



# Intro to Web Design

## 网页设计简介

### Intro to Web Conversation

### Web 对话简介

0:00 / 35:03

## Keypoints 关键点

## Course Plan 课程计划

- Review Course Outline and Class Plan.

查看课程大纲和课程计划。

- Classroom policies 课堂政策
- Expectations 期望值
- Contact Info 联系信息
- Course Overview 课程概述
- Project, Exams, Assignments

项目、考试、作业

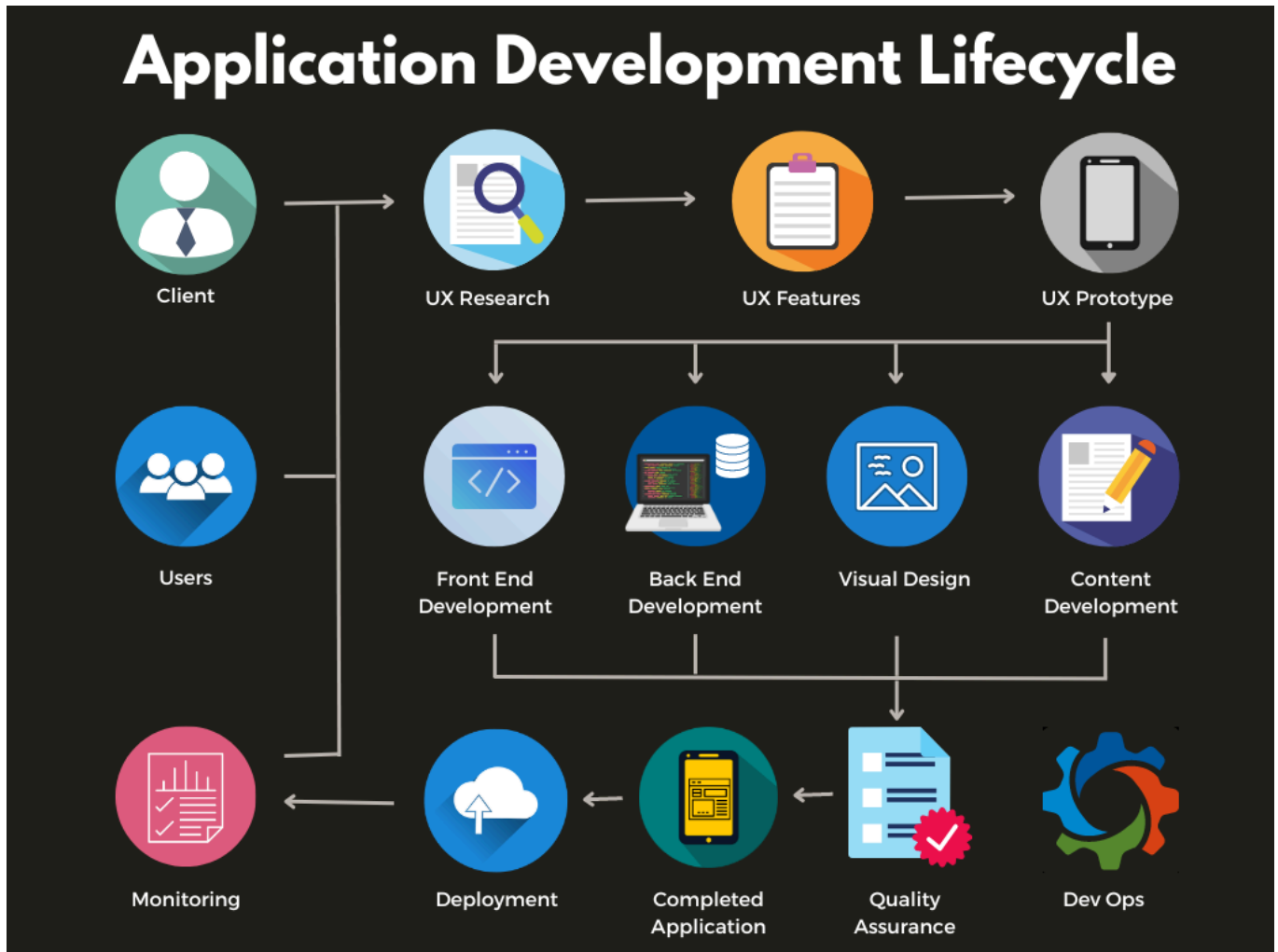
At the end of this class, students should be able to:

在本课程结束时，学生应该能够：

- Describe a general web development lifecycle  
描述一般 Web 开发生命周期
- Explain how the browser, device and resolution landscape affects web workers  
解释浏览器、设备和分辨率环境如何影响 Web 工作人员

# Web Development Lifecycle

## Web 开发生命周期



### Application Development Lifecycle

### 应用程序开发生命周期

While UX researchers lead the analysis phase with the client, senior designers and developers, Quality Assurance, and Project Managers should be included in the initial analysis so that work plans, feasibility, costs, team requirements, hardware, and software requirements can be established.

虽然用户体验研究人员与客户一起领导分析阶段，但高级设计师和开发人员、质量保证和项目经理应包括在初始分析中，以便建立工作计划、可行性、成本、团队要求、硬件和软件要求。

## UX Research and Features

## 用户体验研究和功能

The relationship between a client and a UX (User Experience) researcher is an essential and collaborative partnership that revolves around understanding and improving the user experience of a product or service. UX researchers are responsible for gathering insights about user behaviors, needs, and preferences, which then inform design and development decisions made by the client.

客户和 UX（用户体验）研究人员之间的关系是一种重要的协作伙伴关系，围绕着理解和改善产品或服务的用户体验。用户体验研究人员负责收集有关用户行为、需求和偏好的见解，然后为客户做出的设计和开发决策提供信息。

## Client's Goals and Objectives

### 客户的目标和目的

The client is typically a business, organization, or project team that seeks to create, improve, or optimize a product or service. They have specific goals and objectives, such as increasing user engagement, improving customer satisfaction, or enhancing the usability of their offerings. In addition to business requirements, the client may also have brand requirements that they must adhere to. In certain instances, business needs and user-centered goals may conflict with each other.

客户通常是寻求创建、改进或优化产品或服务的企业、组织或项目团队。他们有特定的目标和目标，例如提高用户参与度、提高客户满意度或增强其产品的可用性。除了业务需求外，客户可能还有他们必须遵守的品牌要求。在某些情况下，业务需求和以用户为中心的目标可能会相互冲突。

## Role of UX Researcher

### 用户体验研究员的角色

The UX researcher is an expert in studying and analyzing user behavior, attitudes, and interactions with a product or service. They conduct various research methods, such as user interviews, surveys, usability testing, and data analysis (including market research), to gather insights into how users perceive and interact with the client's offering.

用户体验研究员是研究和分析用户行为、态度以及与产品或服务互动的专家。他们进行各种研究方法，例如用户访谈、调查、可用性测试和数据分析（包括市场研究），以收集有关用户如何看待客户产品以及与客户产品互动的见解。

## Data Analysis and Insight Generation

### 数据分析和洞察生成

After gathering data, the UX researcher analyzes it to uncover patterns, trends, and user insights. They identify pain points, challenges, and opportunities for improvement in the user experience. The insights gained from research help the client make informed design

and development decisions.

收集数据后，用户体验研究人员对其进行分析，以发现模式、趋势和用户洞察。他们确定了用户体验的痛点、挑战和改进的机会。从研究中获得见解有助于客户做出明智的设计和开发决策。

## Recommendations, Feedback, and Iteration

### 建议、反馈和迭代

Based on the research findings, the UX researcher provides recommendations to the client. These recommendations may include the features of the app, user interface, and design recommendations. The relationship is iterative, with ongoing feedback and collaboration. The client provides feedback on the researcher's insights and recommendations, and adjustments are made as needed. This iterative process continues throughout the product's lifecycle.

根据研究结果，用户体验研究员向客户提供建议。这些建议可能包括应用程序的功能、用户界面和设计建议。这种关系是迭代的，需要持续的反馈和协作。客户对研究人员的见解和建议提供反馈，并根据需要进行调整。此迭代过程在产品的整个生命周期中持续进行。

## Measuring Success 衡量成功

The success of the relationship is measured by the extent to which the research insights positively impact the user experience and align with the client's goals. Increased user satisfaction, improved usability, enhanced engagement and increased revenue are some of the key indicators of success.

关系的成功与否是通过研究见解对用户体验产生积极影响并与客户目标保持一致的程度来衡量的。提高用户满意度、提高可用性、增强参与度和增加收入是成功的一些关键指标。

## UX Protoype 用户体验原型

Now that features have been established, UX Designers will now create wireframes or interactive prototypes so that all stakeholders will be able to see and navigate through the application. For the first time, stakeholders will be able to explore a more concrete vision of the application.

现在功能已经建立，UX 设计师现在将创建线框图或交互式原型，以便所有利益相关者都能够查看和浏览应用程序。利益相关者将首次能够探索应用程序的更具体愿景。

## Identify Key User Flows

### 确定关键用户流

Based on the requirements, the UX designer identifies the key user flows or pathways that users will take through the product. This helps in determining the structure and hierarchy of content and features. The goal of the UX designer is to create screens and interactions so that users would be able to perform specific tasks.

根据需求，用户体验设计人员确定用户将通过产品的关键用户流或路径。这有助于确定内容和功能的结构和层次结构。UX 设计器的目标是创建屏幕和交互，以使用户能够执行特定任务。

## Sketching Ideas 素描想法

Before moving to digital tools, the designer often starts with rough sketches on paper or a whiteboard. This quick and informal brainstorming helps in exploring different layout possibilities.

在转向数字工具之前，设计师通常会从纸上或白板上的粗略草图开始。这种快速和非正式的头脑风暴有助于探索不同的布局可能性。

## Choosing Wireframing Tools

### 选择线框图工具

The UX designer selects a wireframing tool to create digital representations of the sketches. Popular tools include [Axure RP](#), [Balsamiq](#), [Sketch](#), and [Figma](#). As at the time of this writing, Adobe has started proceedings to acquire Figma for USD \$20 billion and is moving Adobe XD toward end-of-life. Recently, Figma has become the de facto wireframing tool in industry. Historically, wireframes are low-fidelity visual representations of a digital product's layout, structure, and functionality. However, modern tools, such as Figma, enable UX designers to create high-fidelity wireframes which look and feel closer to the final application. It must be noted, however, that the purpose of wireframes is to help to outline the basic design elements and interactions without getting into detailed visual aesthetics.

UX 设计师选择线框图工具来创建数字草图的表示。流行的工具包括 [迅捷 RP](#)、[香蕉](#)、[草图](#)，以及 [图玛](#)。在撰写本文时，Adobe 已开始处理以 200 亿美元收购 Figma，并将 Adobe XD 推向生命周期结束。最近，Figma 已成为事实上的线框图工具工业。从历史上看，线框图是低保真视觉表示数字产品的布局、结构和功能。然而 Figma 等现代工具使 UX 设计师能够创建高保真度线框，外观和感觉更接近最终应用。一定是然而，请注意，线框图的原则是为了帮助勾勒出基本设计元素和交互，无需详细介绍视觉美学。

## Creating Basic Layouts 创建基本布局

The designer begins by creating basic layout structures for different screens or pages. These layouts define the placement of elements such as headers, navigation menus,

content sections, and footers.

设计器首先为不同的屏幕或页面创建基本布局结构。这些布局定义了页眉、导航菜单、内容部分和页脚等元素的位置。

Based on the requirements, the designer adds placeholders for content (text, images, videos) and key features (buttons, forms, interactive elements). The emphasis is on functionality rather than visual design at this stage.

根据需求，设计人员为内容（文本、图像、视频）和关键功能（按钮、表单、交互元素）添加占位符。现阶段的重点是功能而不是视觉设计。

The UX designer outlines how users will interact with the product by adding basic interactions and navigation elements. This might include clickable links, buttons, and simple animations

UX 设计器通过添加基本交互和导航元素来概述用户将如何与产品交互。这可能包括可点击的链接、按钮和简单的动画

## Annotations and Notes 注释和注释

The designer adds annotations or notes to the wireframes to explain functionality, user interactions, and any specific design decisions. This helps in communicating the intended user experience to the development team. While wireframes do a great job of visually representing the app, complex features such as voice-to-text, chat simulation and AI features may be difficult to simulate. Therefore, annotations and notes are invaluable in addressing these shortcomings.

设计人员在线框图上添加注释或注释，以解释功能、用户交互和任何特定的设计决策。这有助于将预期的用户体验传达给开发团队。虽然线框图在直观地表示应用程序方面做得很好，但语音转文本、聊天模拟和人工智能功能等复杂功能可能难以模拟。因此，注释和注释对于解决这些缺点非常宝贵。

## Refinement and Finalization

### 细化和最终确定

The wireframes are shared with relevant stakeholders, including developers, product managers, and clients. Feedback is gathered to ensure that the wireframes accurately represent the project's requirements and goals.

线框图与相关利益相关者共享，包括开发人员、产品经理和客户。收集反馈以确保线框准确代表项目的要求和目标。

Based on feedback, the designer refines the wireframes, making necessary adjustments to layout, interactions, and content placement.



根据反馈，设计人员对线框进行细化，对布局、交互和内容放置进行必要的调整。

Once the wireframes are finalized, they are handed off to the content development team, graphic designers, and coders. The wireframes serve as a blueprint for building the user interface and interactions, as well as more complex functional and data-oriented requirements.

线框最终确定后，它们将移交给内容开发团队、图形设计师和编码人员。线框图可作为构建用户界面和交互以及更复杂的功能和面向数据的需求的蓝图。

## Production 生产

### Content 内容

Content development refers to the process of creating, planning, producing, and optimizing various types of content to meet specific goals and objectives. Content can include written text, images, videos, audio, and any other form of media that is designed to inform, engage, and resonate with a target audience. Content development is a key component of various fields, including marketing, education, entertainment, and digital communication.

内容开发是指创建、规划、制作和优化各种类型的内容以满足特定目标的过程。内容可以包括书面文本、图像、视频、音频和任何其他形式的媒体，旨在告知、吸引目标受众并引起目标受众的共鸣。内容开发是营销、教育、娱乐和数字传播等各个领域的关键组成部分。

### Visual Design 视觉设计

Visual Designers usually have 2 main roles in the creation of an app:

视觉设计师在创建应用程序时通常有 2 个主要角色：

- **Design Elements.** Wireframes focus on features and the app's user flow. Visual designers focus on the finer visual design elements such as visual brand messaging, positioning of UI elements, final colour schemes, ornaments, final fonts etc.

**设计元素。** 线框侧重于特征和 应用的用户流。视觉设计师专注于更精细的视觉设计 视觉品牌信息、UI 元素的定位、最终的配色方案、装饰品、最终字体等。

- **Content.** The creation of visuals that directly relate to content, such as an image that describes or supports a piece of content on the app. Another example maybe a welcome video or info graphic that describes the app's features.

**内容。** 创建直接相关的视觉效果 到内容，例如描述或支持一段 应用程序上的内容。另一个例子可能是欢迎视频或信息 描述应用功能的图形。

# Front-end Development 前端开发

Front-end development, often referred to as client-side development, is the practice of creating and implementing the visual and interactive aspects of a website or web application that users directly interact with. Front-end developers are responsible for building the user interface (UI) and ensuring that the website or application is visually appealing, responsive, and functional across various devices and browsers. In this course, we will focus on Front-end Development. In the other Web course, we will focus on Back-end development.

前端开发，通常称为客户端开发，是创建和实现用户直接与之交互的网站或 Web 应用程序的视觉和交互方面的实践。前端开发人员负责构建用户界面（UI）并确保网站或应用程序在各种设备和浏览器上具有视觉吸引力、响应迅速且功能齐全。在本课程中，我们将重点介绍前端开发。在 Web 课程中，我们将专注于后端开发。

- **HTML (Hypertext Markup Language).** HTML is the foundation of any web page. Front-end developers use HTML to structure the content of the website, including headings, paragraphs, lists, links, images, and other elements.

**HTML (超文本标记语言)。** HTML 是任何网页的基础。前端开发人员使用 HTML 来构建网站内容，包括标题、段落、列表、链接、图像和其他元素。

- **CSS (Cascading Style Sheets).** CSS is used to style the appearance of the web content created with HTML. Front-end developers use CSS to define colors, typography, spacing, layout, and other visual aspects, making the website visually appealing and consistent.

**CSS (级联样式表)。** CSS 用于样式使用 HTML 创建的 Web 内容的外观。前端开发人员使用 CSS 来定义颜色、排版、间距、布局和其他视觉方面，使网站具有视觉吸引力和一致。

- **JavaScript.** JavaScript is a programming language that adds interactivity and dynamic behavior to web pages. Front-end developers use JavaScript to create features like interactive forms, animations, pop-up modals, and more.

**JavaScript。** JavaScript 是一种编程语言，它增加了交互性和动态行为。前端开发人员使用 JavaScript 来创建交互式表单、动画、弹出模式等功能更多。

- **Integration with Back-End.** Front-end developers work closely with back-end developers to integrate the front-end interface with server-side logic and databases, enabling data exchange and dynamic content rendering.

**与后端集成。** 前端开发人员与后端开发人员密切合作，以将前端接口与服务器端逻辑集成，并数据库，支持数据交换和动态内容呈现。



# Back-end Development 后端开发

Backend development, often referred to as server-side development, involves the creation, maintenance, and management of the behind-the-scenes components of a website or web application. Backend developers focus on building the server, databases, and application logic that enable the front-end (user interface) to function smoothly. Backend development is essential for creating the infrastructure and functionality that power web applications, ensuring that data is processed, stored, and retrieved efficiently, and enabling seamless communication between the front-end and various services.

后端开发，通常称为服务器端开发，涉及网站或 Web 应用程序的幕后组件的创建、维护和管理。后端开发人员专注于构建服务器、数据库和应用程序逻辑，使前端（用户界面）能够顺利运行。后端开发对于创建为 Web 应用程序提供支持的基础设施和功能、确保高效处理、存储和检索数据以及实现前端和各种服务之间的无缝通信至关重要。

- **Server-Side Languages.** Backend developers use programming languages such as Python, PHP, Java, PHP, Node.js, and more to write the server-side code that handles requests, implement the application's business logic, algorithms, and data processing tasks. This may involve user authentication, data validation and security, and communications such as sending emails or responding to an automated chat request. One common task of a Back-end developer is to create Content Management Systems (CMS) that allow non-technical content administrators to update the app's content.

**服务器端语言。** 后端开发人员使用编程语言，例如 Python、PHP、Java、PHP、Node.js 等编写服务器端代码 处理请求，实现应用程序的业务逻辑， 算法和数据处理任务。这可能涉及用户 身份验证、数据验证和安全性，以及通信，例如 如发送电子邮件或回复自动聊天请求。— 后端开发人员的常见任务是创建内容管理 允许非技术内容管理员 更新应用的内容。

- **Database and API development.** Backend developers design and implement databases to store and organize data. They interact with databases using query languages like SQL (Structured Query Language) like MySQL or NoSQL databases like MongoDB. Backend developers create Application Programming Interfaces (APIs) that allow different parts of a web application to communicate with each other or enable third-party integration. APIs facilitate data exchange and functionality between the front-end and back-end.

**数据库和 API 开发。** 后端开发人员设计和实现数据库以存储和 组织数据。他们使用查询语言与数据库交互，例如 SQL（结构化查询语言）如 MySQL 或 NoSQL 数据库（如 MongoDB 的 MongoDB 中。后端开发人员创建应用程序编程接口（API） 允许 Web 应用程序的不同部分进行通信 或启用第三方集成。API 促进 前端和后端之间的数据交换和功能。

Backend developers integrate the application with external services, such as payment gateways, APIs from other platforms, and social media platforms, to enhance

functionality and features.

后端开发人员将应用程序与外部服务集成，例如支付网关、来自其他平台的 API 和社交媒体平台，以增强功能和特性。

- **Server Configuration, Scalability, and Performance.** Back-end developers and Dev Ops (Developer Operations) set up and configure web servers (e.g., Apache, Nginx) to handle incoming requests and deliver responses. They also manage server security, scalability, and performance. Back-end developers optimize the performance of the application by fine-tuning database queries, caching data, and minimizing response times to ensure a smooth user experience. These tasks sometimes involve Front-end developers, and, even UX, depending on the issue.

**服务器配置、可扩展性和性能。** 后端开发人员和开发运营（开发人员运营）设置和配置 Web 服务器（例如 Apache、Nginx）以处理传入请求并交付响应。他们还管理服务器安全，可扩展性和性能。后端开发人员优化通过微调数据库查询来提高应用程序的性能，缓存数据，并最大限度地缩短响应时间以确保流畅的用户体验。这些任务有时涉及前端开发人员，并且，甚至 UX，具体取决于问题。

## DevOps 开发运营

This note on DevOps (Developer Operations) is short as it is beyond the scope of this program. However, a concise description is provided.

关于 DevOps（开发人员运营）的这篇说明很简短，因为它超出了该计划的范围。但是，提供了简明的描述。

DevOps supports the development team by enhancing collaboration between the IT team and software development teams. IT is usually responsible for infrastructure-related resources, such as server hardware and software. DevOps supports the app development team with automation techniques for infrastructure, deployment, containerization, and security.

DevOps 通过加强 IT 团队和软件开发团队之间的协作来支持开发团队。IT 通常负责与基础设施相关的资源，例如服务器硬件和软件。DevOps 通过基础设施、部署、容器化和安全性的自动化技术为应用程序开发团队提供支持。

## General Development 一般发展

Both Front-end and Back-end developers:

前端和后端开发人员：

- **Use version control systems** like Git to manage code changes, collaborate with other developers, and maintain a history of modifications.

使用 Git 等**版本控制系统管理**代码更改，与其他开发人员协作，并维护修改。

- **Test their code**, identify and fix issues, and ensure that the application functions correctly. There are various types of testing, including automated unit tests that can identify when an existing feature breaks after fixing another, or implementing a new feature.

**测试他们的代码**，识别和修复问题，并确保应用程序正常运行。有各种类型测试，包括自动化单元测试，这些测试可以识别修复另一个功能或实现新功能后现有功能中断特征。

- **Manage their projects.** Project management tools help us work together effectively. [JIRA](#) is a popular tool used in industry to help us manage projects from conception, through UX, development and testing. Developers use JIRA primarily for tracking features, tasks and bugs. Git integration makes communication and status updates seamless.

**管理他们的项目。** 项目管理工具帮助 我们有效地合作。[吉拉](#) 是工业界使用的一种流行工具，用于帮助我们管理项目 构思，通过用户体验、开发和测试。开发人员使用 JIRA 主要用于跟踪功能、任务和错误。Git 集成使 无缝沟通和状态更新。

## Quality Assurance 质量保证

The worst person to test code is the one who wrote it. In short, QA (Quality Assurance) testers find your bugs! QA refers to the process of systematically testing a website to identify and resolve issues, ensuring that it functions correctly, meets design specifications, and provides a positive user experience. The primary goal of website QA is to uncover and address any defects, errors, or inconsistencies before the app is launched or updated, thereby ensuring its overall quality and reliability. QA is a critical step in the web development lifecycle to deliver a polished and error-free application. Once QA has approved, the application is considered complete. Here is a breakdown of QA tasks:

最不擅长测试代码的人是编写代码的人。简而言之，QA（质量保证）测试人员会发现您的错误！QA 是指系统地测试网站以识别和解决问题的过程，确保其正常运行、满足设计规范并提供积极的用户体验。网站质量保证的主要目标是在应用程序启动或更新之前发现并解决任何缺陷、错误或不一致之处，从而确保其整体质量和可靠性。QA 是 Web 开发生命周期中的关键步骤，以交付精美且无错误的应用程序。一旦 QA 批准，申请即被视为完成。以下是 QA 任务的细分：

- **Functionality Testing.** QA testers verify that all website features, functionalities, and interactions work as intended. This includes testing forms, buttons, links, navigation menus, and any interactive elements.

**功能测试。** QA 测试人员验证所有 网站特性、功能和交互按预期工作。这包括测试表单、按钮、链接、导航菜单和任何 互动元素。

- **Compatibility Testing.** QA ensures that the website functions properly on different browsers (such as Chrome, Firefox, Safari, Edge) and devices (desktops, tablets,

smartphones) to provide a consistent experience.

**兼容性测试。** QA 确保网站在不同浏览器上正常运行（例如 Chrome、Firefox、Safari、Edge）和设备（台式机、平板电脑、智能手机）提供一致的体验。

- **Responsive Design Testing.** QA testers confirm that the website design responds appropriately to various screen sizes and orientations, ensuring that content is displayed correctly and remains usable.

**响应式设计测试。** QA 测试人员确认网站设计对各种屏幕尺寸和方向，确保内容正确显示并保持可用。

- **Performance Testing.** QA assesses the website's speed, load times, and responsiveness to ensure optimal performance under different conditions and user loads.

**性能测试。** QA 评估网站的速度、加载时间和响应能力 确保在不同条件和用户负载下实现最佳性能。

- **Cross-Device Testing.** QA involves testing the website on a variety of devices and operating systems to ensure compatibility and functionality across different platforms.

**跨设备测试。** QA 涉及在各种设备上测试网站并运行 确保不同系统之间的兼容性和功能平台。

- **Usability Testing.** QA assesses the overall user experience by evaluating the website's ease of use, navigation, and clarity of content. Usability testing helps identify any user experience issues.

**可用性测试。** QA 通过评估网站的难易程度来评估整体用户体验 内容的使用、导航和清晰度。可用性测试有助于 识别任何用户体验问题。

- **Content and Visual Checks.** QA ensures that all text, images, videos, and other content are displayed correctly, have the right formatting, and align with the design guidelines.

**内容和视觉检查。** QA 确保所有文本、图像、视频和其他内容都正确显示，具有正确的格式，并与设计指南。

- **Forms and Data Validation.** QA verifies that forms function correctly, validate user input, and provide appropriate error messages. It ensures that user-submitted data is processed accurately.

**表单和数据验证。** QA 验证表单是否正常运行，验证用户输入，以及提供适当的错误消息。它确保用户提交的数据被准确处理。

- **Security and Privacy Testing.** QA includes security checks to identify vulnerabilities, assess data protection measures, and ensure that the website follows

best practices for securing user information.

**安全和隐私测试。** QA 包括安全检查，以识别漏洞、评估数据保护措施，并确保网站遵循最佳实践 用于保护用户信息。

- **Accessibility Testing.** QA ensures that the website adheres to accessibility standards (such as WCAG) to make it usable by people with disabilities. This includes testing screen readers, keyboard navigation, and other assistive technologies.

**可访问性测试。** QA 确保网站遵守可访问性标准（例如 WCAG），使其可供残障人士使用。这包括 测试屏幕阅读器、键盘导航和其他辅助功能 技术。

- **Integration and Database Testing.** QA tests the integration of the website with databases, third-party services, APIs, and other backend components to ensure data consistency and proper functionality.

**集成和数据库测试。** QA 测试网站与数据库、第三方的集成 服务、API 和其他后端组件，以确保数据一致性和适当的功能。

- **Regression Testing.** After making changes or updates to the website, QA includes regression testing to ensure that existing features and functionalities are not adversely affected.

**回归测试。** 对网站进行更改或更新后，QA 包括回归 测试以确保现有特性和功能没有 受到不利影响。

- **User Acceptance Testing (UAT).** In UAT, stakeholders or end-users review and test the website to validate that it meets their expectations and requirements.

**用户验收测试 (UAT)。** 在 UAT 中，利益相关者或最终用户审查和测试网站以 验证它是否满足他们的期望和要求。

## Deployment and Monitoring

### 部署和监控

Deployment refers to making the app live by setting it up on user-accessible servers. QA will test the live app to ensure that deployment went well. One of the most important aspects of making an app available to end users is Monitoring. Monitoring may include usage statistics, performance logs, error logs, and so on. While there are third-party tools available (such as Google Analytics and Cloudflare), Back-end, Front-end, and DevOps developers may create their own tools. Amazon AWS, Microsoft Azure and IBM



Performance Monitoring integrate with their infrastructure offerings.

部署是指通过在用户可访问的服务器上设置应用程序来使应用程序上线。QA 将测试实时应用程序，以确保部署顺利进行。向最终用户提供应用程序的最重要功能之一是监控。监视可能包括使用情况统计信息、性能日志、错误日志等。虽然有可用的第三方工具（例如 Google Analytics 和 Cloudflare），但后端、前端和 DevOps 开发人员可以创建自己的工具。Amazon AWS、Microsoft Azure 和 IBM Performance Monitoring 与其基础架构产品集成。

## Maintenance and Updating

### 维护和更新

During the app's lifetime, changes may need to be made, developers may encounter bugs, or additional features may need to be implemented. These form new tasks for the production teams. These new tasks will generally need to go through the development lifecycle starting with UX Research.

在应用程序的生命周期内，可能需要进行更改，开发人员可能会遇到错误，或者需要实现的其他功能。这些为生产团队形成了新的任务。这些新任务通常需要从用户体验研究开始经历开发生命周期。

## The ecosystem 生态系统

### Browsers 浏览器

Web browsers typically retrieve and present files that are located on a remote machine. Web browsers read markup and presentation files such as HTML, CSS and JavaScript files, tries to make sense of these files, then display the formatted content on the screen. However, not all browsers read, parse and present files in the same way and some browsers are more capable than others. Due to these differences, designers/developers cannot always be sure that their application will look or function the same way across multiple browsers. Therefore, designers/developers should **always test on different browsers**.

Web 浏览器通常检索并显示位于远程机器。Web 浏览器读取标记和演示文件，例如 HTML、CSS 和 JavaScript 文件，试图理解这些文件的意义，然后在屏幕上显示格式化的内容。但是，并非所有浏览器以相同的方式读取、解析和呈现文件，有些浏览器更多比其他人有能力。由于这些差异，设计人员/开发人员无法始终确保他们的应用程序的外观或功能相同跨多个浏览器。因此，设计师/开发人员应该 **始终在不同的浏览器上进行测试**。

The major web browsers are Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari. The Brave browser has been gaining market share in the past few years.



主要的网络浏览器是 Google Chrome、Mozilla Firefox、Microsoft Edge 和 Apple Safari。Brave 浏览器在过去几年中一直在获得市场份额。

Developers should be aware of 2 internal aspects of browsers. The browser's Rendering engine focuses on displaying and arranging web content visually on the screen, while the browser's JavaScript engine is responsible for executing the code that adds interactivity and dynamic behavior to web pages.

开发人员应该注意浏览器的 2 个内部部分。浏览器的渲染引擎专注于在屏幕上直观地显示和排列网页内容，而浏览器的 JavaScript 引擎则负责执行为网页添加交互性和动态行为的代码。

## Rendering engine 渲染引擎

The browser rendering engine is responsible for rendering HTML and CSS, and laying out web content on the screen. It handles the process of taking HTML, CSS, and other resources (like images) and transforming them into a visual representation that users can see and interact with on a web page. The Rendering engines for each browser are listed below:

浏览器渲染引擎负责渲染 HTML 和 CSS，并在屏幕上布局 Web 内容。它处理获取 HTML、CSS 和其他资源（如图像）并将它们转换为用户可以在网页上查看和交互的视觉表示的过程。下面列出了每个浏览器的渲染引擎：

- Blink: Google Chrome, Brave, Edge  
眨眼：谷歌浏览器、勇敢、边缘
- Quantum: Mozilla Firefox  
量子：Mozilla Firefox
- WebKit: Safari WebKit: Safari 浏览器

## JavaScript engine JavaScript 引擎

A JavaScript engine is responsible for executing JavaScript code within web pages. JavaScript is a programming language commonly used to add interactivity and dynamic behavior to websites. When a web page contains JavaScript code, the browser's JS engine processes and executes that code. The JavaScript engines for each browser are listed below:

JavaScript 引擎负责在网页中执行 JavaScript 代码。JavaScript 是一种常用于为网站添加交互性和动态行为的编程语言。当网页包含 JavaScript 代码时，浏览器的 JS 引擎会处理并执行该代码。下面列出了每个浏览器的 JavaScript 引擎：

- V8: Google Chrome, Brave  
V8：谷歌浏览器，勇敢

- Chakra: Edge 查克拉: 边缘
- SpiderMonkey: Mozilla Firefox  
蜘蛛猴: Mozilla Firefox
- JavaScript Core: Safari JavaScript 核心: Safari

## Browser Capabilities 浏览器功能

<https://caniuse.com/> tracks HTML, CSS and JavaScript features that are available in various browsers.

<https://caniuse.com/> 跟踪各种可用的 HTML、CSS 和 JavaScript 功能 浏览器。

<https://whatwebcando.today/> tracks general browser capabilities.

<https://whatwebcando.today/> 跟踪常规浏览器功能。

## Testing tools 测试工具

**The most important tool is Chrome dev tools**

**最重要的工具是 Chrome 开发工具**

Fun & powerful: Intro to Chrome DevTools #DevToolsTips



<https://developers.google.com/web/tools/chrome-devtools/>

Other browsers have similar dev tools!

其他浏览器也有类似的开发工具!

Other tools 其他工具

- <http://www.browserstack.com/>
- <https://smartbear.com/product/bitbar/>
- <https://www.browserling.com/>

- <https://saucelabs.com/>

Remember it is always better to test on actual devices. However the aforementioned tools provide a convenient alternative.

请记住，最好在实际设备上进行测试。然而，上述工具提供了一种方便的替代方案。

- <https://www.seleniumhq.org/> Automates user actions and scripted behaviours  
<https://www.seleniumhq.org/> 自动执行用户作和脚本化行为
- <https://developers.google.com/web/tools/lighthouse> Lighthouse (Audits built into Chrome developer tools)  
<https://developers.google.com/web/tools/lighthouse> Lighthouse (Chrome 开发者工具中内置的审核)

## ► EXTRA CONTENT: The history of browser standards - Contains AI content

浏览器标准的历史 - 包含 AI 内容

# Devices, Resolutions and Operating Systems 设备、分辨率和作系统

We view the web on a variety of devices with different resolutions. As designers, we need to ensure that our websites are usable across a wide range of resolutions. Physical size and pixel density on different devices also affect the way we design. **Front-end developers use Responsive Design techniques to accommodate for different screen resolutions.**

我们在各种不同分辨率的设备上查看网络。如 设计师，我们需要确保我们的网站在广泛的范围内可用 分辨率范围。不同设备上的物理尺寸和像素密度 也会影响我们的设计方式。 **前端开发人员使用响应式设计技术来适应不同的屏幕分辨率。**

Devices and Operating Systems have different capabilities and our applications must live within these limitations. For example, some applications that use geolocation may not be reliable due to the capabilities or even the location of the device. Certain JavaScript features may not be available to all browsers (or they may work differently). **Know the limitations of target devices.**

设备和作系统具有不同的功能，我们的应用程序必须存在于这些限制范围内。例如，由于设备的功能甚至位置，某些使用地理定位的应用程序可能不可靠。某些 JavaScript 功能可能并非适用于所有浏览器（或者它们的工作方式可能不同）。 **了解目标设备的局限性。**

Even the same browser on different Operating Systems may function differently!

即使是不同作系统上的相同浏览器也可能有不同的功能!

See <http://screensiz.es/> for a list of device resolutions.

看 <http://screensiz.es/> 以获取设备分辨率列表。

## Network Speed 网络速度

The network is the biggest bottleneck in ecosystem. Good web developers make things work. Great web developers make things work faster. Network speed can also affect the usability and stickiness of your application. Loading times and responsiveness drastically affect your traffic. As designers/developers, we do not have control over the client's network speed but we do have control of how much data needs to be transferred and the transfer rate of our server. **Be efficient with your network transfer.**

网络是生态系统中最大的瓶颈。优秀的 Web 开发人员 让事情顺利进行。优秀的 Web 开发人员让事情运行得更快。网络 速度也会影响应用程序的可用性和粘性。加载时间和响应能力极大地影响您的流量。如 设计师/开发人员，我们无法控制客户的网络 速度，但我们确实可以控制需要传输多少数据，并且 我们服务器的传输速率。 **提高网络传输效率。**

## Server capabilities 服务器功能

Even though server administration/DevOps is a career of its own, web designers/developers should understand the basics of working with servers. Server hardware, software, security, traffic and speed all affect users and they affect designers/developers. **Learn about the basics of servers**

尽管服务器管理/DevOps 本身就是一个职业，但 Web 设计人员/开发人员应该了解使用服务器的基础知识。服务器硬件、软件、安全性、流量和速度都会影响用户 它们会影响设计师/开发人员。**了解服务器的基础知识**

## Analytics 分析学

The ability to measure and understand the effectiveness of our websites and apps is critical to improving them. This is where analytics comes in. Using Analytics tools, we can view reports on different types of users, browser usage, device usage etc. **Analytics helps stakeholders make decisions**

衡量和了解我们网站有效性的能力 应用程序对于改进它们至关重要。这就是分析的用武之地。使用分析工具，我们可以查看有关不同类型用户的报告，浏览器使用情况、设备使用情况等。 **分析帮助利益相关者做出决策**

One of the most common reporting tools is [Google Analytics](#). There are a variety of Analytics tools available, however, Google Analytics is widely used.

最常见的报告工具之一是 [谷歌分析](#)。有多种可用的分析工具，但是，谷歌分析被广泛使用。

## Environments 环境

When developers code apps, they usually code on their own machine (local environment) and deploy to another machine (server environment). If the machines do not have the same configurations, software versions, dependencies, operating systems etc., the app may not run properly, or may even fail to start in different environments. Here are 4 common environments:

当开发人员编写应用程序代码时，他们通常在自己的计算机（本地环境）上编写代码并部署到另一台计算机（服务器环境）。如果机器没有相同的配置、软件版本、依赖项、操作系统等，应用程序可能无法正常运行，甚至可能无法在不同环境中启动。以下是 4 种常见环境：

- **Local**. This is your machine that you use to code

**本地**。这是您用来编码的机器

- **Dev**. This is a machine that your development team shares. Usually, dev is used to test and experiment with new features

**开发**。这是您的开发团队的机器 股票。通常，dev 用于测试和试验新功能

- **QA**. This is a machine that your QA team uses to test your completed work.

**质量保证**。这是您的 QA 团队用来测试的机器 您完成的工作。

- **Staging**. This is a machine that your Client (and possibly some users) use to view your completed work before they approve it.

**分期**。这是您的客户（和 可能有些用户）用于在他们批准之前查看已完成的工作 它。

- **Production** or "Live". This is a machine that your end users access to use your final application.

**生产或“直播”**。这是一台机器，你的终结 用户访问权限以使用您的最终应用程序。

Git "branches" can be used to manage deployment to different environments. More on that later on.

Git“分支”可用于管理对不同环境的部署。稍后会详细介绍。

**Containerization** enables apps and all their dependencies and configurations to be packaged and together become more portable and avoid environment-related discrepancies. Containers can also be "versioned". There are other benefits to containerization including advanced topics like orchestration and scalability. However,

those are more DevOps-related topics that go beyond the scope of his course.

**容器化支持**应用程序及其所有依赖项 以及要打包并一起变得更加便携的配置 避免与环境相关的差异。容器也可以是“版本化”。容器化还有其他好处，包括 业务流程和可伸缩性等高级主题。然而，这些都是 更多与 DevOps 相关的主题超出了他的课程范围。