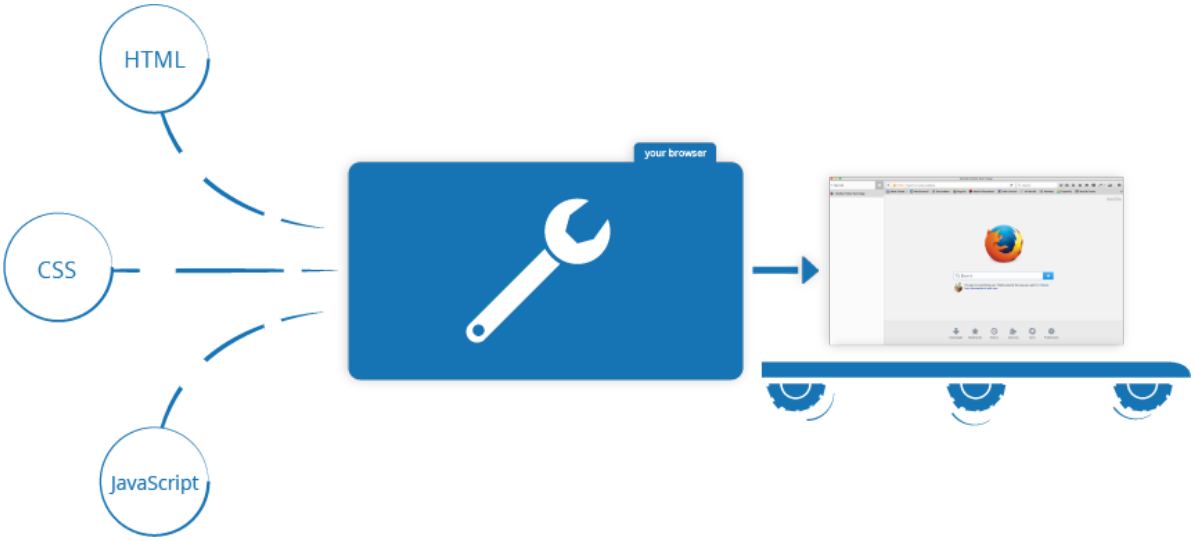
- `文档对象模型 API（DOM（Document Object Model）API）` 能通过创建、移除和修改 HTML，为页面动态应用新样式等手段来操作 HTML 和 CSS。比如当某个页面出现了一个弹窗，或者显示了一些新内容（像上文小 demo 中看到那样），这就是 DOM 在运行。
- `地理位置 API（Geolocation API）` 获取地理信息。这就是为什么 谷歌地图 可以找到你的位置，而且标示在地图上。
- `画布（Canvas）` 和 `WebGL` API 可以创建生动的 2D 和 3D 图像。人们正运用这些 web 技术制作令人惊叹的作品。参见 Chrome Experiments 以及 webglsamples。
- 诸如 `HTMLMediaElement` 和 `WebRTC` 等 影音类 API 让你可以利用多媒体做一些非常有趣的事，比如在网页中直接播放音乐和影片，或用自己的网络摄像头获取录像，然后在其他人的电脑上展示（试用简易版 截图 demo 以理解这个概念）。

第三方 API 并没有默认嵌入浏览器中，一般要从网上取得它们的代码和信息。比如：

- Twitter API、新浪微博 API 可以在网站上展示最新推文之类。
- 谷歌地图 API、高德地图 API 可以在网站嵌入定制的地图等等。

JavaScript 在页面上做了什么？现在我们实实在在的学习一些代码，与此同时，探索 JavaScript 运行时背后发生的事情。

让我们简单回顾一下，浏览器在读取一个网页时都发生什么（CSS 如何工作 一文中首次谈及）。浏览器在读取一个网页时，代码（HTML, CSS 和 JavaScript）将在一个运行环境（浏览器标签页）中得到执行。就像一间工厂，将原材料（代码）加工为一件产品（网页）。



在 HTML 和 CSS 集合组装成一个网页后，浏览器的 JavaScript 引擎将执行 JavaScript 代码。这保证了当 JavaScript 开始运行之前，网页的结构和样式已经就位。

这样很好，因为JavaScript 最普遍的用处是通过 DOM API（见上文）动态修改 HTML 和 CSS 来更新用户界面 （user interface）。如果 JavaScript 在 HTML 和 CSS 就位之前加载运行，就会引发错误。

JavaScript 运行次序

当浏览器执行到一段 JavaScript 代码时，通常会按从上往下的顺序执行这段代码。这意味着你需要注意代码书写的顺序。比如，我们回到第一个例子中的 JavaScript 代码：这里我们选定一个文本段落（第 1 行），然后给它附上一个事件监听器（第 3 行），使得在它被点击时，`updateName()` 代码块（code block） （5 – 8 行）便会运行。`updateName()` （这类可以重复使用的代码块称为“函数”）向用户请求一个新名字，然后把这个名字插入到段落中以更新显示。

如果你互换了代码里最初两行的顺序，会导致问题。浏览器开发者控制台将返回一个错误：`TypeError: para is undefined`。这意味着 `para` 对象还不存在，所以我们不能为它增添一个事件监听器。

服务器端代码 vs 客户端代码

你或许还听说过**服务器端（server-side）**和 **客户端（client-side）**代码这两个术语，尤其是在web开发时。客户端代码是在用户的电脑上运行的代码，在浏览一个网页时，它的客户端代码就会被下载，然后由浏览器来运行并展示。这就是**客户端 JavaScript。**

而服务器端代码在服务器上运行，接着运行结果才由浏览器下载并展示出来。我们之后会学习使用 Java 语言作为服务器端代码。

怎样向页面添加 Javascript

可以像添加 CSS 那样将 JavaScript 添加到 HTML 页面中。CSS 使用 `<link>` 元素链接外部样式表，使用 `<style>` 元素向 HTML 嵌入内部样式表，JavaScript 这里只需一个元素——`<script>`。我们来看看它是怎么工作的。

内部 Javascript

外部 Javascript

内联 Javascript

示例-猜数字游戏

随课堂讲解

进阶资料

JavaScript